

# EASYDL 文档





#### 【版权声明】

版权所有©百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司。未经本公司书面许可，任何单位和个人不得擅自摘抄、复制、传播本文档内容，否则本公司有权依法追究法律责任。

#### 【商标声明】



和其他百度系商标，均为百度在线网络技术（北京）有限公司、北京百度网讯科技有限公司的商标。本文档涉及的第三方商标，依法由相关权利人所有。未经商标权利人书面许可，不得擅自对其商标进行使用、复制、修改、传播等行为。

#### 【免责声明】

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导。如您购买本文档介绍的产品、服务，您的权利与义务将依据百度智能云产品服务合同条款予以具体约定。本文档内容不作任何明示或暗示的保证。

## 目录

目录	2
平台简介	9
什么是EasyDL	9
EasyDL产品体系	12
EasyDL产品优势	14
EasyDL常用概念	15
AI开发基础知识	17
文心大模型	18
新手指南	19
EasyDL图像-图像分类快速开始	19
EasyDL图像-物体检测快速开始	31
EasyDL文本-文本分类单标签快速开始	44
EasyDL零售行业版快速开始	54
EasyDL图像SDK集成快速开始	61
通用设备端Android ARM	61
通用设备端Linux ARM	77
通用设备端Windows x86加速版	83
服务器端Linux GPU 加速版	88
专项适配硬件EdgeBoard(FZ)	94
专项适配硬件Jetson	100
价格说明	104
EasyDL图像价格说明	104
EasyDL图像本地服务器部署价格说明	104
EasyDL图像软硬一体方案价格说明	105
EasyDL图像通用小型设备部署价格说明	105
EasyDL图像价格常见问题	105
EasyDL图像公有云API价格说明	106
EasyDL图像价格整体说明	108
EasyDL图像算力资源价格说明	109
EasyDL文本价格说明	110
文本私有服务器部署价格说明	110
EasyDL文本公有云API价格说明	110
EasyDL文本价格整体说明	112
EasyDL文本算力价格说明	113
EasyDL结构化数据价格说明	113
EasyDL结构化数据公有云API价格说明	113
表格预测算力资源价格说明	115
EasyDL结构化数据价格整体说明	115
EasyDL视频价格说明	116
EasyDL视频公有云API价格说明	116

EasyDL视频本地服务器部署价格说明	117
EasyDL视频设备端部署价格说明	118
EasyDL视频软硬一体方案价格说明	118
EasyDL视频价格整体说明	118
EasyDL视频算力资源价格说明	119
EasyDL语音价格说明	120
EasyDL语音公有云API价格说明	120
EasyDL语音本地服务器部署价格说明	121
EasyDL语音本地设备端部署价格说明	122
EasyDL语音价格整体说明	122
EasyDL语音算力资源价格说明	123
EasyDL零售行业版价格说明	123
价格整体说明	123
公有云API价格说明	124
本地部署价格说明	129
公有云API价格说明	129
价格整体说明	135
EasyDL跨模态价格说明	135
EasyDL跨模态公有云API价格说明	135
EasyDL跨模态算力资源价格说明	137
EasyDL 图像使用说明	138
EasyDL图像介绍	138
图像分类	140
整体介绍	140
数据准备	141
模型训练	156
模型发布	166
端云协同服务	167
常见问题	346
物体检测	348
整体介绍	348
数据准备	349
模型训练	364
模型发布	374
端云协同服务	374
常见问题	567
图像分割	570
整体介绍	570
数据准备	570
模型训练	587
模型发布	594

端云协同服务	594
常见问题	711
EasyDL 文本使用说明	713
EasyDL文本介绍	713
文本分类-单标签	715
整体介绍	715
数据准备	715
模型训练	731
模型部署	735
常见问题	748
文本分类-多标签	749
整体介绍	749
数据准备	750
模型训练	761
模型部署	765
情感倾向分析	775
整体介绍	775
数据准备	776
模型训练	786
模型部署	791
文本实体关系抽取	801
整体介绍	801
数据准备	801
模型训练	806
模型部署	808
文本实体抽取	818
整体介绍	818
数据准备	818
模型训练	829
模型部署	832
短文本相似度	838
整体介绍	838
数据准备	839
模型训练	844
模型部署	848
评论观点抽取	858
整体介绍	858
数据准备	859
2.导入未标注文本数据	860
模型训练	864

Baidu 百度智能云文档	目录
模型发布	868
文本创作 (已下线)	880
文本创作介绍	880
数据准备	880
模型训练	884
模型发布	885
EasyDL 语音使用说明	891
EasyDL语音介绍	891
语音识别	892
语音识别介绍	892
创建模型	894
训练模型	896
上线模型	897
模型调用	897
声音分类	899
声音分类整体说明	899
数据准备	900
模型训练	909
模型发布	913
9.浏览器打开webui上传文件提示 500 internal server error	940
EasyDL 视频使用说明	941
EasyDL视频介绍	941
视频分类	942
视频分类介绍	942
创建模型	943
数据准备	944
上传视频分类数据集	945
模型训练	953
模型发布	958
常见问题	963
目标跟踪	964
目标跟踪介绍	964
创建模型	965
数据准备	966
模型训练	971
模型发布	975
EasyDL 结构化数据使用说明	1012
EasyDL结构化数据介绍	1012
表格数据预测	1012
表格数据预测介绍	1012
简介	1012



数据准备	1013
模型训练	1014
模型发布	1020
故障处理	1038
时序预测	1039
时序预测介绍	1039
简介	1039
数据准备	1040
模型训练	1042
模型发布	1047
故障处理	1051
EasyDL OCR使用说明	1052
EasyDL OCR介绍	1052
API文档	1053
错误码	1054
EasyDL 跨模态使用说明	1055
EasyDL跨模态整体介绍	1055
图文匹配	1056
整体介绍	1056
数据准备	1057
模型训练	1060
模型评估	1062
模型部署	1064
发布模型，生成在线API	1064
EasyDL 零售行业版使用说明	1069
零售版服务介绍	1069
购买指南	1071
场景范例	1078
拓展门店验证	1078
业务门店拜访	1079
店内陈列洞察	1081
异常拍照监测	1082
定制商品检测服务	1083
服务介绍	1083
购买指南	1084
快速训练一个模型	1087
模型创建	1095
数据准备	1097
数据标注	1123
模型训练	1140
模型效果评估	1142

模型优化	1145
模型发布	1149
模型使用	1151
门店管理	1158
API文档	1165
翻拍识别服务	1176
服务介绍	1176
购买指南	1177
API文档	1178
购买指南	1181
门店拜访SDK (原货架拼接服务)	1183
服务介绍	1183
购买指南	1184
体验APP	1186
API文档	1187
SDK文档	1205
价签识别服务	1228
服务介绍	1228
购买指南	1228
API文档	1230
窜拍识别服务	1233
服务介绍	1233
API文档	1234
零售版常见问题	1241
零售版服务介绍	1243
购买指南	1246
数据看板	1252
零售版常见问题	1252
EasyDL桌面版使用说明 (已下线)	1255
产品简介	1255
产品介绍	1255
功能介绍	1255
系统支持	1256
下载与激活	1257
AI开发基础知识	1261
快速开始	1262
用零代码开发实现图像分类	1262
用零代码开发实现物体检测	1270
用零代码开发实现实例分割	1278
用零代码开发实现语义分割	1285

数据管理	1290
数据导入	1290
查看数据集	1294
创建数据集	1296
数据标注	1297
开发与训练	1310
零代码开发	1310
Notebook开发	1324
模型中心	1334
模型列表	1334
模型部署	1335
服务列表	1335
智能边缘控制台	1335
离线SDK部署说明	1341
专项适配硬件离线部署	1341
服务器离线部署	1351
通用小型设备离线部署	1370
AutoDL模式算法适配硬件	1385
高级调参模式算法适配硬件	1386
常见问题	1387
分享我的模型	1388
分享EasyDL定制化模型	1388
分享EasyDL定制化API	1391
版本更新记录	1393
2021年01月	1395
常见问题	1397
常见问题	1397
智能边缘控制台	1400
智能边缘控制台-单节点版	1400
智能边缘控制台-多节点版	1411
EasyEdge 智能边缘控制台-单节点版 IEC API	1422
EasyEdge 智能边缘控制台-多节点版 IECC API	1440

# 平台简介

## 什么是EasyDL

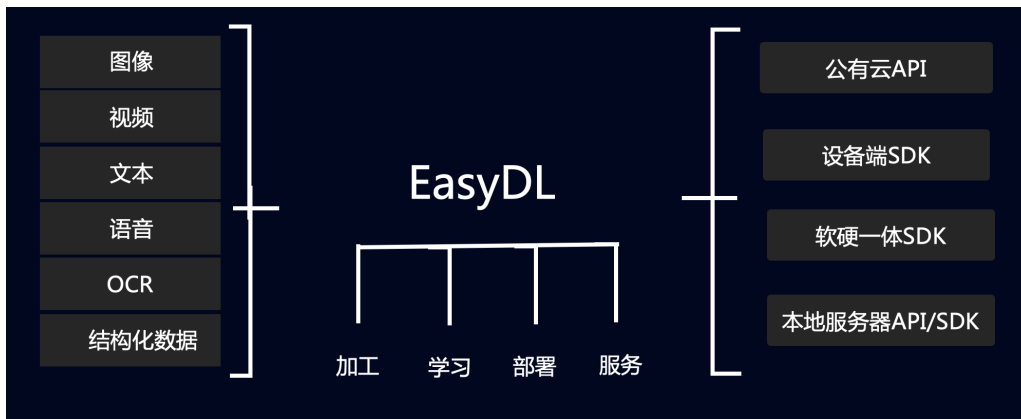
### 目录

1. 产品介绍
2. 应用场景及案例

### 产品介绍

EasyDL从2017年11月中旬起，在国内率先推出针对AI零基础或者追求高效率开发的企业用户的零门槛AI开发平台，提供从数据采集、标注、清洗到模型训练、部署的一站式AI开发能力。对于各行各业有定制AI需求的企业用户来说，无论您是否具备AI基础，EasyDL设计简约，极易理解，最快5分钟即可上手学会，15分钟完成模型训练。

您采集到的原始图片、文本、音频、视频、OCR、表格等数据，经过EasyDL加工、学习、部署后，可通过公有云API调用，或部署在本地服务器、小型设备、软硬一体方案的专项适配硬件上，通过离线SDK或私有API进一步集成，流程如下：



根据企业用户的应用场景及深度学习的技术方向，EasyDL共推出6大通用产品及1个行业产品：

- **EasyDL 图像**：定制基于图像进行多样化分析的AI模型，实现图像内容理解分类、图中物体检测定位等，适用于图片内容检索、安防监控、工业质检等场景
- **EasyDL 文本**：定制基于文心大模型的语义理解AI模型，提供一整套文本定制与应用能力，适用于文本内容审核、文本自动生成、留言分类、电商评价打分等场景
- **EasyDL 语音**：定制语音识别模型，精准识别业务专有名词，适用于数据采集录入、语音指令、呼叫中心等场景，以及定制声音分类模型，适用于区分不同声音类别等场景
- **EasyDL OCR**：定制文字识别模型，结构化输出关键字段内容，满足个性化卡证票据识别需求，适用于证照电子化审批、财税报销电子化等场景
- **EasyDL 视频**：定制基于视频片段内容进行分类的AI模型，适用于区分不同短视频类别等场景，以及定制目标追踪AI模型，实现跟踪视频中特定目标对象及轨迹，适用于视频内容审核、人流/车流统计、养殖场牲畜移动轨迹分析等场景
- **EasyDL 结构化数据**：挖掘数据中隐藏的模式，解决二分类、多分类、回归等问题，适用于客户流失预测、欺诈检测、价格预测等场景
- **EasyDL 零售行业版**：面向零售场景的ISV、零售行业服务商等企业用户，提供基于商品识别场景的AI服务解决方案，适用于货架巡检、自助结算台、无人零售柜等场景

### 应用场景及案例

#### 工业质检

**产品组装合格性检查**：在流水线作业中针对组合型产品键盘可能存在的不合格情况进行图片收集，将图片完成标注后发起模型训练，从而训练出自动判断合格或不合格的模型，辅助人工判断产品质量

具体案例：对键盘生产可能存在错装、漏装、合格等情况进行识别

任务类型：物体检测



**商品瑕疵检测：** 针对商品微小瑕疵进行图片收集，并对原始图片或基于光学成像形成的图片进行瑕疵标注及训练，将模型集成在检测设备或流水线中，辅助人工提升质检效率，降低人力成本

具体案例：针对地板质检的常见问题（例如，虫眼、毛面、棘爪等）进行检测

任务类型：物体检测



电商/网站图片分析

**海量图片打分类标签：** 电商将图片按定制标签标注并训练，构建对海量图片自动打标签模型，实现将图片面向不同C端用户的精准展示，以及基于分析用户所点击图片的内容从而进行相关图片推荐等功能

具体案例：家装网站将卧室、餐厅、厨房等图片进行内容分类

任务类型：图像分类



**定制图像审核策略：** 根据需求制定图片审核标准，网站用EasyDL训练的模型判断UGC图片是否合规，适用于视频、新闻等内容平台定制内容审核策略，过滤不良信息，或用于线上活动，判断C端用户提交图片的合规性。

具体案例：房产网站审核用户提交信息是房源图还是非房源图片

任务类型：图像分类



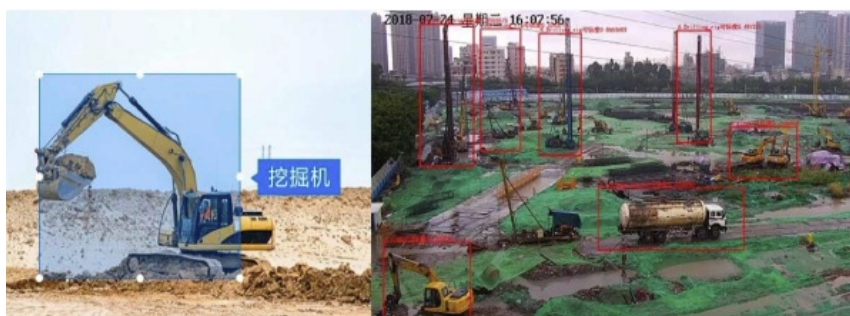


## 安防监控

**生产环境安全监控：**对生产环境现场进行安全性监控，对是否出现挖掘机等危险物品、工人是否佩戴安全帽/穿工作服、施工区是否有烟火等情况进行检查，辅助人工判断安全隐患并及时预警，保证生产环境安全运行

具体案例：对输电线路附近是否存在挖掘机、吊车等外部隐患物体进行检测

任务类型：物体检测



**超市防损监控：**在安全死角安装摄像头，将采集到图片进行标注及训练，实现实时检测图片中是否有未结算商品

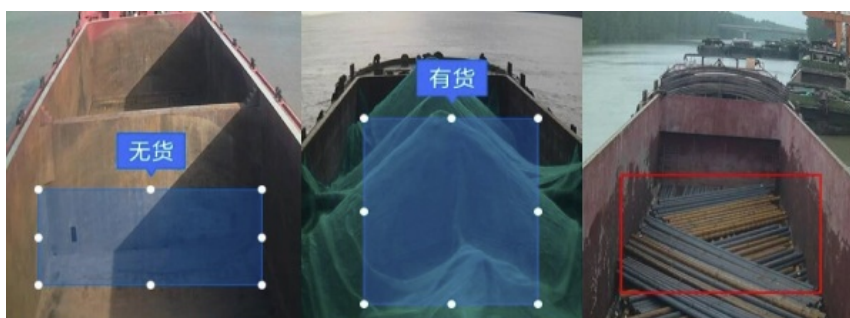
具体案例：在超市结算台下层安装摄像头，将视频抽帧为图片后可判断图片中有未结算商品、无未结算商品

任务类型：物体检测



**货物状态监控** 根据实际业务场景在轮船内安装摄像头，采用定时抓拍或视频抽帧的方式，自动判断货物状态，提升业务运营、货品管理效率

具体案例：货船调运公司安装摄像头智能监控船上货品状态为有货或无货

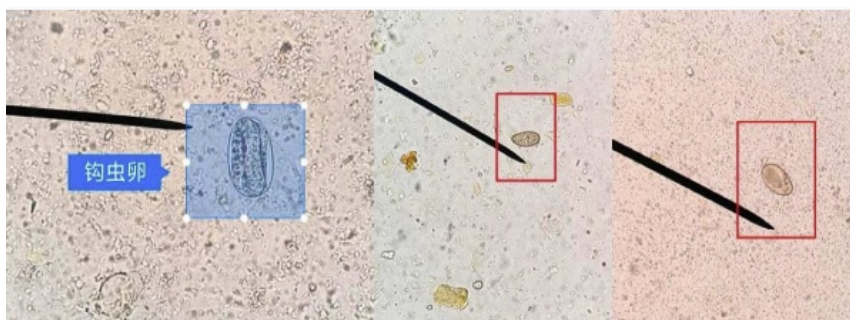


## 专业领域研究

**医疗镜检识别：**针对医疗检验场景中可能存在的正常或异常结果进行图片收集，并基于图片关键特征进行标注完成训练，协助医生高效完成结果判断

具体案例：针对寄生虫卵镜检图片，判断虫卵类型从而对症下药

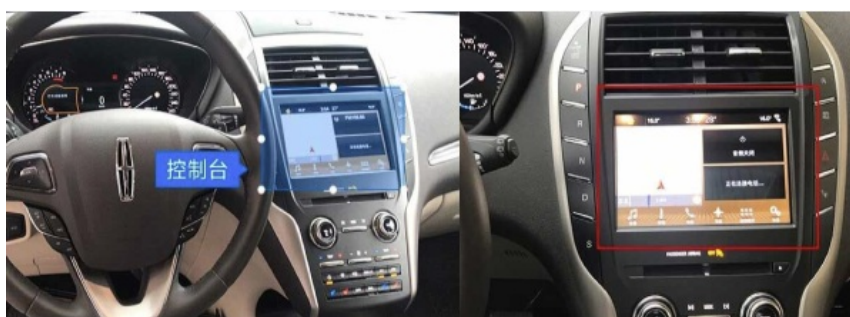
任务类型：物体检测



**培训或科普：**将图片中专业部件进行标注，训练出专业零部件识别模型，拍照后识别出详细部件并配合提供详细的部件介绍信息，将该能力集成在公司内部使用的培训app中，方便新人通过拍照识图快速上手业务

具体案例：汽车公司对内部人员提供车辆零部件拍图能力

任务类型：物体检测



#### 零售商品识别

**货架陈列合规性检查：**将商品陈列图片进行采集、标注及训练，集成在app中供巡货员或者店员拍照上传自动识别，通过系统自动检查完成合规性的准确检查

具体案例：零售快消公司在各商超的货架及货柜中拍照自动检测出第一排商品个数、位置完成自动巡店

任务类型：商品检测



**电商平台商品管理：**对电商商品图进行采集并标注明显特征，训练出商品识别模型，实现对商品图库的快速分类，减少人工入库的标注成本

具体案例：鞋类电商对鞋底花纹进行识别，自动判断鞋子品类

任务类型：物体检测



[查看更多案例](#)

EasyDL产品体系

## 目录

1. 通用产品
2. 行业产品

### 通用产品

根据企业用户的应用场景及深度学习的技术方向，EasyDL先后推出了以下6大通用产品，充分满足各行业的AI定制化需求

### 适合人群

对AI有需求的各行业企业客户或服务集成商

### 产品特性

- **零门槛**：无需具备算法基础，最快5分钟上手学会，15分钟完成模型训练，并将产出模型快速集成到业务产线
- **高精度**：基于百度文心大模型底座，内置百度超大规模预训练模型和自研AutoDL技术，只需少量数据就能训练出高精度模型

### 具体产品

- **EasyDL 图像**：定制基于图像进行多样化分析的AI模型，实现图像内容理解分类、图中物体检测定位等，适用于图片内容检索、安防监控、工业质检等场景 [立即使用](#)
- **EasyDL 文本**：定制基于文心大模型的语义理解AI模型，提供一整套文本定制与应用能力，适用于文本内容审核、文本自动生成、留言分类、电商评价打分等场景 [立即使用](#)
- **EasyDL 语音**：定制语音识别模型，精准识别业务专有名词，适用于数据采集录入、语音指令、呼叫中心等场景，以及定制声音分类模型，适用于区分不同声音类别等场景 [立即使用](#)
- **EasyDL OCR**：定制文字识别模型，结构化输出关键字段内容，满足个性化卡证票据识别需求，适用于证照电子化审批、财税报销电子化等场景 [立即使用](#)
- **EasyDL 视频**：定制基于视频片段内容进行分类的AI模型，适用于区分不同短视频类别等场景，以及定制目标追踪AI模型，实现跟踪视频中特定目标对象及轨迹，适用于视频内容审核、人流/车流统计、养殖场牲畜移动轨迹分析等场景 [立即使用](#)
- **EasyDL 结构化数据**：挖掘数据中隐藏的模式，解决二分类、多分类、回归等问题，适用于客户流失预测、欺诈检测、价格预测等场景 [立即使用](#)

### 行业产品

**EasyDL零售行业版**是专用于零售行业用户训练商品检测模型的模型训练平台，平台提供海量预置的商品图片，开放基于百度大规模零售数据的预训练模型、及数据增强合成技术，实现低成本获得高精度商品检测AI模型服务

### 适合人群

有商品识别需求的零售行业企业或服务集成商，对货架巡检、无人货柜、无人结算台等零售场景有AI定制化需求

### 产品特性

- **高可用**：针对零售场景专项算法调优，基于百度大规模零售数据的预训练模型，集成图像合成与增强技术提升模型泛化能力，模型准确率可达到97%，保证模型在生产环境中具有高可用性
- **全功能**：对货架巡检场景的业务场景提供了货架拼接SDK及API接口，功能强大，体验更优

### 具体产品

- **定制商品检测API**：训练定制化商品检测模型，平台提供海量预置商品图片，开放基于百度大规模零售数据的预训练模型及数据增强合成技术，实现低成本获得高精度商品检测AI模型服务
- **标准商品检测API**：无需训练即可直接使用的商品检测API，支持零售商超常见商品品类检测，针对货架合规性检查场景专项调优，适应大型商超、便利店、街边店等多种复杂货架场景接口返回商品名称、规格、品类及在图片中的位置
- **货架拼接SDK**：货架拼接服务支持将多个货架局部图片或视频，组合为完整货架图片，同时支持输出在完整货架图中的商品检测结果，包含SKU的名称、数量，适用于需要在长货架进行商品检测的业务场景

[立即使用](#)



## EasyDL产品优势

### 目录

1. [零门槛](#)
2. [高精度](#)
3. [低成本](#)
4. [广适配](#)
5. [可交易](#)

#### ☞ 零门槛

EasyDL提供围绕AI开发、部署的端到端一站式能力，包括数据采集、标注、清洗、模型训练、模型评估、模型部署。平台设计简约，极易理解，最快5分钟即可上手，15分钟完成模型训练



操作流程如下：

**Step 1 创建模型** 确定模型名称，可添加模型描述便于后续模型迭代管理

**Step 2 上传并标注数据** 上传数据后，根据不同模型类型的数据要求进行标注，如果有本地已标注的数据，也可以直接上传

**Step 3 训练模型并校验效果** 选择算法类型、配置训练任务相关参数完成训练任务启动。模型训练完毕后支持可视化查看模型效果评估报告，也支持通过**模型校验**功能在线上传实测数据验证模型效果

**Step 4 部署模型** 根据业务场景，支持将模型部署为公有云API实现在线调用，或部署在本地服务器/小型设备/软硬一体方案的专项适配硬件上，通过API/SDK集成离线应用

具体操作流程详见[新手指南](#)

#### ☞ 高精度

EasyDL底层框架由百度自研的飞桨深度学习框架构建而成，内置基于百度文心大模型底座基础之上的成熟预训练模型，并结合百度自研的AutoDL技术，助力用户基于少量数据就能获得具备出色效果与性能模型

**文心大模型** 面向语言理解、语言生成等文本场景，具备超强语言理解能力以及对话生成、文学创作等能力。创新性地将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

**AutoDL技术** 面向视觉场景，百度研发的AutoDL Transfer模型结合了模型网络结构搜索、迁移学习技术，可针对用户数据进行算法自动优化。与通用算法相比，更适用于细分类场景。例如，通用算法可用于区分猫与狗，但如果要区分不同品种的猫，则AutoDL效果会更好

#### ☞ 低成本

数据对于模型效果至关重要，面向数据服务，EasyDL除提供基础的数据上传、存储、标注外，还提供业界领先的数据采集、智能标注、多人标注、云服务数据回流等多种低成本数据管理服务，大幅降低开发者训练数据处理成本

**端云协同数据采集** EasyDL提供便捷的本地数据采集软件，支持定时拍照、视频抽帧（支持自定义抽帧规则）多种采集方式，并将图片即时同步到平台管理，无需摄像头数据反复下载与重新导入 [了解详情](#)

**智能标注** 智能标注为一套人机交互的协作标注方式，在手工标注少量数据后，系统会从数据集所有样本中筛选出最关键的难例并提示需优先标注。通常情况下，只需标注数据集30%左右的数据即可训练模型，与标注所有数据后训练相比，模型效果几乎等同 [了解详情](#)

**多人标注** 通过将数据集在线共享给团队成员，实现多人分工标注、审核数据，有效降低标注成本，通过多人协作提高标注效率 [了解详情](#)

**云服务数据回流** 当将EasyDL训练的模型以公有云方式部署在业务场景中时，通过开通\*云服务数据回流功能，可将实际调用后的业务场景真实数据及识别结果在平台中查看和管理，将识别错误的图片结果人工筛选后保存至数据集持续训练，可有效长期提升模型效果 [了解详情](#)

## 🔗 广适配

EasyDL模型训练阶段需要在线训练，训练完成后，可将模型部署在公有云服务器、本地服务器、小型设备、软硬一体方案专项适配硬件上，通过API或SDK进行集成，充分满足各种业务场景对模型部署的要求

### 公有云API

1. 训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整快速集成
2. 支持在线调用的高并发请求及灵活的扩缩容策略，提高资源利用率
3. 支持查找云端模型识别错误的结果，纠正结果并将其加入模型迭代的训练集，不断优化模型效果

### 本地服务器部署

1. 可将训练完成的模型部署在私有CPU/GPU服务器上，支持API和SDK两种集成方式
2. 可在内网/无网环境下使用模型，确保数据隐私

### 本地设备端部署

1. 训练完成的模型被打包成适配智能硬件的SDK，可进行设备端离线计算，有效满足业务场景中无法联网、对数据保密性要求较高、响应时延要求更快的需求
2. 支持iOS、Android、Linux、Windows四种操作系统，基础接口封装完善，满足灵活的应用侧二次开发

### 软硬一体方案

为进一步提升前端智能计算的用户体验，EasyDL推出[前端智能计算-软硬一体方案](#)，将百度推出的高性能硬件与EasyDL图像分类/物体检测模型深度适配，可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松

具体部署流程详见[图像分类](#)、[物体检测](#)

## 🔗 可交易

EasyDL致力于打造全面开放的AI开发生态，与百度AI市场无缝对接，支持模型、AI服务、智能硬件等自由交易

**AI模型交易** 售卖者层面，用户将EasyDL训练完成的模型在AI市场开放售卖，可获取现金收益或平台积分；购买者层面，用户在AI市场购买业务场景相似的EasyDL模型，并基于已购模型再训练，仅需添加少量数据，即可快速获得高精度AI模型

**AI服务交易** 用户将成功发布的公有云API在AI市场开放售卖，可获取现金收益或平台积分

**智能硬件交易** 用户可在AI市场购买EasyDL软硬一体方案，同时获得EasyDL专项适配硬件和EasyDL软件的使用授权，实现EasyDL产出的模型无缝集成进专项适配硬件

具体交易详情详见[AI市场](#)

## EasyDL常用概念

### 目录

1. [模型与模型类型](#)
2. [模型训练相关](#)
3. [模型效果相关](#)
4. [模型部署相关](#)
5. [数据相关](#)



## 🔗 模型与模型类型

EasyDL支持6大技术方向，每个方向包括不同的模型类型：

- **EasyDL 图像**：图像分类、物体检测、图像分割
- **EasyDL 文本**：文本分类-单标签、文本分类-多标签、文本实体抽取、情感倾向分析、短文本相似度
- **EasyDL 语音**：语音识别、声音分类
- **EasyDL OCR**：文字识别
- **EasyDL 视频**：视频分类、目标跟踪
- **EasyDL 结构化数据**：表格预测

## 🔗 模型训练相关

### 🔗 AutoDL Transfer

AutoDL Transfer模型是百度研发的AutoDL技术之一，结合了模型网络结构搜索、迁移学习技术、并针对用户数据进行自动优化。与通用算法相比，训练时间较长，但更适用于细分类场景。例如，通用算法可用于区分猫和狗，但如果要区分不同品种的猫，则AutoDL效果会更好

### 🔗 ERNIE

领先的语义理解技术与平台文心（ERNIE），依托飞桨打造，集先进的预训练模型、全面的NLP算法集、端到端开发套件和平台化服务于一体，提供一站式NLP开发与服务，让您更简单、高效地定制企业级文本模型。文心提供的ERNIE预训练模型，已累计学习10亿多知识，能够助力各NLP任务快速提升效果。平台内置了最新的ERNIE2.0，并提供了ERNIE2.0-Base、ERNIE2.0-Large两个版本供用户选择。

## 🔗 模型效果相关

### 🔗 准确率

图像分类/文本分类/声音分类等分类模型的衡量指标，正确分类的样本数与总样本数之比，越接近1模型效果越好

### 🔗 F1-score

对某类别而言为精确率和召回率的调和平均数，对图像分类/文本分类/声音分类等分类模型来说，该指标越高效果越好

### 🔗 精确率(Precision)

对某类别而言为正确预测为该类别的样本数与预测为该类别的总样本数之比

### 🔗 召回率(Recall)

对某类别而言为正确预测为该类别的样本数与该类别的总样本数之比

### 🔗 top1、top2...top5

在查看图像分类/文本分类/声音分类/视频分类模型评估报告中，top1-top5指的是针对一个数据进行识别时，模型会给出多个结果，top1为置信度最高的结果、top2次之...正常业务场景中，我们通常会采信置信度最高的识别结果，重点关注top1的结果即可。

### 🔗 mAP

mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务，每一类object都可以计算出其精确率(Precision)和召回率(Recall)，在不同阈值下多次计算/试验，每个类都可以得到一条P-R曲线，曲线下的面积就是average

### 🔗 阈值

物体检测模型会存在一个可调节的阈值（threshold），是正确结果的判定标准，例如阈值是0.6，置信度大于0.6的识别结果会被当作正确结果返回。每个物体检测模型训练完毕后，可以在模型评估报告中查看推荐阈值，在推荐阈值下F1-score的值最高。

## 🔗 模型部署相关

### 🔗 公有云API

模型部署为Restful API，可以通过HTTP请求的方式进行调用。

### 🔗 设备端SDK

模型部署为设备端SDK，可集成在前端智能计算硬件设备中，可完全在无网环境下工作，所有数据皆在设备本地运行处理。目前支持IOS、

ANDROID、WINDOWS、LINUX四种操作系统及多款主流智能计算硬件。

## 本地服务器部署

模型部署为本地服务器部署，可获得基于定制EasyDL模型封装而成的本地化部署的方案，此软件包部署包开发者本地的服务器上运行能够得到与在线API功能完全相同的接口。

## 软硬一体方案

目前EasyDL支持两款软硬一体硬件，包括EasyDL-EdgeBoard软硬一体方案及EasyDL-十目计算卡。通过在AI市场购买，可获得硬件+专项适配硬件的设备端SDK，支持在硬件中离线计算。

## 数据相关

## 智能标注

智能标注为一套人机交互的协作标注方式，目前EasyDL物体检测训练任务支持智能标注，在手工标注少量数据后，系统会从数据集所有图片中筛选出最关键的图片并提示需要优先标注。通常情况下，只需标注数据集30%左右的数据即可训练模型。与标注所有数据后训练相比，模型效果几乎等同。

# AI开发基础知识

## 目录

- [1. AI概念及基本原理](#)
- [2. AI模型训练的基本流程介绍](#)

## AI概念及基本原理

人工智能（Artificial Intelligence，英文缩写为AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能企图生产出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理等。

在EasyDL平台背后主要使用了深度学习的技术，深度学习是机器学习(ML, Machine Learning)领域中一个新的研究方向。通过学习样本数据的内在规律和表示层次，最终目标是让机器能够像人一样具有分析学习能力，能够识别文字、图像和声音等数据。

## AI模型训练的基本流程介绍



### 1. 分析业务需求

在正式启动训练模型之前，需要有效分析和拆解业务需求，明确模型类型如何选择。这里我们可以举一些实际业务场景进行分析。

**举例：原始业务需求**—某企业希望为某个高端小区物业做一套智能监控系统，希望对多种现象智能监控并及时预警，包括保安是否在岗、小区是否有异常噪音、小区内各个区域的垃圾桶是否已满等多个业务功能。

针对这个原始业务需求，我们可以分析出不同的监控对象所在的位置不同、监控的数据类型不同（有的针对图片进行识别、有的针对声音进行判断），需要多个模型综合应用。

**监控保安是否在岗**——通过图像分类模型进行判断

**监控小区是否有异常噪音**——定时收集声音片段通过声音分类模型进行判断

**监控小区内各个区域垃圾桶是否已满**——由于监控区域采集的画面可能会存在多个垃圾桶，此处需要通过物体检测模型进行判断。

### 2. 采集/收集数据

在通过上述第一步分析出基本的模型类型，需要进行相应的数据收集工作。数据的主要原则为尽可能采集真实业务场景一致的数据，并覆盖可能的各种情况

### 3. 标注数据

采集数据后，可以通过EasyDL在线标注工具或线下其他标注工具对已有的数据进行标注。如上述保安是否在岗的图像分类模型，需要将监控视频抽帧后的图片按照【在岗】及【未在岗】两类进行整理；小区内各个区域垃圾桶是否已满，需要将监控视频抽帧后的图片标注其中每个垃圾桶的【空】【满】两种状态进行标注。

#### 4. 训练模型

训练模型阶段可以将已有标注好的数据基于已经确定的初步模型类型，选择算法进行训练。通过使用EasyDL平台，可以可视化在线操作训练任务的启停、训练任务的配置。可以大幅减少线下搭建训练环境、自主编写算法代码的相关成本。

#### 5. 评估模型效果

训练后的模型在正式集成之前，需要评估模型效果是否可用。在这个环节上EasyDL提供了详细的模型评估报告，以及在线可视化上传数据测试模型效果的功能。

#### 6. 部署模型

当确认模型效果可用后，可以将模型部署至生产环境中。传统的方式需要将训练出的模型文件加入工程化相关处理，通过使用EasyDL，可以便捷地将模型部署在公有云服务器或本地设备上，通过API或SDK集成应用，或直接购买软硬一体产品，有效应对各种业务场景所需，提供效果与性能兼具的服务。

## 文心大模型

### 文心大模型是什么

文心大模型是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。目前文心大模型包括：

### 文心·NLP大模型

面向语言理解、语言生成等NLP场景，具备超强语言理解能力以及对话生成、文学创作等能力。创新性地将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

### 文心·CV大模型

基于领先的视觉技术，利用海量的图像/视频等数据，为企业/开发者提供强大的视觉基础模型，以及一整套视觉任务定制与应用能力。

### 文心·跨模态大模型

基于知识增强的跨模态语义理解关键技术，可实现跨模态检索、图文生成、图片文档的信息抽取等应用的快速搭建，落实产业智能化转型的AI助力

### 如何在EasyDL上使用文心大模型

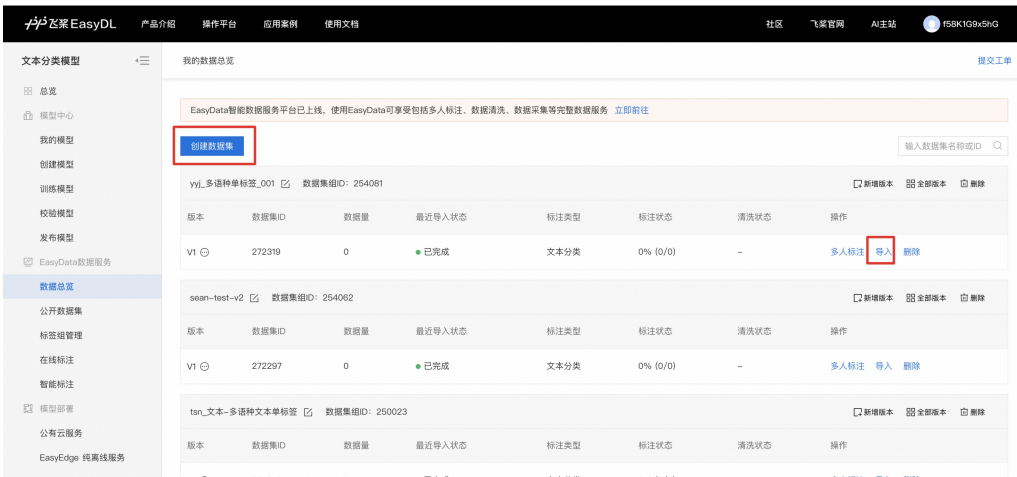
### 文心·NLP大模型

文心·NLP大模型已接入EasyDL文本技术方向。通过下方简单几步即可上手使用。

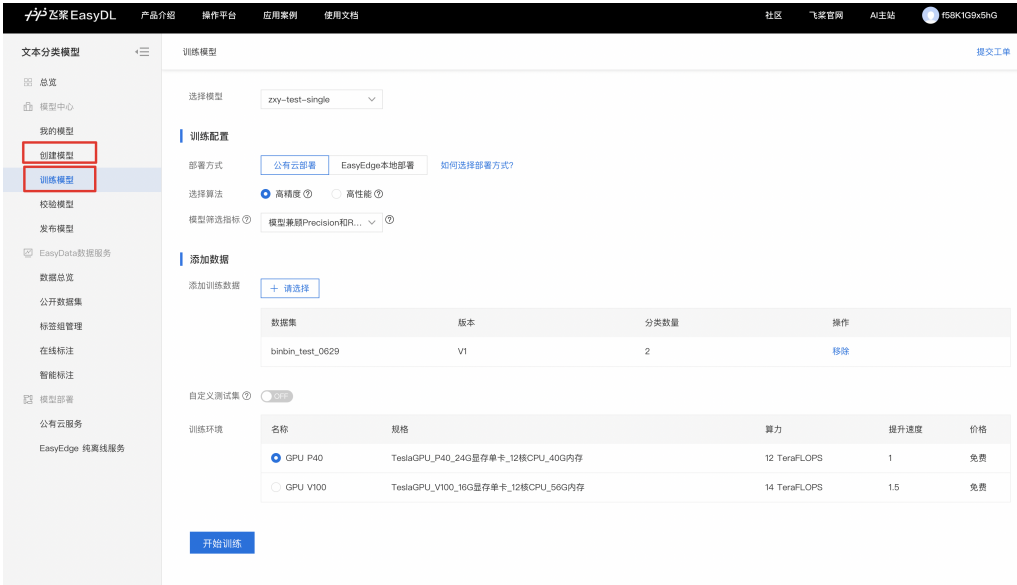
①在EasyDL官网选择一个文本方向的模型类型，以文本分类-单标签为例



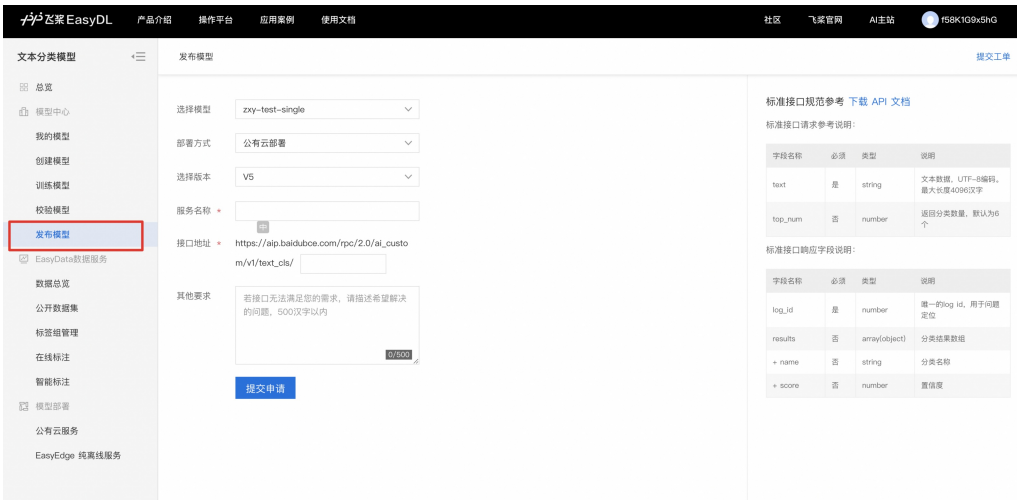
②创建数据集-导入文本数据，完成数据标注



③创建模型-完成训练配置-开始训练，此时将会以文心大模型为基座开始训练模型



④完成训练后，将模型发布为公有云服务接口，即可参考API文档调用服务



☞ 文心·CV大模型

文心·CV大模型即将接入，敬请期待！

☞ 文心·跨模态大模型

文心·跨模态大模型即将接入，敬请期待！

# 新手指南

## EasyDL图像-图像分类快速开始

## 目录

1. [模型示例说明](#)
2. [实现步骤](#)
3. [产品特点](#)
4. [更多参考](#)

### 🔗 模型示例说明

图像分类模型主要用于识别一张图中是否是某类物体/状态/场景，是和图片中主体或状态单一的场景。本文以猫狗识别模型为示例演示图像分类模型训练全过程。

### 🔗 实现步骤

只需四步即可完成自定义AI模型的训练及发布的全过程。

#### 🔗 Step1：成为百度AI开放平台的开发者

要使用百度EasyDL的模型训练能力首先需要注册成为百度AI开放平台的开发者，首先让我们用5分钟来注册百度AI开放平台的开发者（如您已经是开发者，可直接登录使用）

先点击此处[注册百度账号](#)进入，如下图的页面快速的注册一个百度账号吧。

#### 🔗 Step2：提前准备训练数据

图像分类需要提供包含不同类别的图片并标注图片即可训练图像分类模型，自动识别图中是否包含某类物体/状态/场景，下面我们来看看这次训练所需的猫狗图片示例：





图片数量越多理论上训练效果越好，图像分类的图片数量建议每个类别不低于20张图片。

Step3 : 使用EasyDL训练图像分类模型

**创建模型**

进入[EasyDL官方平台](#)点击【立即使用】



点击【图像分类】进入操作台



在模型中心下点击【创建模型】



模型信息填写完成后点击【下一步】

模型列表 > 创建模型



模型创建完成后可在【我的模型】栏查看已创建的模型信息



### 创建数据集

在数据总览界面点击【创建数据集】



在数据集创建界面输入数据集名称后点击【完成】

我的数据总览 &gt; 创建数据集

数据集名称

数据类型

数据集版本

标注类型 

标注模板

数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面

我的数据总览 提交工单

EasyData智能数据服务平台已上线，使用EasyData可享受包括多人标注、数据清洗、数据采集等完整数据服务 [立即前往](#)

[创建数据集](#) 我的数据集

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	192301	0	● 已完成	图像分类	0% (0/0)	-	多人标注 <input type="button" value="导入"/> <input type="button" value="删除"/> <input type="button" value="共享"/>

数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息-本地导入-上传图片的导入方式示例，其余各类型导入方式可参考[图像分类页面上传数据集并在线标注](#)

### 导入数据

数据标注状态  无标注信息  有标注信息

选择数据标注状态及导入方式后点击【上传图片】

导入方式

上传图片

**注意：上传图片时，一定要注意格式要求！**

上传图片 ×

对同一数据集存在多个内容完全一致的图片，将会做去重处理。  
为保证模型训练效果，所上传的图片应与实际业务场景的图片（光线、角度、采集设备）尽可能一致。



**添加图片**

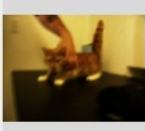
按住command可选多个文件


1. 图片类型为jpg/png/bmp/jpeg，单次上传限制100个文件
2. 图片大小限制在4M内，长宽比在3:1以内，其中最长边需要小于4096px，最短边需要大于30px
3. 您的账户下图片数据集大小限制为10万张图片，如果需要提升数据额度，可在平台提交工单


开始上传


选择好图片后，点击【开始上传】

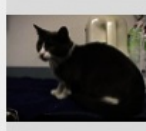
上传图片 ×


  
删除


  
删除


  
删除


  
删除

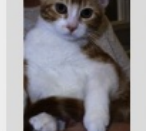
  
删除

  
删除

  
删除

  
删除

  
删除

  
删除

开始上传 继续添加

上传完成后，点击【确认并返回】跳转至数据总览页

### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式 本地导入 ↓ 上传图片 ↓

上传图片 ↑ 上传图片 已上传25个文件

确认并返回

在数据总览页可看到所建数据集，图片上传到平台，需要一段时间，等待片刻刷新页面后，待状态由【正在导入】转为【已完成】即为导入成功。

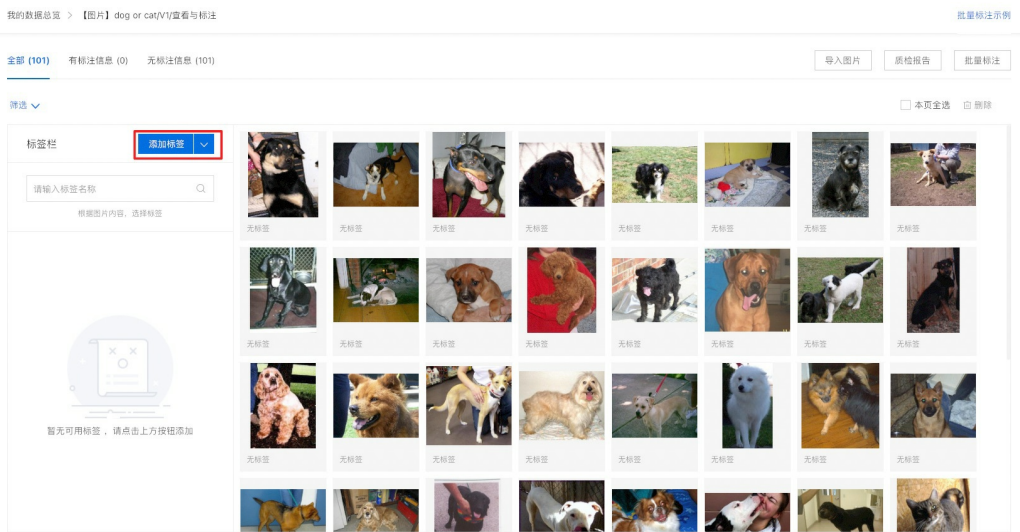


### 数据标注

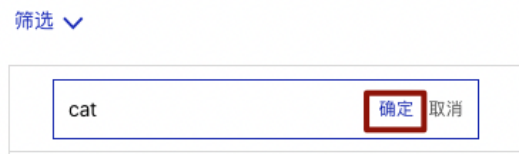
在数据总览页找到需要标注的数据集，点击【查看与标注】，跳转至标注页面



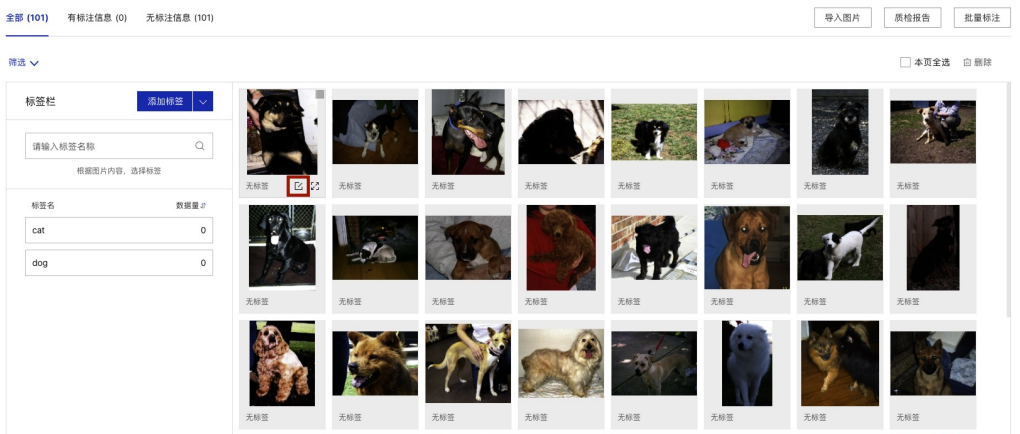
在左侧标签栏下，点击【添加标签】创建数据集标签



分别输入dog、cat并点击【确认】添加数据标签

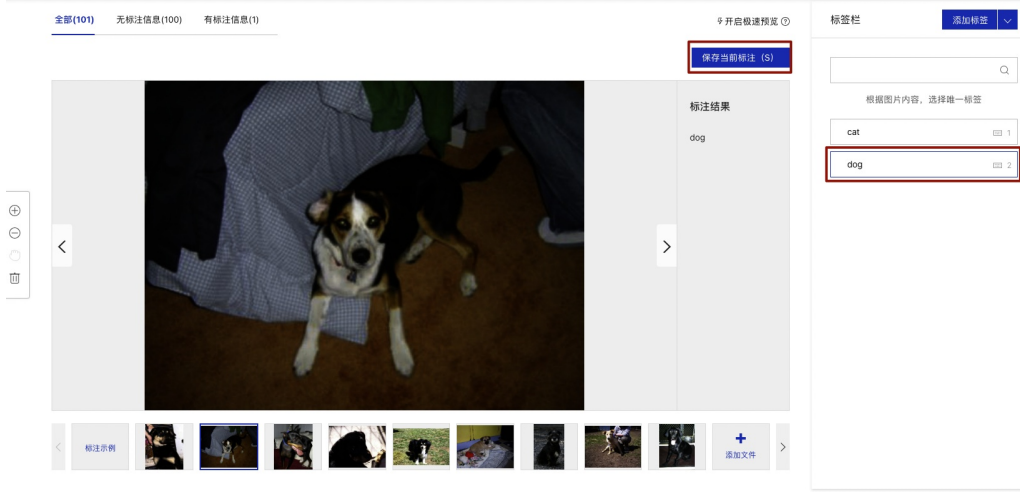


点击图片右下角红框内图标进入到数据标注界面



在当前图片下选择右侧标签栏内的某一类别，代表为图片打上相应的标签，点击【保存当前图片】或直接点击下一张图片图标，在保存标注结果后自动跳转至下一张。标注完所有图片后，该数据集便可用于后续模型训练

我的数据总览 > 【图片】dog or cat/V1/查看与标注 > 标注



### 模型训练

数据集准备完成后，点击【训练】，进入模型训练配置阶段



根据需求选择模型各项配置后，添加训练数据集，点击【开始训练】



在模型列表下，可以看到处于训练状态的模型，将鼠标放置感叹号图标处，即可查看训练进度，同时若勾选短信提醒，在模型训练完成后会以短信的形式通知



### 模型校验

模型训练完成后，可在模型列表下，点击【校验】



模型列表 操作文档 教学视频 常见问题 提交工单

**创建模型**

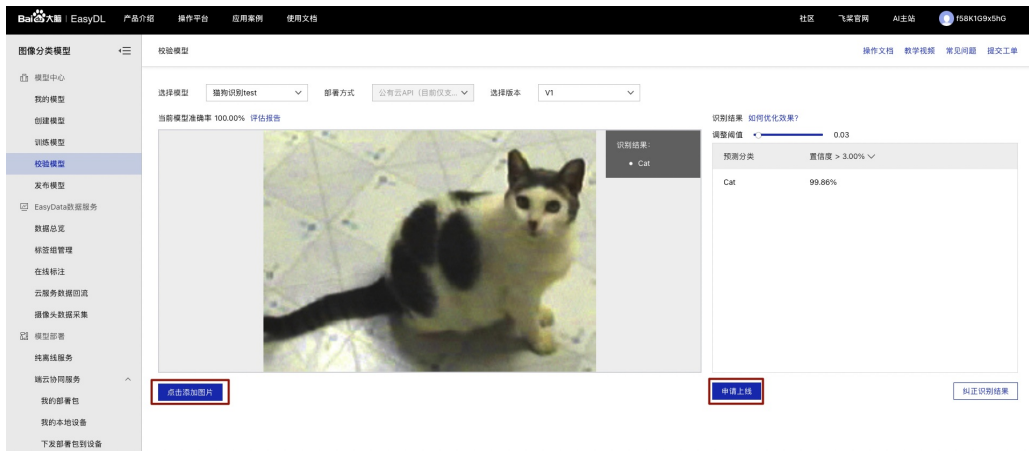
【图像分类】猫狗识别test 模型ID: 120920 训练 历史版本 删除

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	未发布	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果	查看成本配置 申请发布 <b>校验</b>

点击【启动模型校验服务】，需等待几分钟



点击【添加图片】，进行模型校验



在此处可以点击【申请上线】，进行模型发布，跳转到[模型发布](#)

### 模型发布

模型训练完成后，点击【申请发布】

模型列表 操作文档 教学视频 常见问题 提交工单

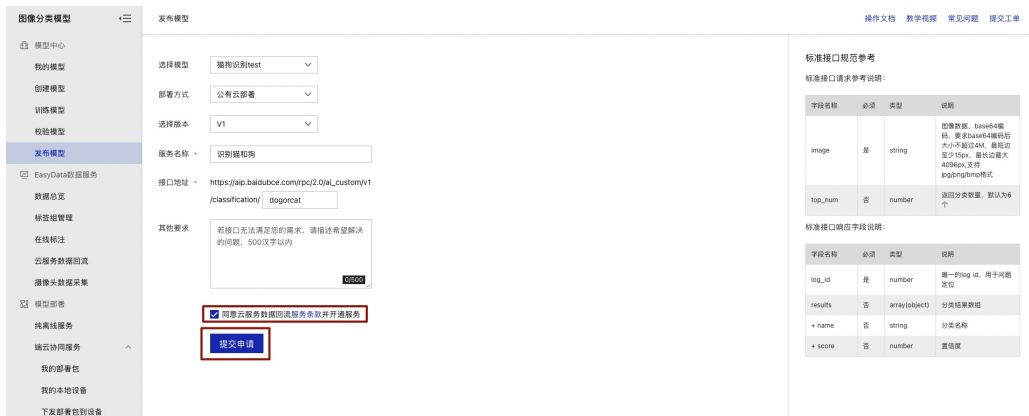
**创建模型**

【图像分类】猫狗识别test 模型ID: 120920 训练 历史版本 删除

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	未发布	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果	查看成本配置 <b>申请发布</b> 校验

按要求填写相应信息后，点击【提交申请】

**注：同时可勾选下方【云服务调用数据管理】的服务条款，通过云服务调用数据反馈，可查找公有云服务模型识别错误的的数据，纠正结果并将其加入下一次用于训练模型的数据集，实现训练数据的持续丰富和模型效果的持续优化。**



提交申请后跳转至【我的模型】栏，服务状态变为【发布中】



等待几分钟，此状态就会变为【已发布】，即发布成功



### 体验H5

模型发布成功后可在模型列表页点击【体验H5】进行模型体验



选择体验H5的模型并点击下一步



自定义样式后点击【生成H5】



体验H5 ×

① 体验H5说明 — ② 自定义样式 — ③ 完成



猫狗识别  
识别猫和狗的模型  
开发者jace  
识图一下

名称

模型介绍

开发者署名

H5分享文案  8/50

生成H5

手机扫描生成的二维码即可在手机端体验模型效果

体验H5 ×

① 体验H5说明 — ② 自定义样式 — ③ 完成

用百度或微信APP扫以下二维码，在手机端体验模型效果



修改已配置的H5页面

完成

#### Step4：模型调用

在 EasyDL“我的模型”列表页，点击【服务详情】后，会得到接口地址

模型列表 <span style="float: right;">操作文档 教学视频 常见问题 提交工单</span>					
自建模型					
【图像分类】猫狗识别test <span style="float: right;">模型ID: 120920</span> <span style="float: right;">训练 历史版本 删除</span>					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果	<a href="#">查看版本配置</a> <span style="border: 1px solid red; padding: 2px;">服务详情</span> <a href="#">投验</a> <a href="#">体验H5</a>

此接口地址在模型调用代码中会用到。点击【立即使用】

服务详情



服务名称: 猫狗识别

模型版本: V1

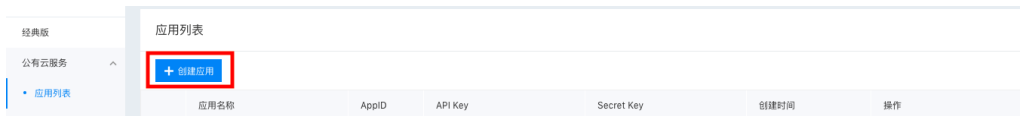
接口地址: https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/classification/dogorcat  
test

服务状态: 已发布

立即使用

查看API文档

需要登录EasyDL控制台中创建一个应用, 点击【创建应用】



填写信息后点击【立即创建】

\* 应用名称:

猫狗识别test

\* 接口选择:

部分接口免费额度还未领取, 请先去领取再创建应用, 确保应用可以正常调用 去领取  
勾选以下接口, 使此应用可以请求已勾选的接口服务, 注意EasyDL图像服务已默认勾选并不可取消。

您已开通付费的接口为: 语音识别-中文普通话、语音识别-粤语、语音识别极速版、实时语音识别-中文普通话、实时语音识别-英文、通用文字识别(高精度版)、通用文字识别(高精度含位置版)、网络图片文字识别、身份证识别、银行卡识别、驾驶证识别、数字识别、手写字识别、火车票识别、iOCR通用版、H5视频活体检测、驾驶行为分析、人脸融合、中文词向量表示、词义相似度、文章分类、内容审核平台-图像、内容审核平台-文本、知识问答、通用物体和场景识别高级版、相同图检索-入库、相似图搜索-入库、手部关键点识别、货架拼接-货架拼接、黑白图像上色、人脸实名认证、人脸实名认证

- EasyDL
- 语音技术
- 文字识别
- 人脸识别
- 自然语言处理
- 内容审核 !
- UNIT !
- 知识图谱
- 图像识别 !
- 智能呼叫中心
- 图像搜索
- 人体分析
- 图像增强与特效
- 智能创作平台
- EasyMonitor
- BML
- 机器翻译

\* 应用描述:

可以快速识别猫和狗的模型

立即创建后, 在应用列表页即可得到 AK SK 密钥

应用列表

应用名称	AppID	API Key	Secret Key	创建时间	操作
1 猫狗识别test	24305311	IMq7VTINNoGqUxrlidahP1fq	..... 显示	2021-06-03 20:16:00	报表 管理 删除

通过使用在线API测试所训练的模型效果

```
import sys
import time
import socket
import json
import base64
import requests
from datetime import datetime

print(datetime.now())
domain = "aip.baidubce.com"
myaddr = socket.getaddrinfo(domain, 'https')
print(str(domain) + " = " + myaddr[0][4][0])

start = time.time()
appid = 'appid'
client_id = 'AK'
client_secret = 'SK'
host = 'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials' + "&client_id=%s&client_secret=%s" % (client_id, client_secret)

session = requests.Session()
response = session.get(host)
access_token = response.json().get("access_token")

request_url = "【模型信息-服务详情-接口地址】"
with open('image.jpg', 'rb') as f:
    image = base64.b64encode(f.read()).decode('UTF8')
headers = {
    'Content-Type': 'application/json'
}
params = {
    "image": image
}
request_url = request_url + "?access_token=" + access_token
response = session.post(request_url, headers=headers, json=params)
content = response.content.decode('UTF-8')
print(json.loads(content))
end = time.time()
print('耗时时长: %1.2f s' % (end-start))
```

## 🔗 产品特点

**可视化操作:** 无需机器学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

**高精度效果:** EasyDL底层结合百度 AutoDL/AutoML技术，针对用户数据自动获得最优网络和超参组合，基于少量数据就能获得出色效果和性能

**端云结合:** 训练完成的模型可发布为云端API或离线SDK，灵活适配各种使用场景及运行环境

**数据支持:** 全方位支持训练数据的高质量采集与高效标注，支持在模型迭代过程中不断扩充数据，助力提升模型效果

## 🔗 更多参考

如对文档说明有疑问或建议，请[微信搜索“BaiduEasyDL”](#)添加小助手交流

备注：文档如使用中遇到报错等问题，请在控制台中通过“工单”联系我们，售后团队为您及时解决问题

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[EasyDL软硬一体方案](#)

[EasyDL应用案例](#)

## EasyDL图像-物体检测快速开始

### 目录

#### 1. 场景介绍

2. 实现步骤
3. 产品特点
4. 更多参考

#### 🔗 场景介绍

在工业场景下，零件分拣、零件计数是在工业生产过程中的一个常见业务需求，由于零件样式多样，市面上没有现成的零件识别服务可以直接使用，往往需要定制企业零件专用的图像识别能力。某个工业质检领域的服务商收到甲方需求，希望能在工厂中实现检测螺丝和螺母，并借助机械臂等装置协助人工完成自动分拣。该服务商经过详细了解需求，发现螺丝与螺母由于排列密集且没有规律，如果要配合机械臂完成就需要精准定位出图片中每个零件的名称及位置。经过调研，由于市场上并没有任何一家公司有现成的螺丝螺母识别服务，同时该服务商缺少相关AI算法工程师及算力资源，如果筹备相关技术及资源成本较高，成为该服务商面临的一大难题。无意中了解到百度EasyDL可以灵活定制并可以快速上手获得业务所需的高精度AI能力，刚好可以解决该服务商面临的问题。

#### 🔗 实现步骤

只需四步即可完成自定义AI模型的训练及发布的全过程。

#### 🔗 Step1：成为百度AI开放平台的开发者

要使用百度EasyDL的模型训练能力首先需要注册成为百度AI开放平台的开发者，首先让我们用5分钟来注册百度AI开放平台的开发者（如您已经是开发者，可直接登录使用）。

先点击此处[注册百度账号](#)进入，如下图的页面快速的注册一个百度账号吧。

#### 🔗 Step2：提前准备训练数据

物体检测需要提供包含目标物体的图片并标注物体即可训练物体检测模型，自动识别图中所有目标物体的位置、名称，下面我们来看看这次需要计数的包含螺丝螺母的图片示例：



图片数量越多理论上训练的效果就越好，物体检测的图片数据建议不低于20张图片 注意图片需要为业务生产的真实环境所采集的图片，与真实场景越贴近，训练模型效果越佳

Step3：使用EasyDL训练物体检测模型

### 创建模型

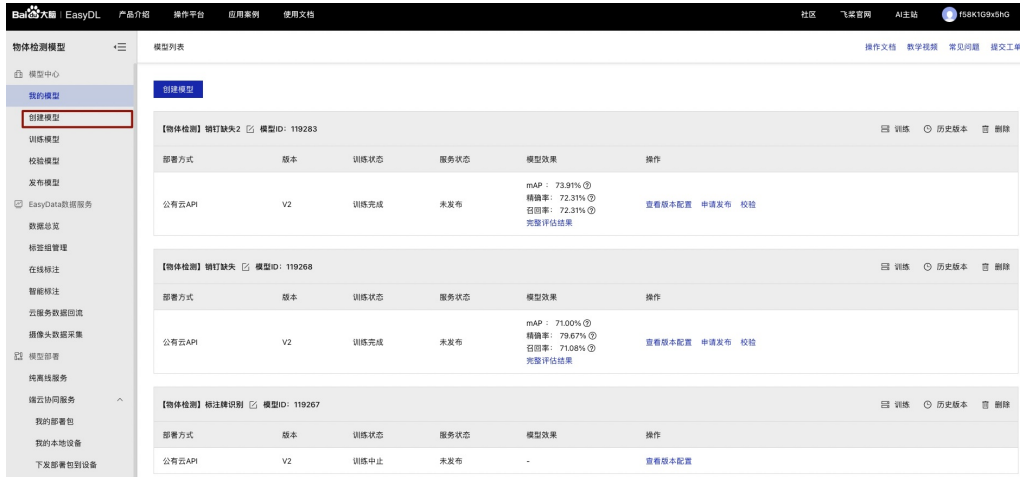
进入[EasyDL官方平台](#) 点击【立即使用】



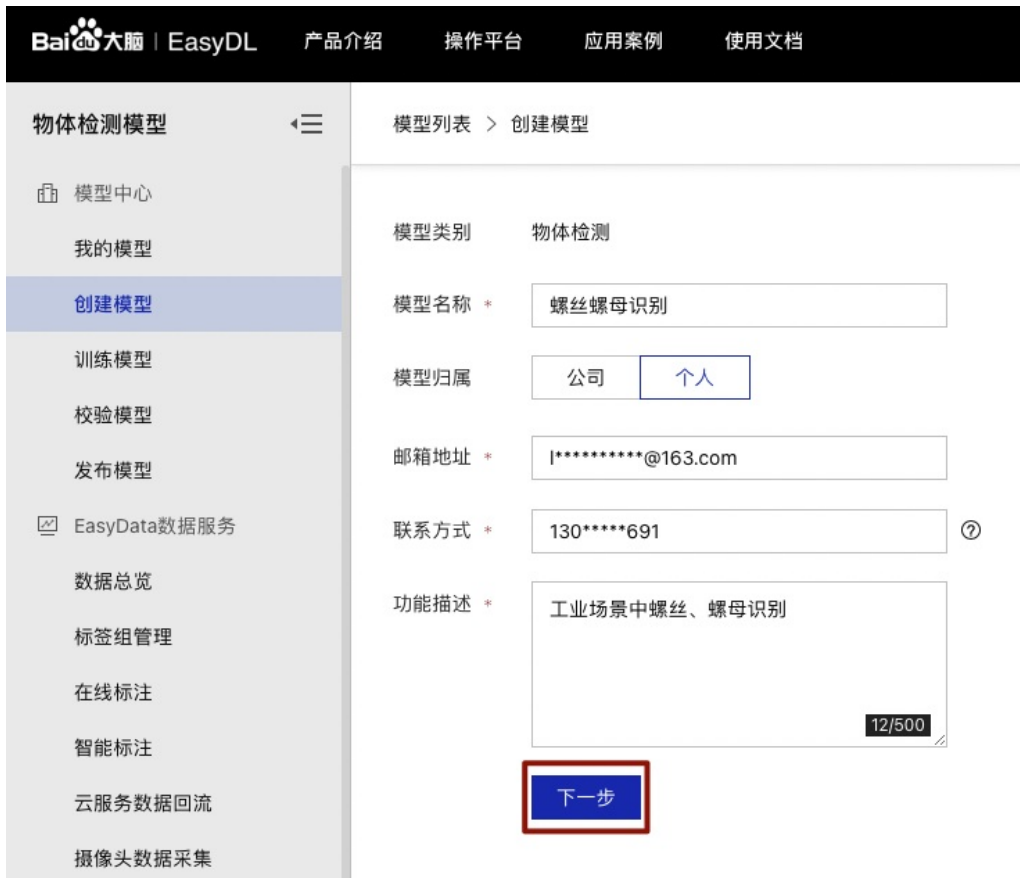
点击【物体检测】，进入操作台



在模型列表下点击【创建模型】



填写模型信息后，点击【下一步】



模型创建完成后可在【我的模型】栏查看已创建的模型信息



### 创建数据

在数据总览界面点击【创建数据集】





在数据集创建界面输入数据集名称后点击【完成】

我的数据总览 > 创建数据集



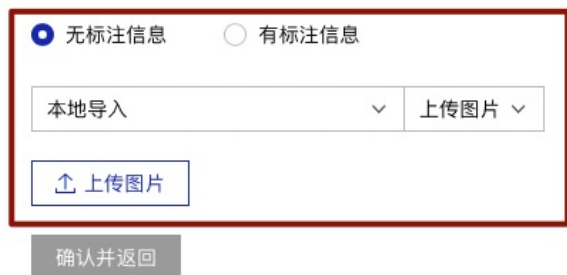
数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面



数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息-本地导入-上传图片的导入方式示例，其余各类型导入方式可参考[页面上传物体检测数据集并在线标注](#)

### 1 导入数据

数据标注状态  
 选择数据标注状态及导入方式后点击【上传图片】  
 上传方式  
 上传图片



注意：上传图片时，一定要注意格式要求！

上传图片 ×

对同一数据集存在多个内容完全一致的图片，将会做去重处理。  
为保证模型训练效果，所上传的图片应与实际业务场景的图片（光线、角度、采集设备）尽可能一致。



**添加图片**

按住command可选多个文件

1. 图片类型为jpg/png/bmp/jpeg，单次上传限制100个文件
2. 图片大小限制在4M内，长宽比在3:1以内，其中最长边需要小于4096px，最短边需要大于30px
3. 您的账户下图片数据集大小限制为10万张图片，如果需要提升数据额度，可在平台提交工单

开始上传

选择好图片后，点击【开始上传】

上传图片 ×

				
删除	删除	删除	删除	删除
				
删除	删除	删除	删除	删除

开始上传

继续添加

上传完成后，点击【确认并返回】跳转至数据总览页

### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式 本地导入 上传图片

上传图片 ↑ 上传图片 已上传25个文件

确认并返回

在数据总览页可看到所建数据集，图片上传到平台，需要一段时间，等待片刻刷新页面后，待状态由【正在导入】转为【已完成】即为导入成功。



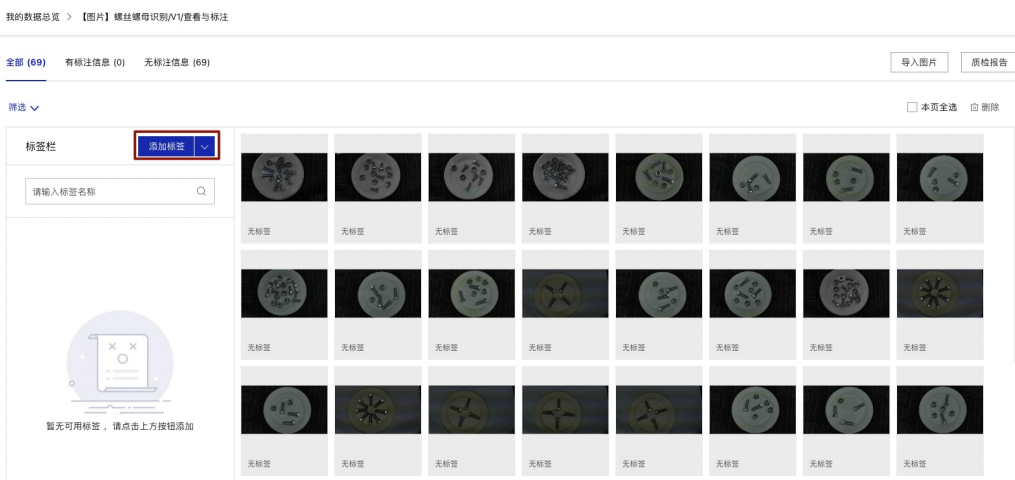


### 数据标注

在数据总览页找到需要标注的数据集，点击【查看与标注】，跳转至标注页面



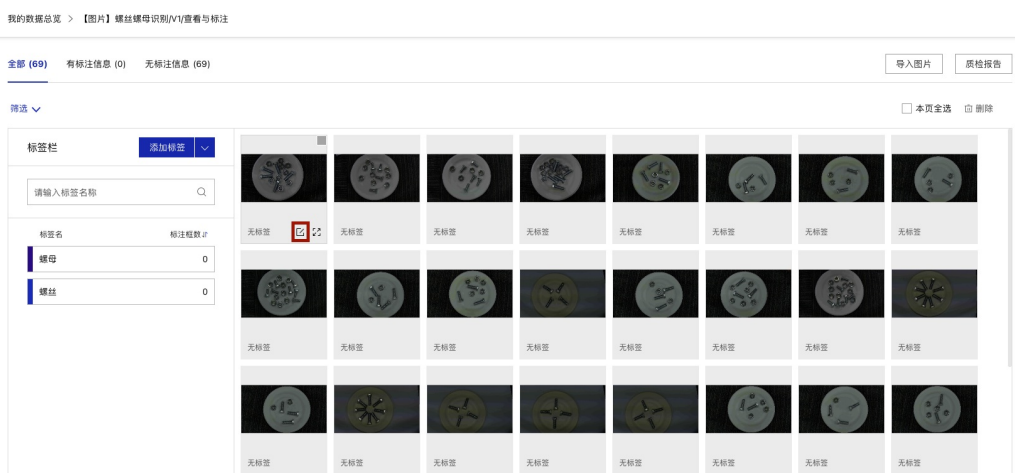
在左侧标签栏下，点击【添加标签】创建数据集标签



分别输入需要创建的标签名称并点击【确认】添加数据标签

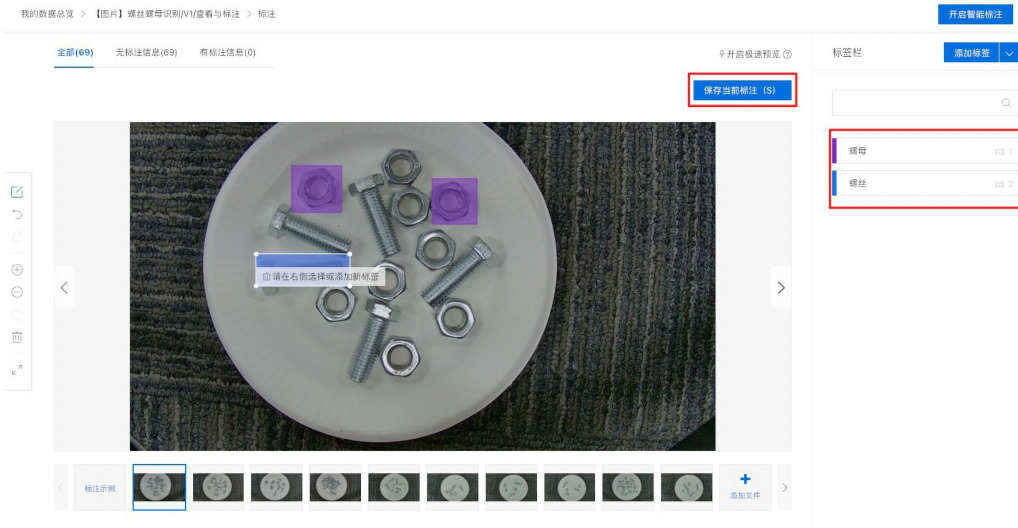


点击图片右下角红框内图标进入到数据标注界面



在当前图片下分别框选出图片中的目标物体并添加标签，点击【保存当前图片】或直接点击下一张图片图标，在保存标注结果后自动跳转至下一张。

标注完所有图片后，该数据集便可用于后续模型训练



### 模型训练

数据集准备完成后，点击【训练】，进入模型训练配置阶段



根据需求选择模型各项配置后，添加训练数据集，点击【开始训练】



在模型列表下，可以看到处于训练状态的模型，将鼠标放置感叹号图标处，即可查看训练进度，同时若勾选短信提醒，在模型训练完成后会以短信的形式通知



### 模型校验

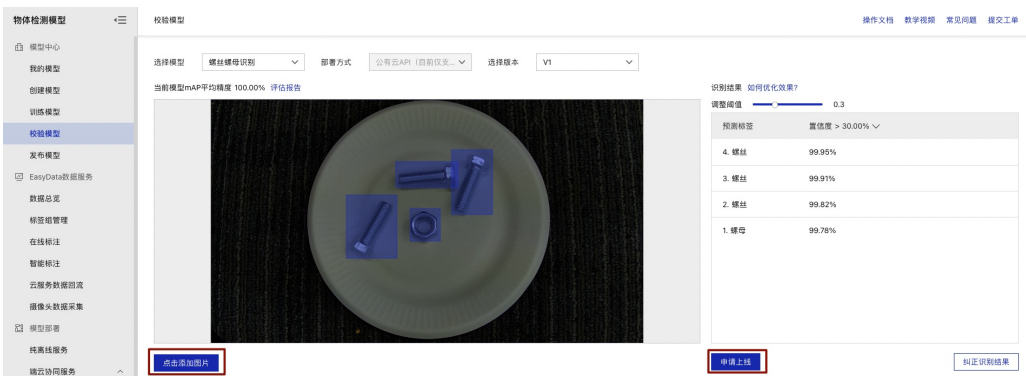
模型训练完成后，可在模型列表下，点击【校验】



点击【启动模型校验服务】，需等待几分钟



点击【添加图片】，进行模型校验



在此处可以点击【申请上线】，进行模型发布，跳转到模型发布

### 模型发布

模型训练完成后，点击【申请发布】



按要求填写相应信息后，点击【提交申请】

注：同时可勾选下方【云服务调用数据管理】的服务条款，通过云服务调用数据反馈，可查找公有云服务模型识别错误的的数据，纠正结果并将其加入下一次用于训练模型的数据集，实现训练数据的持续丰富和模型效果的持续优化。

## 发布模型

选择模型

部署方式

选择版本

服务名称 \*

接口地址 \*

其他要求

0/500

同意云服务数据回流服务条款并开通服务

提交申请

提交申请后跳转至【我的模型】栏，服务状态变为【发布中】

模型列表 操作文档 教学视频 常见问题 提交工单

创建模型

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	发布中	mAP : 100.00% 精确率 : 100.00% 召回率 : 100.00%	查看版本配置 校验 完整评估结果

等待几分钟，此状态就会变为【已发布】，即发布成功

图像分类模型 操作文档 教学视频 常见问题 提交工单

模型列表

我的模型

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	top1准确率 : 100.00% top5准确率 : 100.00% 完整评估结果	查看版本配置 服务详情 校验 体验H5

## 体验H5

模型发布成功后可在模型列表页点击【体验H5】进行模型体验

模型列表 操作文档 教学视频 常见问题 提交工单

创建模型

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	mAP : 100.00% 精确率 : 100.00% 召回率 : 100.00%	查看版本配置 服务详情 校验 体验H5

选择体验H5的模型并点击下一步

体验H5

×

① 体验H5说明 ——— ② 自定义样式 ——— ③ 完成

- H5中的商品检测功能将使用你的APP进行调用。
- 每次体验检测将消耗个人帐号下的调用次数

调用APP: 螺丝螺母识别-APPID: 24311653

下一步

自定义样式后点击【生成H5】

体验H5

×

① 体验H5说明 ——— ② 自定义样式 ——— ③ 完成



名称

模型介绍

开发者署名

H5分享文案

生成H5

手机扫描生成的二维码即可在手机端体验模型效果

体验H5

×

① 体验H5说明 ——— ② 自定义样式 ——— ③ 完成

用百度或微信APP扫以下二维码，在手机端体验模型效果



[修改已配置的H5页面](#)

完成

在 EasyDL“我的模型”列表页，点击【**服务详情**】后，会得到接口地址

模型列表 操作文档 教学视频 常见问题 提交工单

创建模型

【物体检测】螺丝螺母识别 模型ID: 120958 <span style="float: right;">训练 历史版本 删除</span>					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	mAP: 100.00% 精确率: 100.00% 召回率: 100.00% 完整评估结果	查看版本配置 <b>服务详情</b> 校验 体验H5

此接口地址在模型调用代码中会用到。点击【**立即使用**】

### 服务详情 ×

服务名称: 识别螺丝和螺母

模型版本: V1

**接口地址: [https://aip.baidubce.com/rpc/2.0/ai\\_custom/v1/detection/jacetest111](https://aip.baidubce.com/rpc/2.0/ai_custom/v1/detection/jacetest111)**

服务状态: 已发布

**立即使用** 查看API文档

需要登录EasyDL控制台中创建一个应用，点击【**创建应用**】

经典版

应用列表

公有云服务 + 创建应用

应用列表

应用名称	AppID	API Key	Secret Key	创建时间	操作
------	-------	---------	------------	------	----

填写信息后点击【**立即创建**】

## 创建新应用

\* 应用名称:

螺丝螺母识别

\* 接口选择:

部分接口免费额度还未领取, 请先去领取再创建应用, 确保应用可以正常调用 [去领取](#)

勾选以下接口, 使此应用可以请求已勾选的接口服务, 注意EasyDL图像服务已默认勾选并不可取消。

您已开通付费的接口为: 短语音识别-中文普通话、短语音识别-粤语、短语音识别极速版、实时语音识别-中文普通话、实时语音识别-英文、通用文字识别(高精度版)、通用文字识别(高精度含位置版)、网络图片文字识别、身份证识别、银行卡识别、驾驶证识别、数字识别、手写文字识别、火车票识别、iOCR通用版、H5视频活体检测、驾驶行为分析、人脸融合、中文词向量表示、词义相似度、文章分类、内容审核平台-图像、内容审核平台-文本、知识问答、通用物体和场景识别高级版、相同图检索-入库、相似图搜索-入库、手部关键点识别、货架拼接-货架拼接、黑白图像上色、人脸实名认证、人脸实名认证

- EasyDL
- 语音技术
- 文字识别
- 人脸识别
- 自然语言处理
- 内容审核 !
- UNIT !
- 知识图谱
- 图像识别 !
- 智能呼叫中心
- 图像搜索
- 人体分析
- 图像增强与特效
- 智能创作平台
- EasyMonitor
- BML
- 机器翻译

\* 应用描述:

可快速识别螺丝和螺母的模型

立即创建后, 在应用列表页即可得到 AK SK 密钥

应用列表

+ 创建应用						
应用名称	AppID	API Key	Secret Key	创建时间	操作	
1 螺丝螺母识别	24311653	qnMuy5xH8q3SqK4GDjovbPI	***** 显示	2021-06-04 16:16:16	报表 管理 删除	

通过使用在线API测试所训练的模型效果



```

import sys
import time
import socket
import json
import base64
import requests
from datetime import datetime

print(datetime.now())
domain = "aip.baidubce.com"
myaddr = socket.getaddrinfo(domain, 'https')
print(str(domain) + " = " + myaddr[0][4][0])

start = time.time()
appid = 'appid'
client_id = 'AK'
client_secret = 'SK'
host = 'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials' host += "&client_id=%s&client_secret=%s" %
(client_id, client_secret)

session = requests.Session()
response = session.get(host)
access_token = response.json().get("access_token")

request_url = "【模型信息-服务详情-接口地址】"
with open("image.jpg", 'rb') as f:
    image = base64.b64encode(f.read()).decode('UTF8')
headers = {
    'Content-Type': 'application/json'
}
params = {
    "image": image
}
request_url = request_url + "?access_token=" + access_token
response = session.post(request_url, headers=headers, json=params)
content = response.content.decode('UTF-8')
print(json.loads(content))
end = time.time()
print('耗时时长: %1.2f s'% (end-start))

```

## 🔗 产品特点

**可视化操作:** 无需机器学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

**高精度效果:** EasyDL底层结合百度 AutoDL/AutoML技术，针对用户数据自动获得最优网络和超参组合，基于少量数据就能获得出色效果和性能

**端云结合:** 训练完成的模型可发布为云端API或离线SDK，灵活适配各种使用场景及运行环境

**数据支持:** 全方位支持训练数据的高质量采集与高效标注，支持在模型迭代过程中不断扩充数据，助力提升模型效果

## 🔗 更多参考

如对文档说明有疑问或建议，请[微信搜索“BaiduEasyDL”](#)添加小助手交流

备注：文档如使用中遇到报错等问题，请在控制台中通过“工单”联系我们，售后团队为您及时解决问题

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[EasyDL软硬一体方案](#)

[EasyDL应用案例](#)

## EasyDL文本-文本分类单标签快速开始

### 目录

1. [场景介绍](#)
2. [实现步骤](#)

### 3. 产品特点

### 4. 更多参考

#### 🔗 场景介绍

目前不少互联网内容平台或者电商平台中有用户评论模块，往往都需要人工维护评论信息，如将评论信息中好的评论与坏的评论进行分类，或者将评论信息中的广告信息能有效过滤/甄别出来，当评论内容越来越高时，人工维护评论的成本就越高。而越来越多的垂直内容平台由于评论信息内容多样，内容不一，为实现更好的文本分类效果，往往需要定制企业专属的文本分类能力。

某个酒店信息聚合平台，希望在其官网的评论模块中增加自动分类功能，能支持将好的评价和不好的评价自动分类，方便酒店管理者在后台查看，同时该网站缺少相关AI算法工程师及算力资源，如何能更高效更低成本的获取定制文本分类服务，成为该服务商面临的一大难题。无意中了解到百度EasyDL可以灵活定制并可以快速上手获得业务所需的高精度AI能力，刚好可以解决该服务商面临的问题。

#### 🔗 实现步骤

只需四步即可完成自定义AI模型的训练及发布的全过程。

##### 🔗 Step1：成为百度AI开放平台的开发者

要使用百度EasyDL的模型训练能力首先需要注册成为百度AI开放平台的开发者，首先让我们用5分钟来注册百度AI开放平台的开发者（如您已经是开发者，可直接登录使用）。

先点击此处[注册百度账号](#)进入，如下图的页面快速的注册一个百度账号吧。

##### 🔗 Step2：提前准备训练数据

文本分类单标签可实现文本内容的自动分类，为每个文本定义一个标签类型。数据数量越多理论上训练效果越好,本次示例以线上公开数据集为例，具体数据处理全过程可参考[文本分类数据集创建](#)

##### 🔗 Step3：使用EasyDL训练文本分类

#### 创建模型

进入[EasyDL官方平台](#) 点击【[立即使用](#)】



点击【文本分类-单标签】，进入操作台



在模型列表下点击【创建模型】



填写模型信息后，点击【下一步】

## 模型列表 &gt; 创建模型

模型类别 文本分类-单标签

任务场景 \* 短文本分类任务 ?

模型名称 \* 酒店评分

模型归属  公司  个人

邮箱地址 \* 1\*\*\*\*\*@163.com

联系方式 \* 182\*\*\*\*\*572 ?

功能描述 \* 识别酒店评价好坏的模型 11/500

**下一步**

模型创建完成后可在【我的模型】栏查看已创建的模型信息

模型列表 提交工单

**创建模型**

【文本分类】酒店评分  模型ID: 121263 训练 删除

模型创建成功。您可以选择使用公开数据集训练模型体验分类效果，或创建数据集使用自有数据进行专属分类模型的训练。模型训练后，可以在此处查看模型的最新版本和状态

## 选择数据集

在模型列表页找到新建的模型并选择【公开数据集】查看现有公开数据集

模型列表 提交工单

**创建模型**

【文本分类】酒店评分  模型ID: 121263 训练 删除

模型创建成功。您可以选择使用公开数据集训练模型体验分类效果，或创建数据集使用自有数据进行专属分类模型的训练。模型训练后，可以在此处查看模型的最新版本和状态

## 本次训练将使用chnsenticorp情感分类-评测数据集

文本分类模型 提交工单

公开数据集

chnsenticorp-情感分类-训练数据集							
版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	20000006	8249	● 已完成	文本分类	100% (8249/8249)	-	查看

chnsenticorp-情感分类-评测数据集							
版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	20000001	1178	● 已完成	文本分类	100% (1178/1178)	-	查看

emotion							
版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	20000000	2710	● 已完成	文本分类	100% (2710/2710)	-	查看

在模型列表页点击【模型训练】进入到数据集选择界面

文本分类模型 提交工单

模型列表

**创建模型**

【文本分类】酒店评分  模型ID: 121263 训练 删除

模型创建成功。您可以选择使用公开数据集训练模型体验分类效果，或创建数据集使用自有数据进行专属分类模型的训练。**模型训练**，可以在此处查看模型的最新版本和状态

选择chnsenticorp情感分类-评测数据集并勾选全部分类名称，点击【添加】



点击【开始训练】进入到模型训练阶段



在模型列表下，可以看到处于训练状态的模型，将鼠标放置感叹号图标处，即可查看训练进度，同时若勾选短信提醒，在模型训练完成后会以短信的形式通知



### 模型校验

模型训练完成后，可在模型列表下，点击【校验】



点击【启动模型校验服务】，需等待几分钟

校验模型

选择模型  部署方式  选择版本

**启动模型校验服务**

输入校验文本，进行模型校验

校验模型 提交工单

选择模型  部署方式  选择版本

当前模型准确率 94.63% [评估报告](#)

请输入校验的文本，或 [点击上传文](#) 支持文本格式: txt, 文本长度上限为512汉字 (字符)

这个酒店的环境很一般，服务态度更不太好

10/512

**校验**

识别结果 如何优化效果?

调整阈值  置信度 > 3.00%

预测分类	置信度
0	94.68%
1	5.32%

**申请上线** 纠正识别结果

在此处可以点击【**申请上线**】，进行模型发布，跳转到[模型发布](#)

模型发布

模型训练完成后，点击【**申请发布**】

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V4	训练完成	未发布	准确率: 89.15% F1-score: 0.891 完整评估结果	<a href="#">查看版本配置</a> <b>申请发布</b> <a href="#">校验</a>

按要求填写相应信息后，点击【**提交申请**】

发布模型

选择模型

部署方式

选择版本

服务名称 \*

接口地址 \*

其他要求

0/500

**提交申请**

提交申请后跳转至【**我的模型**】栏，服务状态变为【**发布中**】

【文本分类】酒店评分 模型ID: 76016					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V4	训练完成	发布中	准确率: 89.15% F1-score: 0.891 完整评估结果	查看版本配置 校验

等待几分钟，此状态就会变为【已发布】，即发布成功

模型列表 提交工单

[创建模型](#)

【文本分类】酒店评分 模型ID: 121263					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	准确率: 94.63% F1-score: 0.946 完整评估结果	查看版本配置 服务详情 校验

### Step4: 模型调用

在【我的模型】栏找到发布完成的模型点击【服务详情】

文本分类模型 模型列表 提交工单

[创建模型](#)

模型中心

**我的模型**

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

【文本分类】酒店评分 模型ID: 121263					
部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V1	训练完成	已发布	准确率: 94.63% F1-score: 0.946 完整评估结果	查看版本配置 <b>服务详情</b> 校验

接口地址在模型调用代码中将会用到，点击【立即使用】跳转至EasyDL控制台

服务详情 ×

---

服务名称: **jiudianpingfen\_v1**

模型版本: **V1**

接口地址: **https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/text\_cls/jiudianpingfen\_v1**

服务状态: **已发布**

---

立即使用
查看API文档

在EasyDL控制台中点击【创建应用】

EasyDL文本 应用列表

公有云部署 [+ 创建应用](#)

信息填写完成后点击【立即创建】



## 创建新应用

\* 应用名称：

酒店评论识别

\* 接口选择：

部分接口免费额度还未领取，请先去领取再创建应用，确保应用可以正常调用 [去领取](#)

勾选以下接口，使此应用可以请求已勾选的接口服务，注意EasyDL文本服务已默认勾选并不可取消。

您已开通付费的接口为：短语音识别-中文普通话、短语音识别-粤语、短语音识别极速版、实时语音识别-中文普通话、实时语音识别-英文、通用文字识别（高精度版）、通用文字识别（高精度含位置版）、网络图片文字识别、身份证识别、银行卡识别、驾驶证识别、数字识别、手写文字识别、火车票识别、iOCR通用版、H5视频活体检测、驾驶行为分析、人脸融合、中文词向量表示、词义相似度、文章分类、内容审核平台-图像、内容审核平台-文本、知识问答、通用物体和场景识别高级版、相同图检索-入库、相似图搜索-入库、手部关键点识别、货架拼接-货架拼接、黑白图像上色、人脸实名认证、人脸实名认证

- EasyDL
- 语音技术
- 文字识别
- 人脸识别
- 自然语言处理
- 内容审核 !
- UNIT !
- 知识图谱
- 图像识别 !
- 智能呼叫中心
- 图像搜索
- 人体分析
- 图像增强与特效
- 智能创作平台
- EasyMonitor
- BML
- 机器翻译

\* 应用描述：

识别评论内容是好评还是差评

创建完成后即刻在应用列表页获取到AK SK密钥

应用列表

[+ 创建应用](#)

应用名称	AppID	API Key	Secret Key	创建时间	操作
1 酒店评论识别	24313338	LKDc0w2cYfBpYcua843peH	..... 显示	2021-06-04 20:09:18	<a href="#">预览</a> <a href="#">管理</a> <a href="#">删除</a>

在获取到API KEY以及 Secret KEY后，我们就可以写一个示例代码调用我们之前创建并训练完成酒店评论文本分类模型

## 准备开发环境

我们选择用python来快速搭建一个原型，如果没有接下来需要安装以下python。可以参考下表列出的不同操作系统的安装方法进行安装。

Python的官方下载地址：[下载python](#)

系统是Windows	系统是Linux	系统是macOS
<ul style="list-style-type: none"> <li>✓ 在Python官网下载Python</li> <li>✓ 解压并双击exe文件安装</li> </ul>	<ul style="list-style-type: none"> <li>✓ 执行`python --version` 看是否输出python版本；</li> <li>✓ 若回显版本则系统已预装Python；</li> <li>✓ 若无版本则执行`sudo apt-get update &amp; sudo apt-get install python3.6`</li> </ul>	<ul style="list-style-type: none"> <li>✓ macOS一般自带了Python，无需自行安装</li> </ul>

## Windows 快速测试包

windows平台的用户如果对上述的python安装感到困难，可以下载我们的一键测试包，下载地址：[windows测试包](#)。

解压zip文件后，双击run.bat即可测试。

## 编写代码

新建一个 `main.py`

粘贴以下内容，不要忘记替换你的 `API_KEY` 以及 `SECRET_KEY`：

```
##### coding=utf-8

import sys
import json

##### 保证兼容python2以及python3
IS_PY3 = sys.version_info.major == 3
if IS_PY3:
    from urllib.request import urlopen
    from urllib.request import Request
    from urllib.error import URLError
    from urllib.parse import urlencode
    from urllib.parse import quote_plus
else:
    import urllib2
    from urllib import quote_plus
    from urllib2 import urlopen
    from urllib2 import Request
    from urllib2 import URLError
    from urllib import urlencode
    reload(sys)
    sys.setdefaultencoding('utf8')

##### 防止https证书校验不正确
import ssl
ssl._create_default_https_context = ssl._create_unverified_context

##### 百度云控制台获取到ak, sk以及
##### EasyDL官网获取到URL

##### ak
API_KEY = 'RgdpDFjOHmRQvphsi8bLhIYE'

##### sk
SECRET_KEY = 'ja1pDyGaF3vgwPNW3TOEqEkkd5hgl8ug'

##### url
EASYDL_TEXT_CLASSIFY_URL = "https://aip.baidubce.com/rpc/2.0/ai_custom/v1/text_cls/hotel_comment"

""" TOKEN start """
TOKEN_URL = 'https://aip.baidubce.com/oauth/2.0/token'

"""
    获取token
"""
def fetch_token():
    params = {'grant_type': 'client_credentials',
             'client_id': API_KEY,
             'client_secret': SECRET_KEY}
    post_data = urlencode(params)
    if (IS_PY3):
        post_data = post_data.encode('utf-8')
    req = Request(TOKEN_URL, post_data)
    try:
        f = urlopen(req, timeout=5)
        result_str = f.read()
    except URLError as err:
        print(err)
    if (IS_PY3):
        result_str = result_str.decode()

    result = json.loads(result_str)
```

```
if ('access_token' in result.keys() and 'scope' in result.keys()):
    if not 'brain_all_scope' in result['scope'].split(' '):
        print ('please ensure has check the ability')
        exit()
    return result['access_token']
else:
    print ('please overwrite the correct API_KEY and SECRET_KEY')
    exit()

"""
调用远程服务
"""

def request(url, data):
    if IS_PY3:
        req = Request(url, json.dumps(data).encode('utf-8'))
    else:
        req = Request(url, json.dumps(data))

    has_error = False
    try:
        f = urlopen(req)
        result_str = f.read()
        if (IS_PY3):
            result_str = result_str.decode()
        return result_str
    except URLError as err:
        print(err)

if __name__ == '__main__':

    # 获取access token
    token = fetch_token()

    # 拼接url
    url = EASYDL_TEXT_CLASSIFY_URL + "?access_token=" + token

    # 好评
    text_good = "这个酒店不错，干净而且安静，早餐也好吃"

    # 差评
    text_bad = "不怎么干净，服务员态度也差强人意，以后不会在预订了"

    # 请求接口
    # 测试好评
    response = request(url,
        {
            'text': text_good,
            'top_num': 2
        })

    result_json = json.loads(response)

    result = result_json["results"]

    # 打印好评结果
    print(text_good)
    for obj in result:
        print(" 评论类别： " + obj['name'] + " 置信度： " + str(obj['score']))
    print("")

    # 请求接口
    # 测试差评
    response = request(url,
        {
            'text': text_bad,
```

```
        'top_num': 2
    })

    result_json = json.loads(response)

    result = result_json["results"]

    # 打印差评结果
    print(text_bad)
    for obj in result:
        print(" 评论类别：" + obj['name'] + " 置信度：" + str(obj['score']))
```

### 运行代码

在命令行中运行python main.py

您还可以在我们的[github地址](#)中找到main.py

### 结果

若代码正确运行，命令行界面上会显示出运行结果：

```
这个酒店不错，干净而且安静，早餐也好吃
评论类别：good 置信度：0.974235713482
评论类别：bad 置信度：0.0257642995566

不怎么干净，服务员态度也差强人意，以后不会在预订了
评论类别：bad 置信度：0.850781261921
评论类别：good 置信度：0.149218738079
```

结果中返回了每个待分类文本的分类以及置信度，置信度高的分类说明预测的文本属于这个分类的可能越大，这样我们就能将上述酒店评论分为好评，差评了，详细的返回和参数文档需要参照API文档[EasyDL文本分类API参考文档](#)

### 🔗 产品特点

**可视化操作：**无需机器学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

**高精度效果：**EasyDL底层结合百度 AutoDL/AutoML技术，针对用户数据自动获得最优网络和超参组合，基于少量数据就能获得出色效果和性能

**端云结合：**训练完成的模型可发布为云端API或离线SDK，灵活适配各种使用场景及运行环境

**数据支持：**全方位支持训练数据的高质量采集与高效标注，支持在模型迭代过程中不断扩充数据，助力提升模型效果

### 🔗 更多参考

如对文档说明有疑问或建议，请[微信搜索“BaiduEasyDL”](#)添加小助手交流

备注：文档如使用中遇到报错等问题，请在控制台中通过“工单”联系我们，售后团队为您及时解决问题

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[EasyDL软硬一体方案](#)

[EasyDL应用案例](#)

## EasyDL零售行业版快速开始

### 🔗 简介

本文档介绍使用EasyDL零售版商品检测快速训练一个识别可口可乐的商品检测模型，基本流程如下：

- 1.创建模型
- 2.创建SKU
- 3.上传和标注训练数据
- 4.训练模型

## 5.发布模型

## 🔗 步骤1.创建模型

这个步骤将会介绍如何创建模型

## 🔗 进入创建模型页面

在[EasyDL零售版商品检测产品主页](#)点击【开始训练】按钮进入到[模型训练页](#)，下面会出现两种情况：

- 第一种，如果您没有登录百度云，则会跳转到百度云登录页面，没有百度账户的客户请先[注册百度账户](#)。登录后，会跳转到[模型概览页](#)，点击【商品检测】卡片上的【立即定制】按钮，会跳转模型训练页面的创建模型页。
- 第二种，如果您已登录，会直接进入【我的模型】页，该页面能够管理已经创建的模型，点击左侧列表中的【创建模型】进入创建模型页面。

## 🔗 创建模型

进入创建模型页面后你会看到如下图中展示的内容

需要填写的项目如下：

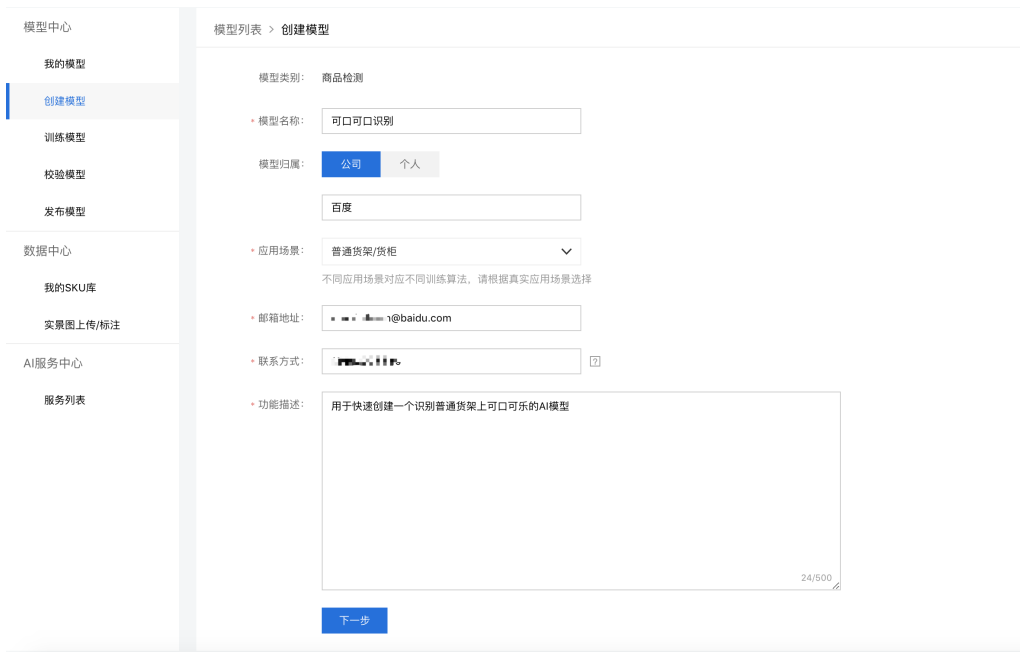
- 模型名称  
模型的名称
- 模型归属  
模型是属于公司的，还是属于个人的，如果是前者，请填写公司名称
- 应用场景

提示：**请根据真实业务应用场景选择**，选择的场景将会关联后端数据增强算法，若不确定，请选择“其他”

可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

- 邮箱地址  
用于联系到您的邮箱地址
- 联系方式  
有效的联系方式将有助于后续模型上线的人工快速审核，以及更快的百度官方支持，推荐填写个人手机号码
- 功能描述  
描述改模型将要应到的业务场景，详细的描述，在获取官方支持时，能帮助我们为您提供准确的使用建议

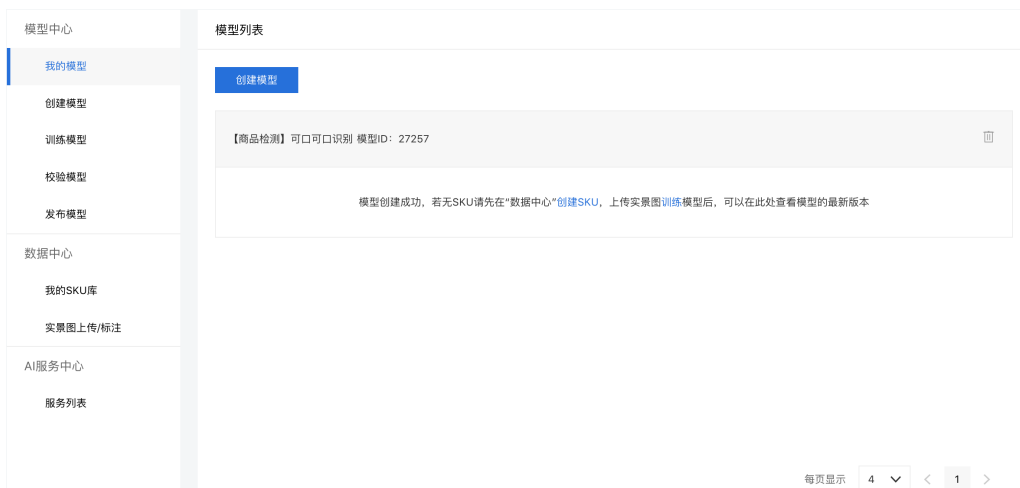
像下图展示的一样完成所有填写项后点击【下一步】按钮完成模型创建，创建完成后会跳转到【我的模型】页面。



## 步骤2.创建SKU

这个步骤将会介绍如何创建SKU，SKU是客户需要检测的商品，在训练品台上有两个作用，其一是“SKU名称\_品牌名称\_规格参数”用于标注训练数据的标签，二是SKU的单品图片用于商品增强合成技术，提高模型效果。

完成上一个步骤后，会跳转到【我的模型】页面，这时您会看到如下图展示的内容，由于模型还未训练，所以模型列表中没有显示模型的效果，在训练模型前，需要先完成SKU的创建。



点击左侧列表中的【我的SKU】进入SKU管理页面，点击【创建SKU】按钮进入创建SKU页面，您会看到如下图展示的内容



提示：在调用API接口识别SKU时，识别结果中SKU的名字是以“SKU名称\_品牌名称\_规格参数”的形式返回的，所以在填写SKU名称、品牌名称和规格参数时避免这三项内容重复。

需要填写的项目如下：

- SKU名称

SKU的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等

- 品牌名称

SKU的品牌名称，如可口可乐，乐事，伊利等

- 规格参数

SKU的规格，如330ml，500g，20片等

- 商品品类

可选择的有饮品、药品、保健品、零食、香烟、调味品、日用品和其他

- 包装类型

可选择的有瓶装、罐装、袋装、盒装和其他

- 商品编号

如果您自身的业务系统中有现成SKU对应的商品编码，比如商品条形码，可以填在该填写框中，之后模型接口将支持返回该内容，用于您快速匹配SKU

- SKU单品图

SKU的单品图将用于商品增强合成，拍摄角度和上传张数基本原则是覆盖实际检测场景可能出现的角度，具体请参考[SKU单品图数据](#)文档中进行单品图采集。如果不上传，将会降低模型的识别效果，可以点击页面上的【示例图片】查看SKU单品图样张。

完成填写和上传SKU单品图上传后，页面内容显示如下图所示

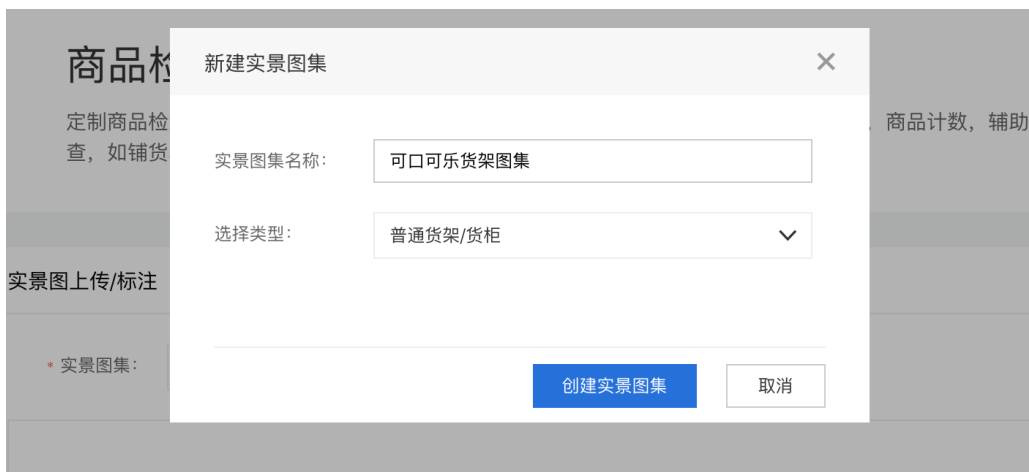
点击【创建SKU】按钮完成创建，点击后回到【我的SKU库】，SKU列表中的SKU图数需要大约5秒的时间进行计算，刷新页面即可显示SKU单品图片数。

### 🔗 步骤3.上传和标注训练数据

这个步骤将会介绍如何上传和标注训练数据，训练数据是SKU在货架上的实景图，需要客户从真实的业务场景中采集，这些图片在被正确标注中，可以用于训练成模型。

完成上一个步骤后，在左侧列表中点击【实景图上传/标注】进入上传和标注页面，在上传前请在实景图集选择栏内创建实景图集，如下图所示

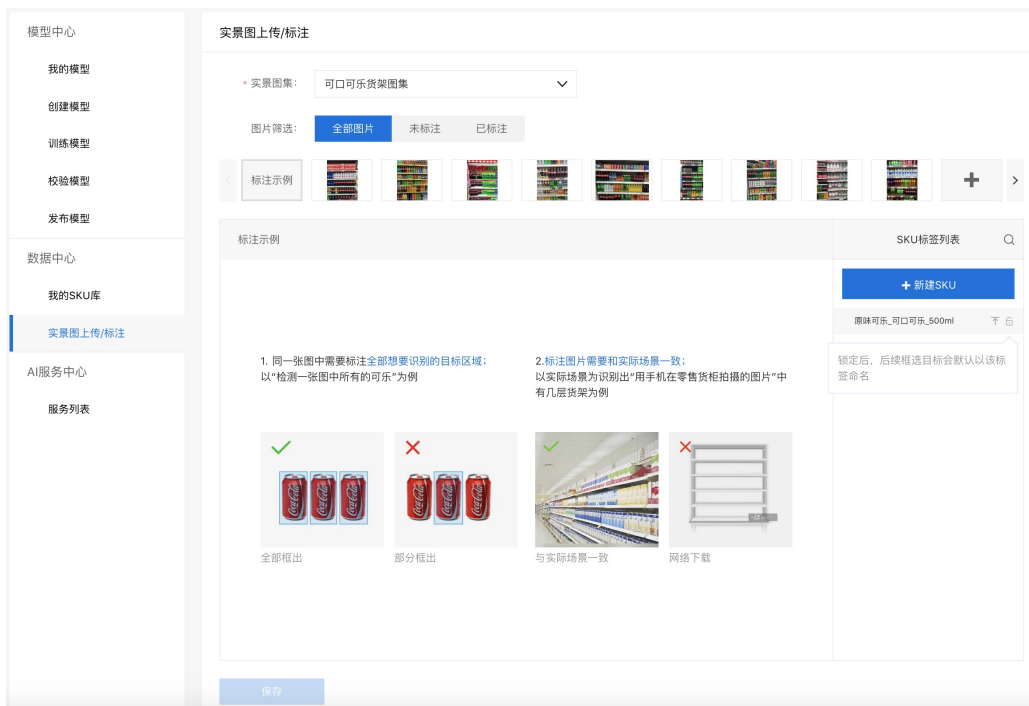




需要填写的项目如下：

- **实景图集名称**  
实景图集的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等
- **选择类型**  
实景图集的类型，请与创建模型时选择的应用场景保持一致，上传时只上传跟选择类型相同的实景图。可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

完成创建实景图集后，页面显示为如下图所示的内容



点击页面上【标注】为该实景图集上传作为训练数据的实景图，点击【标注示例】右侧的加号上传实景图。

实景图基本要求如下：

实景图的具体采集要求，请参考[实景图数据要求文档](#)

- 实景图片需要是从真实业务场景中采集来的数据
- 支持上传的图片格式为jpg, png, jpeg, bmp，大小限制为4M
- 建议图片尺寸：最长不超过4096px，最小不低于30px，长宽比3：1以内

标注基本要求如下：

实景图的具体标注要求，请参考[实景图标注规范文档](#)

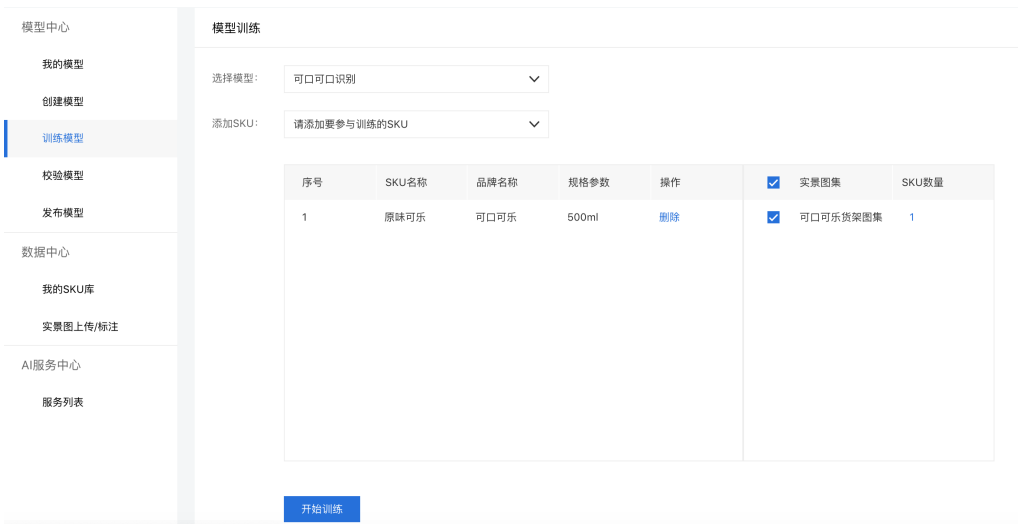
- 完整并仅仅框选要识别的SKU
- 标注框不要框选到其它SKU或是价目标签等非要识别的SKU的干扰信息
- 在实景图中出现的所有要识别的SKU必须全部标注，不能遗漏

完成所有实景图的标注后，返回到【我的SKU库】可以查看到SKU列表中【实景图数】列显示标注了该SKU的实景图片的数量，如下图所示



#### 步骤4.训练模型

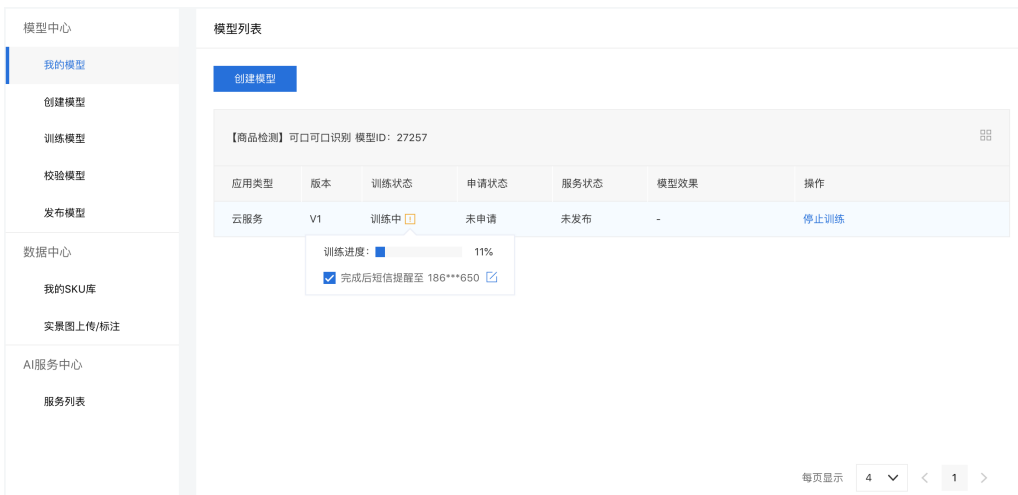
这个步骤将会介绍如何训练模型



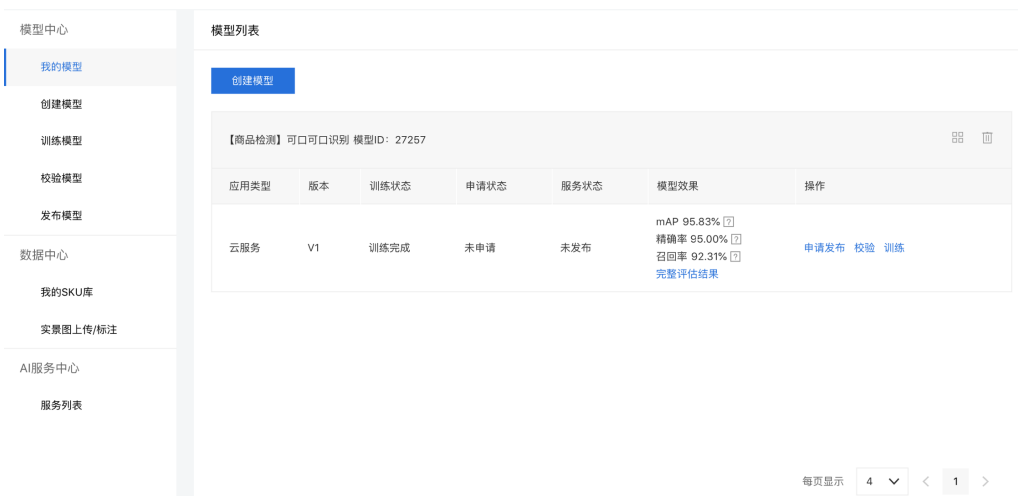
如上面图片所示，点击左侧列表中的【训练模型】，需要先后完成下面三项选择：

1. 选择要训练的模型
2. 选择需要模型支持检测的SKU，选择完成后，下方左侧会显示已添加的SKU，右侧会显示包含已添加SKU的实景图集
3. 选择要参与训练的实景图集

完成选择后，点击【开始训练】按钮页面跳转至【我的模型】页面，如下图所示，可以看到模型已进入训练状态，将鼠标移至状态"训练中"右边的小问号上，可以查看训练进度，训练进度数值只是作为参考，所以推荐打开短信通知功能，这样就第一时间知晓模型训练完成了。

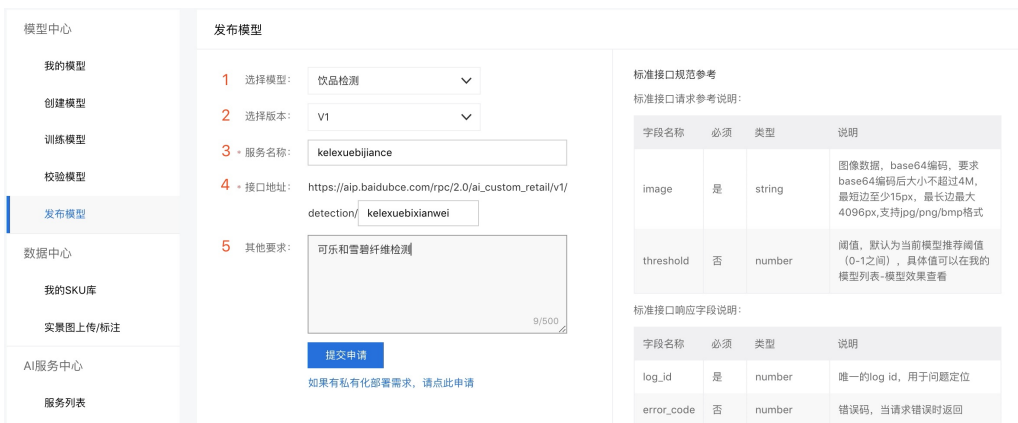


训练完成后，可以点击校验和申请发布。



## 步骤5.发布模型

这个步骤将会介绍如何将训练好的模型发布为服务API



在模型训练好后，点击模型列表内对应模型「操作」列中的「申请发布」，或是在左侧导航栏点击「发布模型」可以进入发布模型页面，如上图所示。在对应选项中选择和输入相应内容发起模型发布的申请：

### 1. 选择模型（必选）

选择需要发布的模型，只能选择已经完成训练的模型

### 2. 选择版本（必选）

选择需要发布的模型版本，只能选择完成训练且没有发布过的版本

### 3. 服务名称（必填）

为发布的服务命名，服务名称不得多于20个字符

### 4. 接口地址（必填）

自定义服务的API URL，接口地址需要多于5个字符但不能超过20个字符，仅限英文

## 5. 其他要求

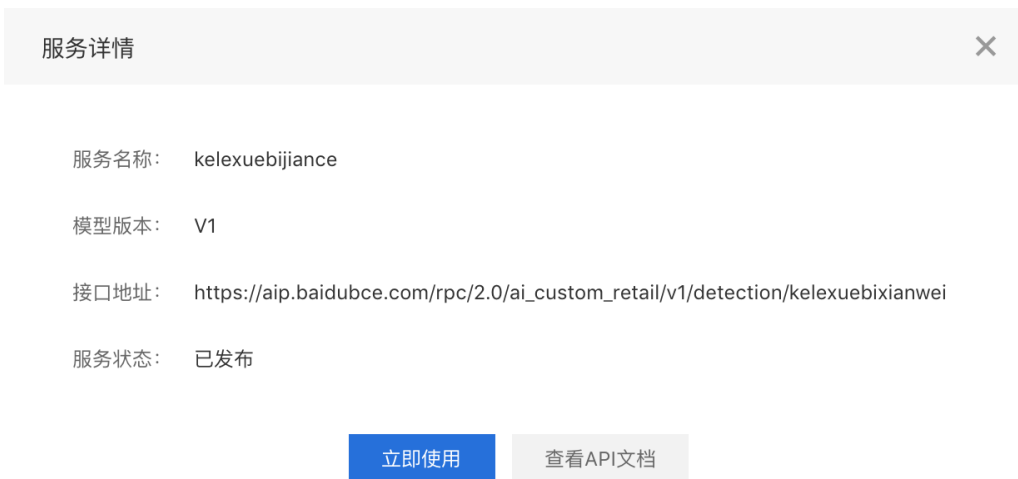
如果有其他要求可以输入要求描述

填写完上述信息后，点击「提交申请」完成发布模型申请。提交申请后，模型列表内该模型的申请状态和服务状态为有以下几种情况：

申请状态	服务状态	状态描述
审核中	未发布	服务刚申请发布，模型在审核中
审核成功	发布中	服务通过审核，进入系统自动发布阶段
审核成功	已发布	服务发布成功
审核失败	未发布	服务未通过审核，通常为模型训练结果mAP < 0.6，如需申诉，可以加入官方QQ群（群号:1009661589）咨询群管

提示：第一次申请发布的模型需要人工审核，通常4小时内完成，如果希望加急上线，请加入官方QQ群（群号:1009661589）咨询群管高优审核。非第一次申请发布的模型，如果模型训练结果mAP>0.6，则会自动通过审批。审批完成后，大约需要5分钟左右自动完成发布。

发布成功后，可以点击模型列表内「操作」列中的「服务详情」获取服务API URL，点击后弹出下图所示窗口：



点击「查看API文档」可以快速跳转至API文档，参考文档调用API获取商品检测AI能力。

## EasyDL图像SDK集成快速开始

### 通用设备端Android ARM

#### 简介

本文档包括两个部分，分别适用于SDK测试、SDK集成。开发者可根据实际需求选择参考：

- SDK测试文档：供测试Demo时参考
- SDK集成文档：供将SDK集成进自己的代码时参考

#### 测试前的准备

- Android系统的硬件及开发环境
  - 详情参考下方文档
- EasyDL平台的Android SDK
  - 以图像分类为例，前往[操作台](#)训练模型后，选择发布为Android系统的通用设备端SDK，发布成功后即可从平台下载
- 用于激活设备端SDK的序列号
  - 前往[控制台](#)申请用于激活通用设备端SDK的序列号

#### Android SDK 测试文档

这个部分将为新手提供一个快速测试Easydl & EasyEdge的Android Demo的图文教程。

效果展示



## 识别结果

置信度



0.30

序号

名称

置信度

1

mao

0.98

## 测试前的准备

- 硬件：

1. 准备一台PC机
2. 准备一台较新款的Android 手机 不支持模拟器

- SDK

1. 您已经生成Android SDK 并且已经下载成功
2. 您已经通过扫描二维码的形式，在这台Android 手机上测试成功想要的功能。

- 生成序列号

1. 如果是开源模型版本，不需要序列号，
2. 如果测试的是“按单台设备激活”，需要额外获取一个序列号
3. 如果需要额外测试“产品线激活”的序列号，需要再准备一个包名。demo的包名com.baidu.ai.easyaimobile.demo。

- 开发环境

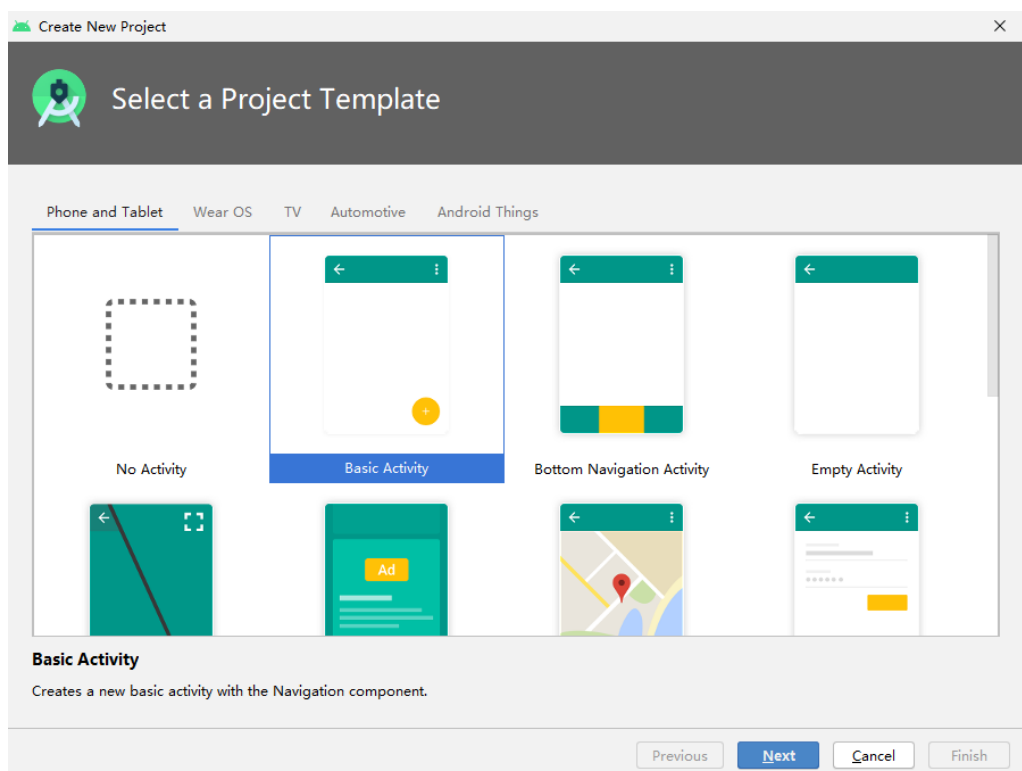
1. 因为Android是google公司的项目，测试过程中可能需要访问国外网站下载资源。
2. PC机上安装较新版本的Android Studio， 本文使用的是4.0.1版本。[下载地址](#)
3. 在这台Android手机上测试通过一个Android HelloWorld项目

## Android Studio的安装及手机测试

本段简单地描述如何通过Android Studio在手机上运行一个自带的Demo，有android基础的可以跳过本段。

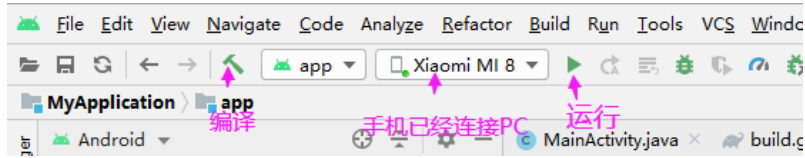
手机上需要打开开发者模式，您也可以下载[91助手](#)，按照软件提示连接手机。

更多Android Studio的安装测试也可以百度下【[Android Studio自动生成Demo](#)】。新建Android Studio自带的测试项目，菜单File->New Project..，弹框中选择“Basic Activity”，点“Next”，之后用默认配置，点“Finish”后项目就生成了。



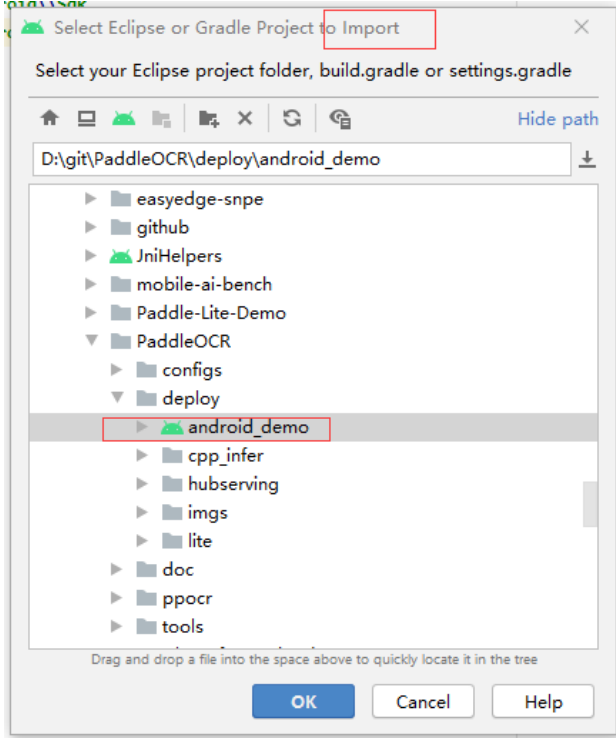
小技巧：Android项目需要同步后才能出现编译和运行的选项。强制同步的方法为：菜单File->Sync Project with Gradle Files

导入成功后，有以下的图标：

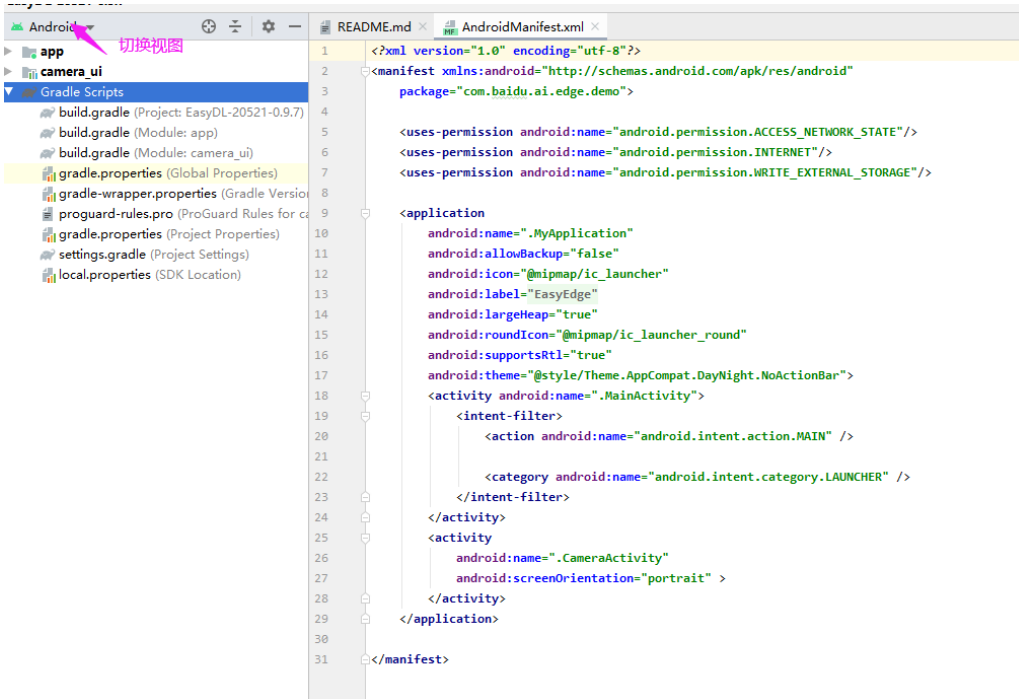


### 导入官方Demo

导入项目： 菜单File->New-> Import Project .., 选择PaddleOCR\deploy\android\_demo目录。 注意千万不要使用菜单File->New-> New Project..



导入项目后，会触发gradle的自动同步，最终效果如下：



此时项目可以正常编译，

- 如果是开源模型版本，此时可以正常运行。
- 如果需要序列号的情况，此时会界面会报错“序列号错误”

填入序列号 如果是开源模型版本，不需要序列号，序列号保持为null即可 在MainActivity开头部分填入您的序列号



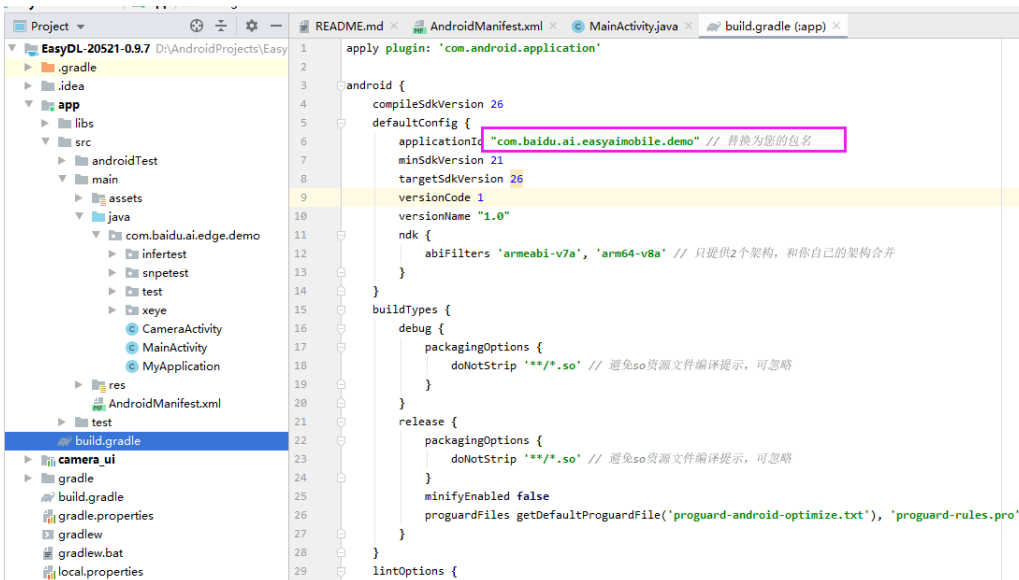


此时项目可以正常编译及运行。

-如果“产品线激活”的序列号，还需要额外修改包名

### 修改包名（仅“产品线激活”需要）

如果您填入的包名是"com.baidu.ai.easyaimobile.demo" 如图修改：



此时项目可以正常编译及运行。

### 精简版测试

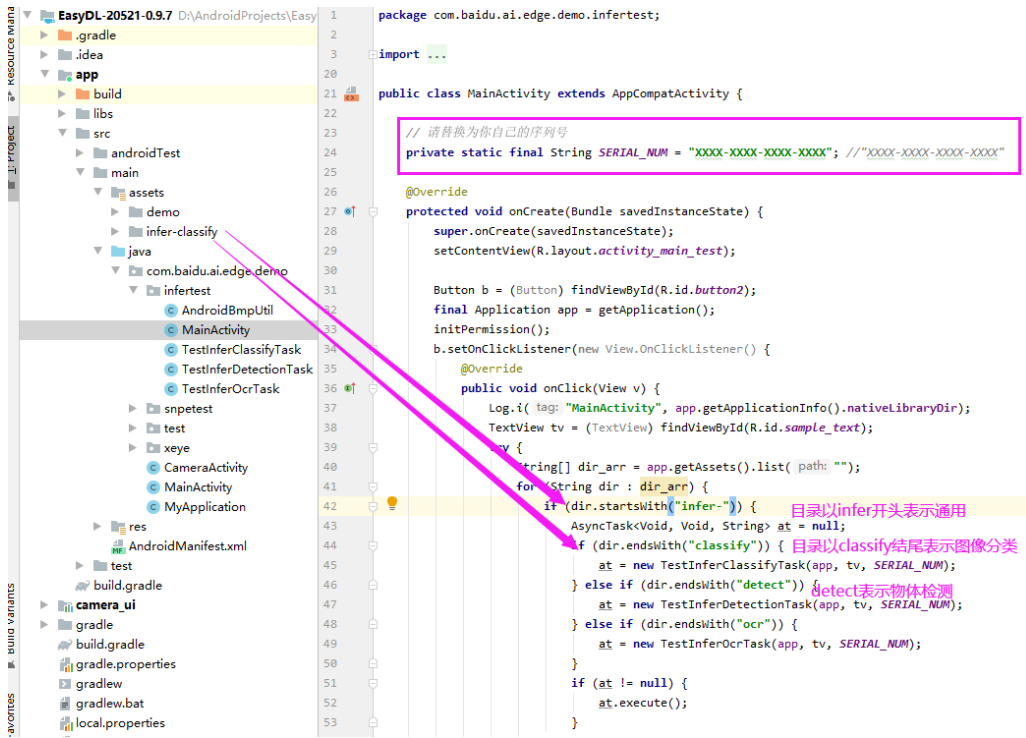
仅限通用arm的 图像分类，物体检测，文字识别。其它引擎可以参考自行写。如果是开源模型版本，不需要序列号，序列号保持为null即可。

使用MiniActivity可以在如下情况下测试：

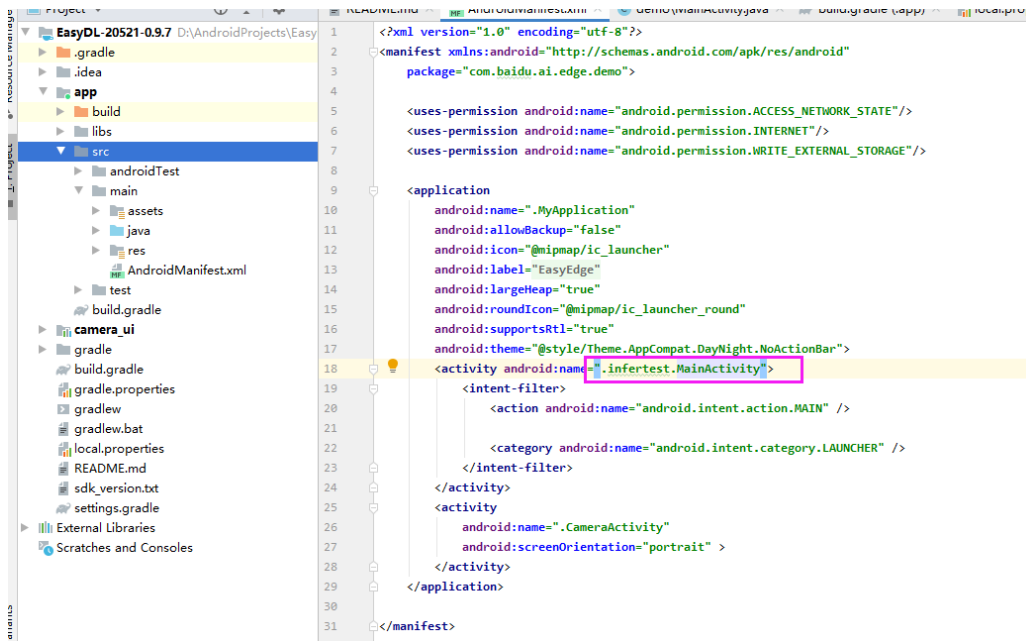
- 不带摄像头或官方demo运行摄像头报错的开发板
- 避免摄像头预览占用CPU导致耗时测试不准确

具体步骤如下：

A. 在infertest.MainActivity中，修改文件开始位置的序列号

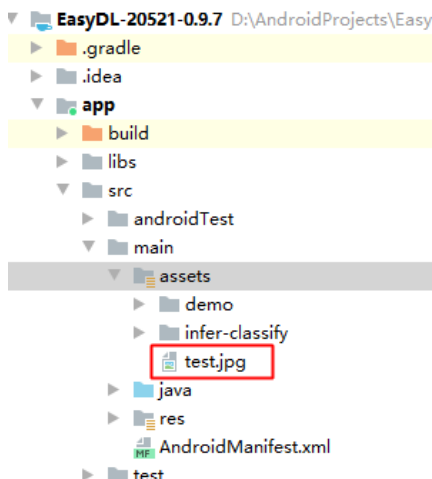


B. 修改启动Activity为infer.test.MainActivity，修改AndroidManifest.xml文件。主要不要漏掉开头的“.”

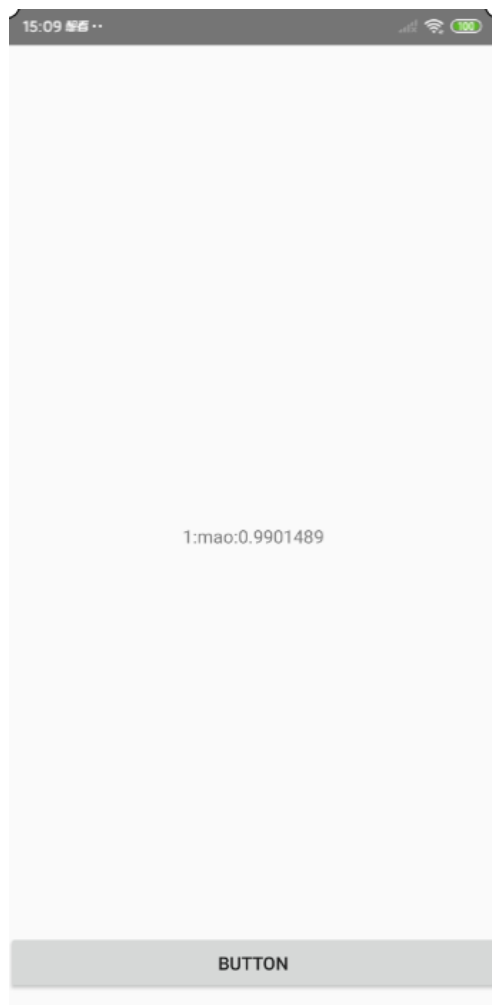


此时启动在logcat中会发现缺少test.jpg

C. 将你的测试图片test.jpg 放入assets目录。



此时再次运行，点击界面上的按钮，有如下测试成功的界面：



#### Android SDK集成文档

这个部分以Android Studio 自带的Empty Activity 模板项目为例，展示如何集成OCR Android的代码到您自己的项目中

#### 集成前的准备

1. 需要一个较新款的Android手机
2. 请先根据上方的测试文档配置环境及测试官方Demo
3. 请先根据上方的测试文档测试MiniActivity，本文以MiniActivity为模板集成

#### 集成后的代码下载

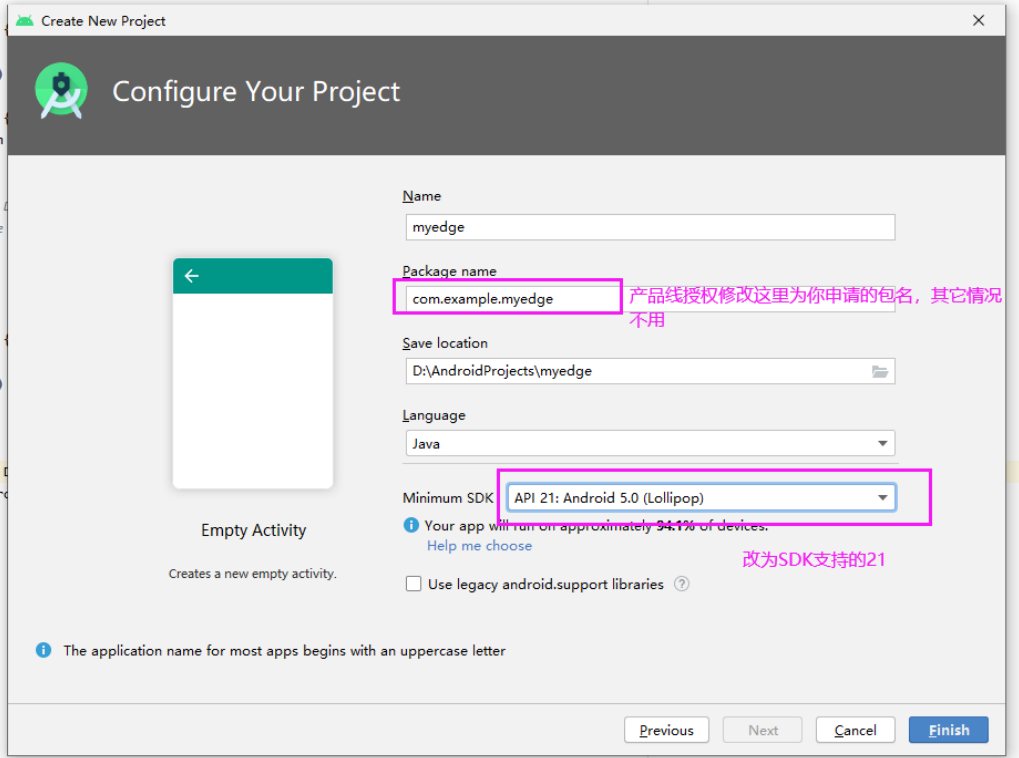
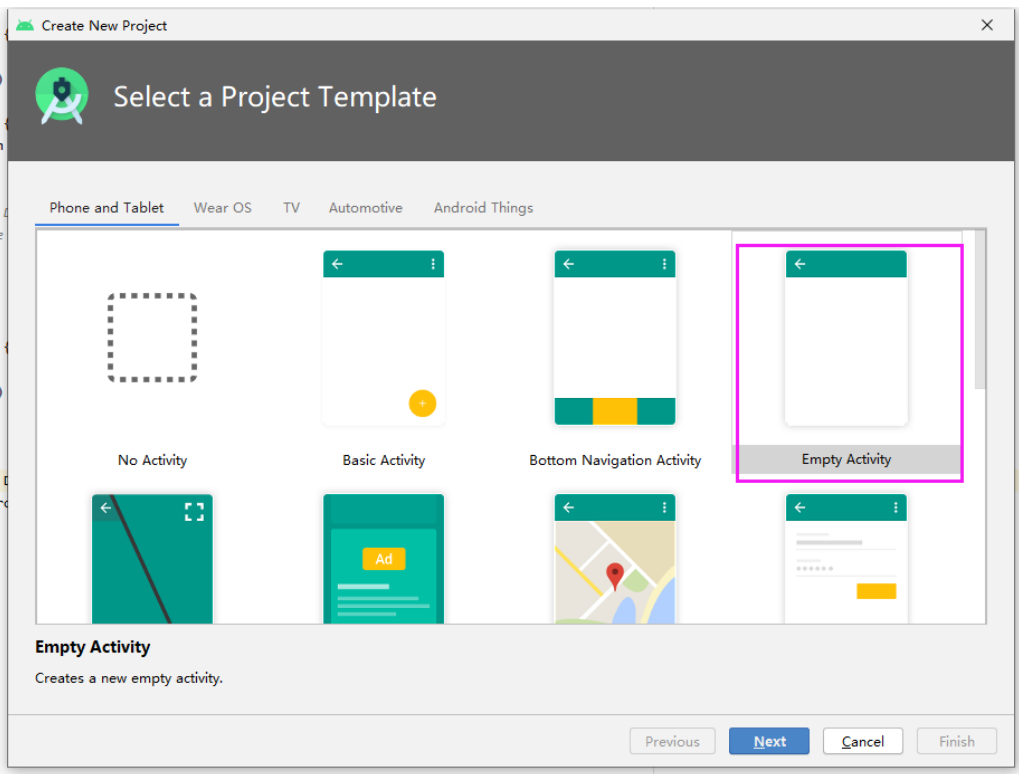
如果觉得下面步骤有模糊的地方，可以参照修改好的代码进行下载。

链接：<https://pan.baidu.com/s/1cTFxYrzb1jp8bWBs6eoF8A> 提取码：u7xv

zip 文件名	说明
myedge-init.zip	初始化的“Empty Activity”模板项目
myedge-finished.zip	做完本文所有步骤后的项目

#### 新建一个项目

新建Android Studio自带的测试项目，菜单File->New Project..，弹框中最后一个项目模板“Empty Activity”，点“Next”，之后用默认配置，点“Finish”后项目就生成了。这里比如给这个项目起名为myedge



查看Logcat

有Android开发经验的用户可以跳过本段。

修改MainActivity文件

```

@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    // 加上下面这行
    Log.i("MainActivity", "SHOW in Logcat"); // 表示记录info级别的日志
    // Example of a call to a native method
    TextView tv = findViewById(R.id.sample_text);
    tv.setText(stringFromJNI());
}

```

代码会标红，此时鼠标在红色的“Log”上点以下，会提示Alt+Enter，按下Alt+Enter，文件的第6行左右会自动添加

```

import android.util.Log;

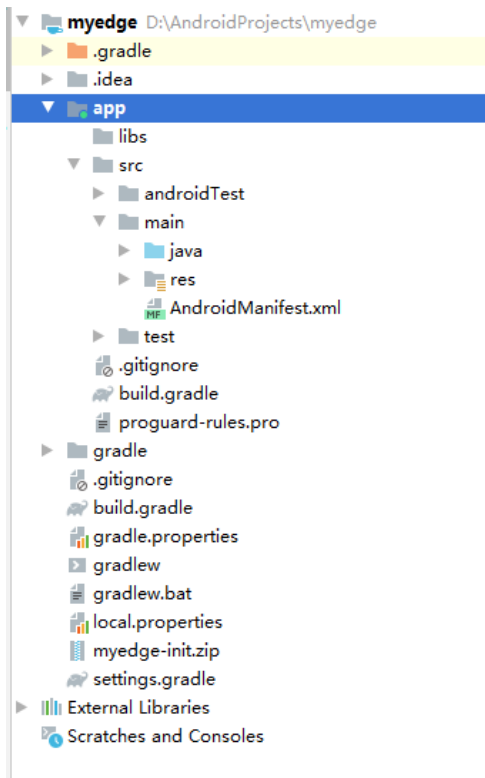
@Override
protected void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);
    Log.i("MainActivity", "SHOW in Logcat");
    // Example of a call to a native method
    TextView tv = findViewById(R.id.sample_text);
    tv.setText(stringFromJNI());
}

```

再次运行项目，可以在界面的“Run”和Logcat里看见我们之前打印的日志 “SHOW in Logcat”



### “Empty Activity”模板项目目录介绍



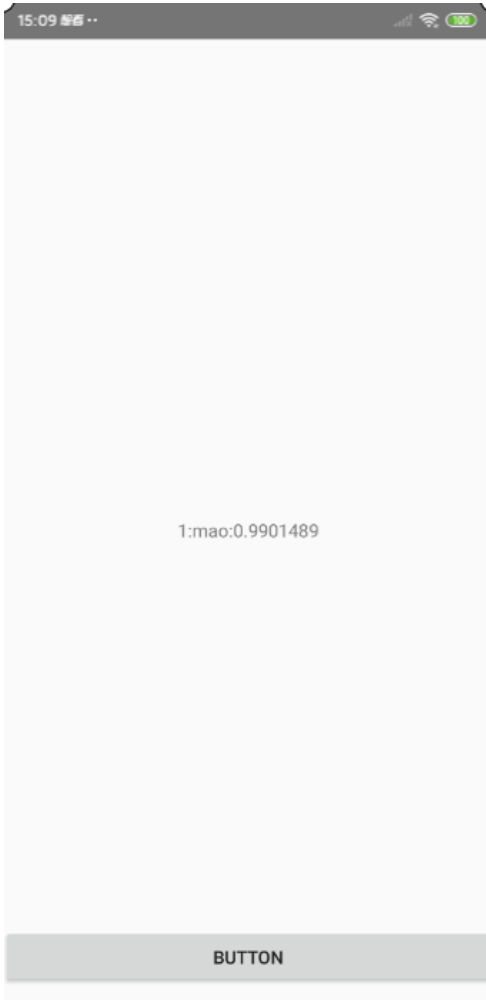
本文会操作上图中的目录及文件:

- app/libs 目录下放入jar库文件，也可以放so文件

- app/src/main/java java代码目录
- app/src/main/assets 目前无此目录，之后放入模型文件
- app/src/main/AndroidManifest.xml AndroidManifest.xml文件，设置启动Activity和权限
- app/src/main/res/layout UI布局目录
- app/build.gradle 编译配置，比如修改包名

### 集成代码

集成之前，请确认已经跑通官方demo的精简版，官方demo有如下界面及类似结果



### 集成步骤：

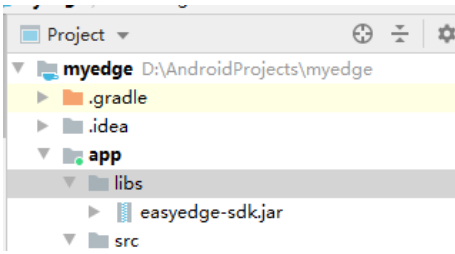
1. 集成库
2. 集成java代码
3. 设置权限及配置项
4. 复制模型文件

#### 1. 集成库

复制app/libs 目录下库文件到自己的项目中

##### A. 复制easyedge-sdk.jar库文件

- 如果项目中已经有其它的jar文件，那么和这些jar文件放一起
- 如果项目中没有其它的jar文件，参照官方demo方式，复制到app/libs目录下（本文的情况），与官方demo放在相同的位置



B. 复制so目录

需要复制官方demo的libs/arm64-v8a 及 armeabi-v7a

- 如果项目中已有so库目录，arm64-v8a 及 armeabi-v7a下的so与已有目录合并。如果比如自己项目只存在arm64-v8a目录，那么官方demo的armeabi-v7a就不需要复制了。
- 如果项目中没有so库目录（本文情况），以下二个方式二选一 复制arm64-v8a 及 armeabi-v7a目录 到 app/src/main/jniLibs目录下参照官方demo方式，复制arm64-v8a 及 armeabi-v7a目录到 app/libs目录下，与官方demo放在相同的位置。并修改app/build.gradle，设



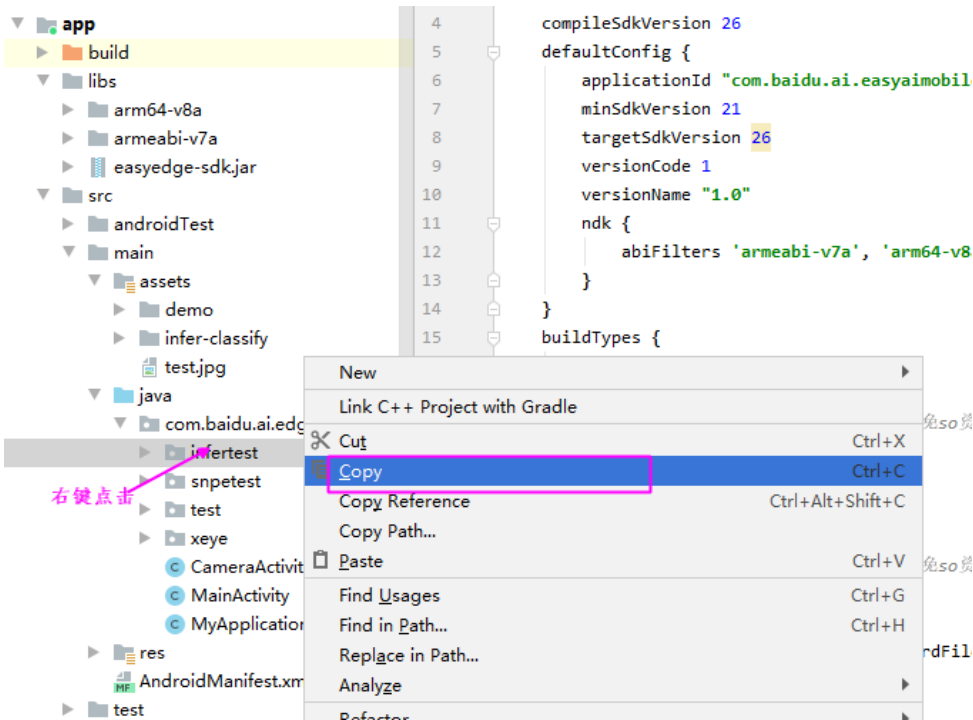
2. 集成java代码

复制官方demo的infertest目录到自己项目中的infertest目录下，不必修改自己项目的包名。复制layout下的activity\_main\_test.xml到自己的项目中。之后修改android.appcompat类为androidx.appcompat下的。

具体步骤如下：

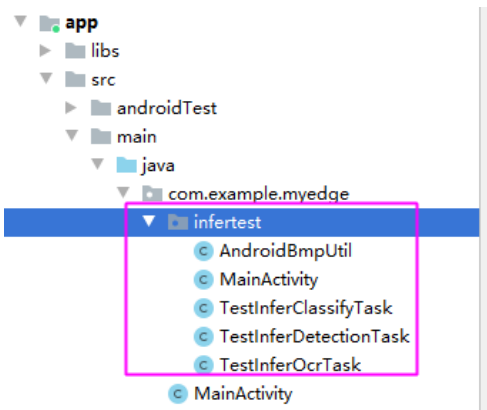
A. 复制官方demo的infertest目录

打开官方demo，右键点击java目录下的infertest目录，点“copy”

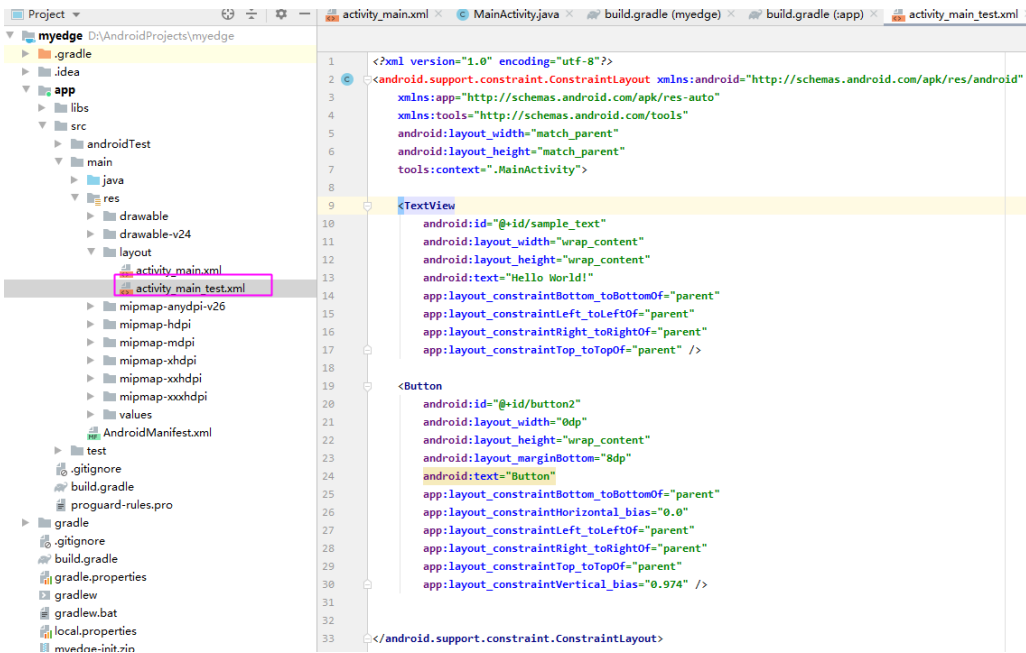


复制到自己项目中类似位置：





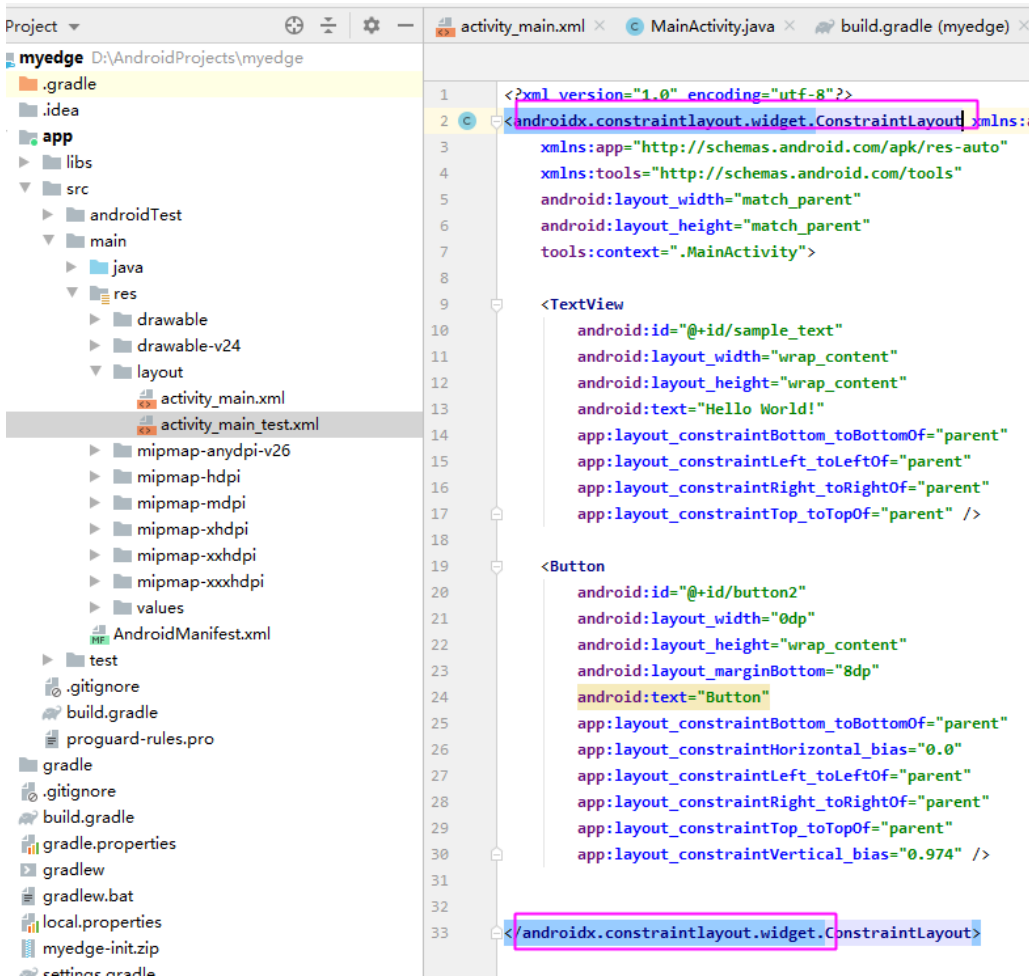
B. 复制官方demo的layout下的app/src/main/res/layout/activity\_main\_test.xml到自己项目的同名文件



此时会发现xml文件里android.support.constraint.ConstraintLayout不存在，原因是自己的项目新建时用的是androidx.appcompat

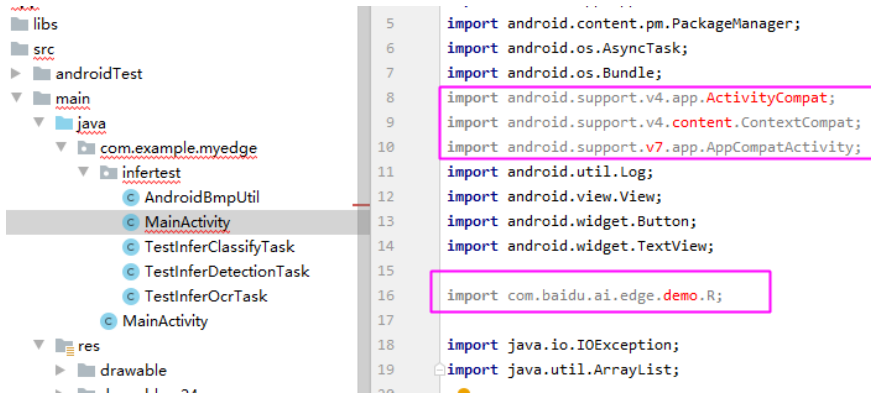
C. 修改activity\_main\_test.xml文件

将 android.support.constraint.ConstraintLayout 修改为androidx.constraintlayout.widget.ConstraintLayout



D. 修改infertest/MainActivity.java文件

删除飘红的类导入：



选下面的飘红的类，使用Alt+Enter自动导入缺失的类。

```

1 package com.example.myedge.infertest;
2
3 import android.Manifest;
4 import android.app.Application;
5 import android.content.pm.PackageManager;
6 import android.os.AsyncTask;
7 import android.os.Bundle;
8 import android.util.Log;
9 import android.view.View;
10 import android.widget.Button;
11 import android.widget.TextView;
12
13
14 import androidx.appcompat.app.AppCompatActivity;
15 import androidx.core.app.ActivityCompat;
16 import androidx.core.content.ContextCompat;
17
18 import com.example.myedge.R;
19
20 import java.io.IOException;
21 import java.util.ArrayList;
22
23 public class MainActivity extends AppCompatActivity {
24
25     // 请替换为你自己的序列号
26     private static final String SERIAL_NUM = "8B83-E684-258D-8D4B"; // "XXXX-XXXX-XXXX-XXXX"
27

```

此时项目可以编译成功，但是不能运行

### 3. 设置权限及配置项

将启动的Activity改为infertest.MainActivity, 并根据官方demo添加网络和外部储存权限

具体步骤如下：

#### A. 将启动的Activity改为infertest.MainActivity

修改app/AndroidManifest.xml文件，将启动的Activity改为从.MainActivity改为infertest.MainActivity,

```

1 <?xml version="1.0" encoding="utf-8"?>
2 <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3     package="com.example.myedge">
4
5     <application
6         android:allowBackup="true"
7         android:icon="@mipmap/ic_launcher"
8         android:label="myedge"
9         android:roundIcon="@mipmap/ic_launcher_round"
10        android:supportRtl="true"
11        android:theme="@style/AppTheme">
12        <activity android:name=".infertest.MainActivity">
13            <intent-filter>
14                <action android:name="android.intent.action.MAIN" />
15
16                <category android:name="android.intent.category.LAUNCHER" />
17            </intent-filter>
18        </activity>
19    </application>
20 </manifest>

```

#### B. 根据官方demo添加网络和外部储存权限

添加如下权限

```

<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
<uses-permission android:name="android.permission.INTERNET"/>
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>

```

```

1  <?xml version="1.0" encoding="utf-8"?>
2  <manifest xmlns:android="http://schemas.android.com/apk/res/android"
3      package="com.example.myedge">
4      <uses-permission android:name="android.permission.ACCESS_NETWORK_STATE"/>
5      <uses-permission android:name="android.permission.INTERNET"/>
6      <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
7
8      <application
9          android:allowBackup="true"
10         android:icon="@mipmap/ic_launcher"
11         android:label="myedge"
12         android:roundIcon="@mipmap/ic_launcher_round"
13         android:supportRtl="true"
14         android:theme="@style/AppTheme">
15         <activity android:name=".infertest.MainActivity">
16             <intent-filter>
17                 <action android:name="android.intent.action.MAIN" />
18
19                 <category android:name="android.intent.category.LAUNCHER" />
20             </intent-filter>
21         </activity>
22     </application>
23 </manifest>

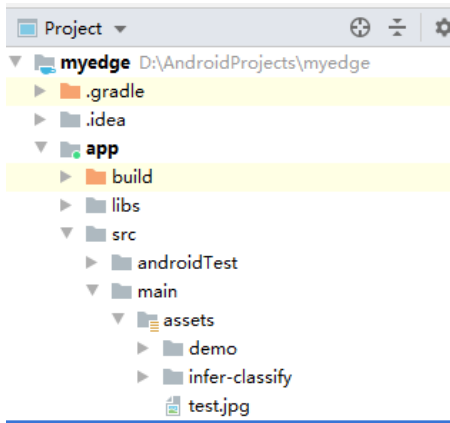
```

注意，android 6.0以上，需要额外代码ActivityCompat.requestPermissions申请权限，具体代码见infertest.MainActivity中initPermission方法

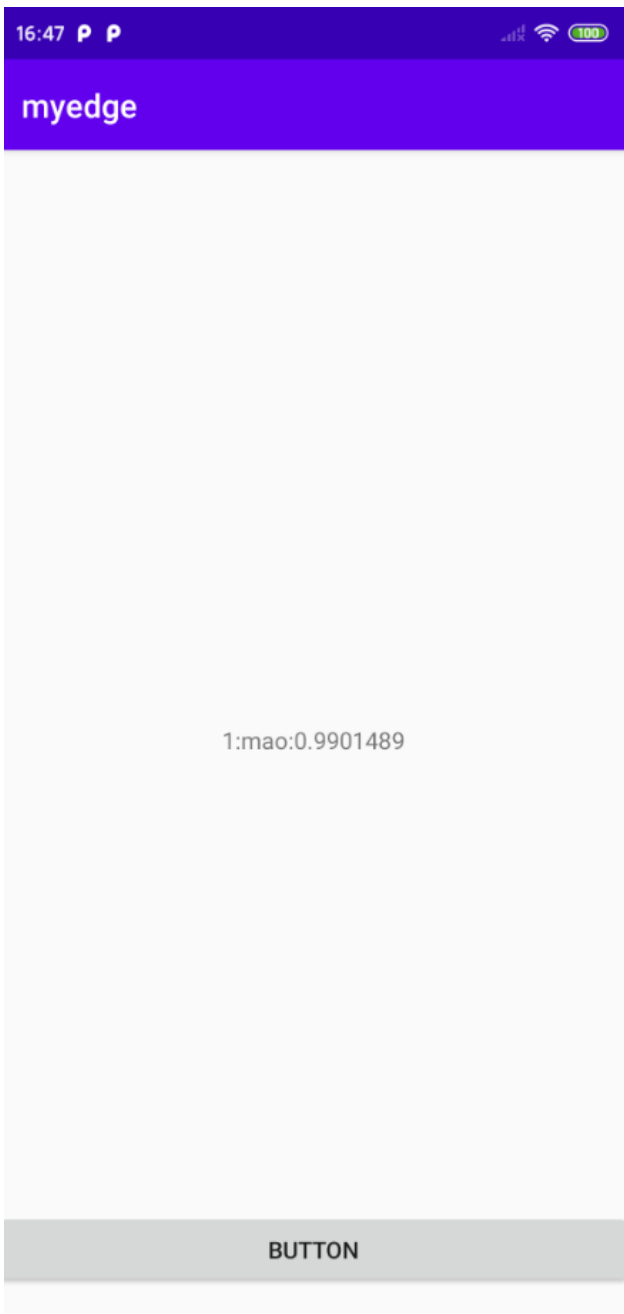
#### 4. 复制模型文件

复制官方demo的app/src/main/assets目录到自己项目的同名目录。

如果已经存在，合并即可。

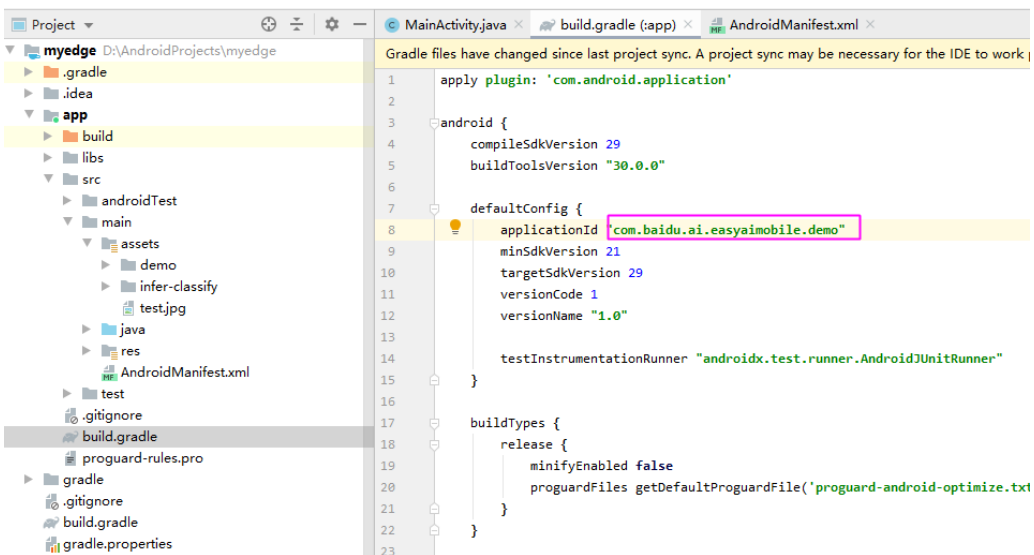


此时运行，可以获得和官方demo精简版一样的效果。



修改包名 (仅“产品线激活”需要)

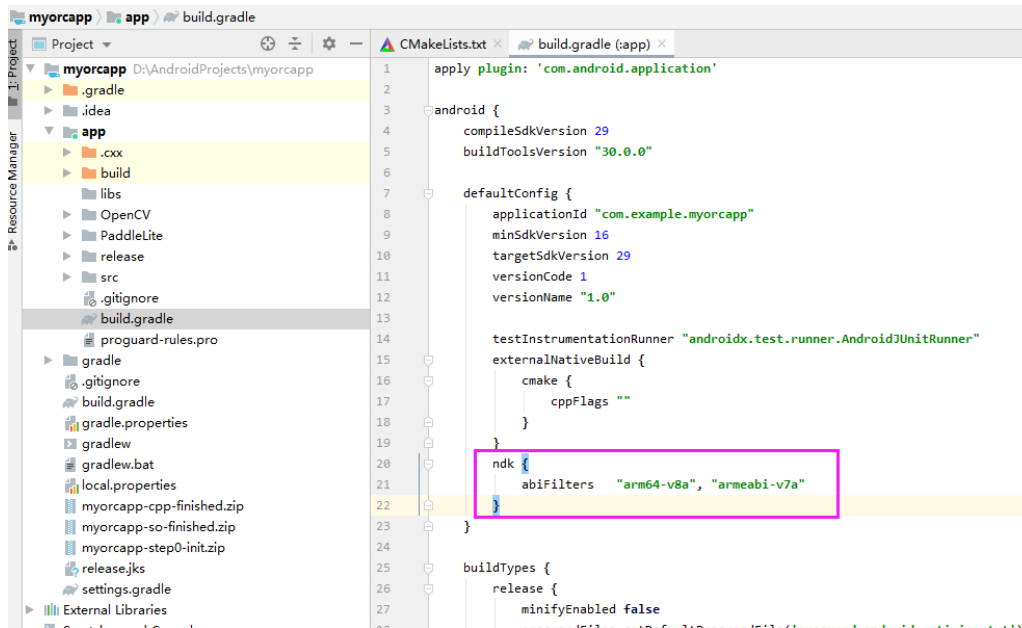
如果您填入的包名是"com.baidu.ai.easyaimobile.demo" 如图修改app/build.gradle :



一点小优化

添加指定架构 (C++ 方式可能需要) app/build.gradle中添加指定

```
ndk {
    abiFilters "arm64-v8a", "armeabi-v7a"
}
```



## 通用设备端Linux ARM

### 简介

本文将为新手提供一个快速测试和集成EasyDL & EasyEdge的 Linux Arm SDK的图文教程。

### 测试前的准备

- Linux ARM的硬件及开发环境
  - 详情参考下方文档
- EasyDL平台的Linux ARM SDK
  - 以图像分类为例，前往[操作台](#)训练模型后，选择发布为Linux ARM的通用设备端SDK，发布成功后即可从平台下载
- 用于激活通用设备端SDK的序列号
  - 前往[控制台](#)申请用于激活通用设备端SDK的序列号
  - 首次使用SDK或者更换序列号、更换设备时，需要联网激活。激活成功之后，有效期内可离线使用

### 效果展示

```
0 build > ./easyedge_demo ~/lvxiangxiang/models/SqueezeNetV1.1-tf/ ~/lvxiangxiang/images/orange.jpg
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 The recommended threshold is 0.3
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Rescale mode is 0, target_size: 0, max_size: 0
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Config file read done
2020-07-16 16:52:33,636 DEBUG [EasyEdge] 548062597120 Local license is ok.
2020-07-16 16:52:33,855 DEBUG [EasyEdge] 548062597120 Image will be resized to 227,227
2020-07-16 16:52:33,986 DEBUG [EasyEdge] 548062597120 Inference costs 131.011(127.854) ms
950, orange, p:0.972394
Done
```

【图像分类】0 AlexNet-fluid



label	置信度
n07747607 orange	0.951
n07749582 lemon	0.030
n07716906 spaghetti squash	0.008
n03942813 ping-pong ball	0.004
n07717556 butternut squash	0.003
n03134739 croquet ball	0.001
n03929690 puck, plectrum, plectron	0.001
n04332243 strainer	0.001
n03930642 honeycomb	0.000
n12144590 corn	0.000
n04409916 tennis ball	0.000
n04019541 puck, hockey puck	0.000

环境准备 硬件环境 本SDK

适用于Linux Arm操作系统，如

- ubuntu、centos等
- 树莓派Raspbian
- ...

且适用于aarch64和armv7hf的CPU架构。

用户使用以上系统和架构的硬件（如RK3399开发板、树莓派4B等）即可。**网络环境** 用户安装软件和测试SDK都需要联网，用户使用的硬件需确保有效的网络连接。

**软件环境** 在选定硬件之后，需要在硬件上安装以下软件和第三方库以保证SDK正常编译和运行：

- cmake 3 +
- gcc 5.4 +
- opencv3.4 (可选)

可以使用以下方法确认cmake是否满足要求：

```
$ cmake --version
cmake version 3.13.3
```

若系统提示找不到cmake命令或者cmake version 低于3.x.x，则需要安装/升级cmake。

```
// apt安装cmake
sudo apt update
sudo apt install cmake
```

ubuntu官方的apt源update较慢，且可能访问不了，可以替换为国内的源：<https://www.cnblogs.com/yongy1030/p/10315569.html>

也可以使用源码编译的方式安装，参考[安装方法](#)。

```
// 升级cmake
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:george-edison55/cmake-3.x
sudo apt-get update
sudo apt-get upgrade
```

安装/升级cmake后可再次执行cmake --version确认版本。

可以使用以下方法确认gcc是否满足要求：



```
$ gcc --version
gcc (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609
```

若系统提示找不到gcc命令或者gcc version 低于5.4.0，则需要安装/升级gcc。

```
// 安装gcc
sudo apt update
sudo apt install build-essential
// 升级gcc
sudo add-apt-repository ppa:ubuntu-toolchain-r/test # 如果找不到add-apt-repository命令，执行：apt-get install software-properties-common
sudo apt-get update
sudo apt-get install -y gcc-5 g++-5

cd /usr/bin # 升级gcc 5之后，还需要替换原来的软链接
sudo rm -r gcc # 移除之前的软链接
sudo ln -sf gcc-5 gcc # 建立gcc5的软链接
sudo rm -r g++ # 同gcc
sudo ln -sf g++-5 g++
```

安装/升级gcc后可再次执行gcc --version确认版本。

目前没有合适的方法确认系统中是否有SDK需要的OpenCV，若用户不确定是否安装OpenCV 3.4 +，并且可以被cmake find\_package到，可以手动编译安装OpenCV 3.4，也可以在之后编译SDK时自动编译OpenCV。

若选择在下一步编译EasyEdge SDK时自动编译OpenCV，则以下编译安装OpenCV的步骤可跳过。

下载OpenCV 3.4源代码包并解压：[下载地址](#)，然后编译安装：

```
// 编译安装OpenCV
cd opencv-3.4.6
mkdir build
cd build

cmake .. -DBUILD_DOCS=OFF -DBUILD_EXAMPLES=OFF -DBUILD_opencv_python2=OFF -DBUILD_opencv_python3=OFF -
DBUILD_WITH_DEBUG_INFO=OFF -DBUILD_PACKAGE=OFF -DBUILD_opencv_core=ON -DBUILD_opencv_imgproc=ON -
DBUILD_opencv_imgcodecs=ON -DBUILD_opencv_highgui=ON -DBUILD_opencv_video=OFF -DBUILD_opencv_videoio=OFF -
DBUILD_opencv_dnn=OFF -DBUILD_opencv_apps=OFF -DBUILD_opencv_flann=OFF -DBUILD_opencv_gpu=OFF -DBUILD_opencv_ml=OFF -
DBUILD_opencv_legacy=OFF -DBUILD_opencv_calib3d=OFF -DBUILD_opencv_features2d=OFF -DBUILD_opencv_java=OFF -
DBUILD_opencv_objdetect=OFF -DBUILD_opencv_photo=OFF -DBUILD_opencv_nonfree=OFF -DBUILD_opencv_ocl=OFF -
DBUILD_opencv_stitching=OFF -DBUILD_opencv_superres=OFF -DBUILD_opencv_ts=OFF -DBUILD_opencv_videostab=OFF -
DBUILD_opencv_contrib=OFF -DBUILD_SHARED_LIBS=ON -DBUILD_TESTS=OFF -DBUILD_PERF_TESTS=OFF -DBUILD_WITH_CAROTENE=OFF -
DCMAKE_BUILD_TYPE:STRING=Release -DWITH_FFMPEG=OFF -DWITH_IPP=OFF -DBUILD_PNG=ON -DBUILD_JPEG=ON -DBUILD_ZLIB=ON -
DBUILD_FAT_JAVA_LIB=OFF -DOPENCV_CXX11=OFF -DCMAKE_INSTALL_PREFIX:PATH=/usr/lib/aarch64-linux-gnu/

make # 如果有多个cpu可以用-j加快编译速度,如4个CPU用 make -j4
make install
```

## 测试demo

**SDK介绍** 用户下载的Linux Arm SDK zip包中包含SDK动态库、模型等资源文件和测试demo.cpp。

需要将SDK zip包完整的放入Arm硬件上再进行解压，否则可能会报错：

```
libeasyedge.so: file format not recognized; treating as linker script
```

Linux下解压命令：tar -xvf xxx.tar

SDK zip包的目录结构如下：

```
EasyEdge-Linux-mxxx-bxxx-arm
├── cpp
│   ├── baidu_easyedge_linux_cpp_aarch64_ARM_gcc5.4_vx.x.x_xxxxxx.tar # aarch64 SDK
│   └── baidu_easyedge_linux_cpp_armv7hf_ARM_gcc5.4_vx.x.x_xxxxxx.tar # armv7hf SDK
└── RES # 模型、标签和配置文件
```

若用户使用的硬件的CPU架构为aarch64，则解压baidu\_easyedge\_linux\_cpp\_aarch64\_ARM\_gcc5.4\_vx.x.x\_xxxxxx.tar。

若CPU架构为armv7hf，则解压baidu\_easyedge\_linux\_cpp\_armv7hf\_ARM\_gcc5.4\_vx.x.x\_xxxxxx.tar。

RK3399等开发板一般是aarch64架构，树莓派一般是armv7hf架构（最新的4B可以刷成aarch64架构）。

可通过下面的命令确认CPU架构（armv7l在树莓派上实际是指armv7hf）：

```
$ uname -m
aarch64 # 或者是armv7l
```

解压完对应的tar包之后的目录结构如下：

```
baidu_easyedge_linux_cpp_aarch64_ARM_gcc5.4_vx.x.x_xxxxxx
├── demo # 测试demo
│   ├── CMakeLists.txt
│   ├── demo.cpp
│   ├── opencv.cmake
│   └── easyedge_serving
├── include # SDK需要的头文件
│   ├── easyedge
│   │   ├── easyedge_config.h
│   │   └── easyedge.h
├── lib # SDK需要的库文件
│   ├── libeasyedge.so -> libeasyedge.so.x.x.x
│   ├── libeasyedge.so.x.x.x
│   ├── libeasyedge_static.a
│   ├── libpaddle_full_api_shared.so
│   └── libverify.so
└── ReadMe.txt # 文档等其他说明
```

编译demo 前面安装了cmake、gcc等工具之后，可以编译SDK demo，生成测试的可执行文件。步骤如下。

一、将获取的序列号填入demo.cpp

```
$ cd demo # 进入demo文件夹
$ vi demo.cpp # 若vi未找到命令，执行 sudo apt install vim
```

在打开的代码编辑页面，找到

```
global_controller()->set_licence_key("set your license here");
```

将序列号填入引号内。如果想打印demo运行过程中的日志，找到

```
log_config.enable_debug = false;
```

将false改为true即可。

二、编译 在demo目录下执行

```
mkdir build # 创建build目录
cd build
cmake .. # 如果系统中安装了opencv3.4以上
##### 或者
cmake .. -DEDGE_BUILD_OPENCV=ON # 自动编译安装opencv
```

若用户需要自定义opencv library path、gcc路径等，修改CMakeList.txt即可。

当出现：

```
-- Configuring done -- Generating done -- Build files have been written to: /xxx/demo/build
```

表示cmake成功。然后执行编译

```
make # 如果有多个cpu可以用-j加快编译速度,如4个CPU用 make -j4
```

当出现：

```
[100%] Built target easyedge_serving [100%] Built target easyedge_demo
```

表示编译成功，在build目录下出现编译的产物：

- easyedge\_demo：测试的可执行文件
- easyedge\_serving：包含http server的测试的可执行文件
- thirdparty：编译安装的opencv

测试easyedge\_demo 在build目录下执行：

```
./easyedge_demo {模型RES文件夹} {测试图片路径}
```

第一个参数为包含模型的文件夹路径，第二个参数为测试的图片的路径。SDK中已经包含模型文件夹，如果用户有其他模型文件，可以指定为其路径。如：

```
./easyedge_demo ../../RES /xxx/test.jpg
```

然后可以看到输出的结果：

```
0 build > ./easyedge_demo ~/lvxiangxiang/models/SqueezeNetV1.1-tf/ ~/lvxiangxiang/images/orange.jpg
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 The recommended threshold is 0.3
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Rescale mode is 0, target_size: 0, max_size: 0
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Config file read done
2020-07-16 16:52:33,636 DEBUG [EasyEdge] 548062597120 Local license is ok.
2020-07-16 16:52:33,855 DEBUG [EasyEdge] 548062597120 Image will be resized to 227,227
2020-07-16 16:52:33,986 DEBUG [EasyEdge] 548062597120 Inference costs 131.011(127.854) ms
950, orange, p:0.972394
Done
```

如果是物体检测或者图像

分割模型，可以打开生成的/xxx/test.result.cpp.jpg图片，查看检测框的效果。

测试easyedge\_serving easyedge\_serving会开启一个http server服务，并实现了一个简单的网页，用户可以在网页上上传图片并查看预测结果。

在build目录下执行：

```
./easyedge_serving {模型RES文件夹} {序列号} {主机ip, 默认 0.0.0.0} {端口, 默认 24401}
```

如：

```
./easyedge_serving ../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

若日志显示：

```
HTTP is now serving at 0.0.0.0:24401
```

表示http server启动成功。此时可以打开浏览器，输入网址http://{设备ip}:24401，上传图片来进行测试。

查看设备ip的方法：

```
ifconfig # 如果没有ifconfig命令, 执行 sudo apt install net-tools
```

如果是有线连接, 找到eth0一栏, 如果是wifi连接, 找到wlan0一栏。

注意: 只有本机电脑和硬件设备的网络ip在同一网段之下, 才可以通过网址访问。

【图像分类】0 AlexNet-fluid



效果如下:

label	置信度
n07747607 orange	0.951
n07749582 lemon	0.030
n07716906 spaghetti squash	0.006
n03942813 ping-pong ball	0.004
n07717556 butternut squash	0.002
n03134739 croquet ball	0.001
n03929660 pick, plectrum, plectron	0.001
n04332243 strainer	0.001
n03530642 honeycomb	0.000
n12144580 corn	0.000
n04409515 tennis ball	0.000
n04019541 puck, hockey puck	0.000

## 集成SDK

SDK提供了一系列模型加载、预测等接口, 用户可以方便的集成进自己的程序之中。

接口说明、数据格式说明以及常见错误请参考SDK技术文档。

建议先测试Demo, 以及参考demo.cpp和demo的CMakeLists.txt调用流程。如果遇到错误, 优先参考文件中的注释以及日志说明。

### 一、导入SDK头文件和库文件

在baidu\_easyedge\_linux\_cpp\_aarch64\_ARM\_gcc5.4\_vx.x.x\_xxxxxx/include下有SDK的头文件。在baidu\_easyedge\_linux\_cpp\_aarch64\_ARM\_gcc5.4\_vx.x.x\_xxxxxx/lib下有SDK的库文件, 包含动态库libeasyedge.so和静态库libeasyedge\_static.a。用户可选择合适的导入方式。

用户将头文件和库文件拷贝至自己的项目中, 并在自己的CMakeLists.txt中引用:

```
find_package(OpenCV REQUIRED)

// 导入头文件
include_directories(
    ${OpenCV_INCLUDE_DIRS}
    ${CMAKE_SOURCE_DIR}/../include/
)
// 导入库文件
link_directories(
    ${CMAKE_SOURCE_DIR}/../lib/
)
// 链接库文件
target_link_libraries(your_executable_file ${OpenCV_LIBS} easyedge paddle_full_api_shared)
```

### 二、在程序中调用SDK接口

```
// 引入SDK头文件
##### include "easyedge/easyedge.h"
// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");
// step 1: 配置模型资源目录
PaddleFluidConfig config;
config.model_dir = (模型文件目录);
// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor(config);
auto img = cv::imread((测试图片路径));
// step 3: 预测图像
std::vector<EdgeResultData> result;
predictor->infer(img, result);
```

如果是口罩检测模型，将PaddleFluidConfig config修改为PaddleMultiStageConfig config。口罩检测模型请注意输入图片中人脸大小建议保持在88到9096像素之间，可根据场景远近程度缩放图片后再传入SDK。

## 通用设备端Windows x86加速版

### 简介

Windows CPU加速版SDK是适用于EasyDL图像模型快速部署的工具包。SDK中包含了EasyDL训练的模型资源文件、SDK和demo文件。

### 测试前的准备

- Windows x86的硬件及开发环境
  - 详情参考下方文档
- EasyDL平台的Windows x86 加速版SDK
  - 以图像分类为例，前往[操作台](#)训练模型后，选择发布为Windows x86的通用设备端SDK并勾选加速版，发布成功后即可从平台下载
- 用于激活通用设备端加速版SDK的序列号
  - 前往[控制台](#)申请用于激活通用设备端SDK的序列号，注意选择加速版序列号
  - 首次使用SDK或者更换序列号、更换设备时，需要联网激活。激活成功之后，有效期内可离线使用

### 安装依赖

在使用SDK之前，首先要确认自己的硬件类型和相应的依赖库安装是否已经符合要求。

#### 硬件要求：

- Intel Xeon with AVX2 and AVX512
- Intel Core Processors with AVX2
- Intel Atom Processors with SSE

#### 软件要求：

- 64位 Windows 10
- .NET Framework 4.5
- Visual C++ Redistributable Packages for Visual Studio 2013
- Visual C++ Redistributable Packages for Visual Studio 2015
- Opencv 2020.1

#### 其他要求：

- 第一次使用SDK请确保联网

### SDK结构

获取到的SDK解压后的目录结构是：

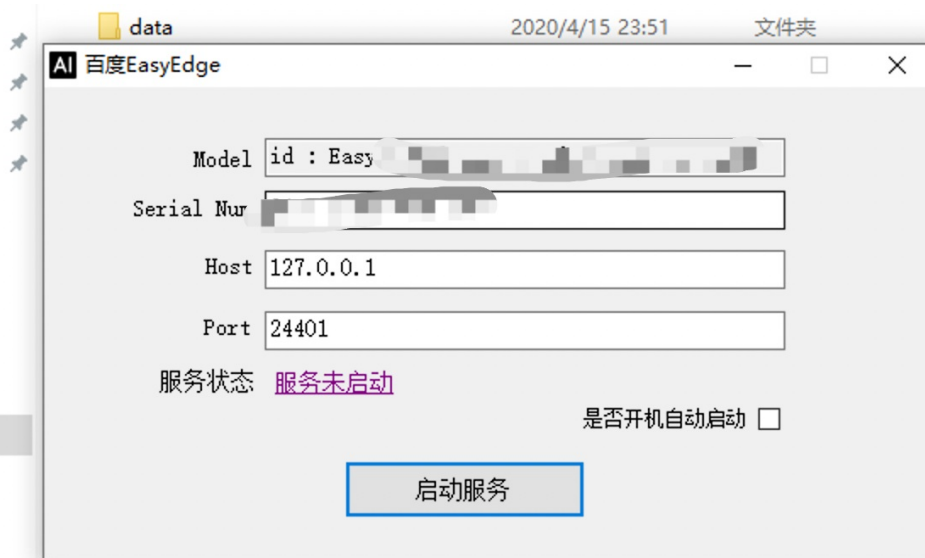


其中：

- bootstrap是SDK的入口脚本。
- data/model文件夹下是EasyDL训练得到的模型资源文件。
- tools文件夹下提供的是模型更新工具，用在迭代训练模型后，直接拉取新训练的模型，而不用重新下载SDK。

### 运行demo

打开EasyEdge.exe，输入Serial Num

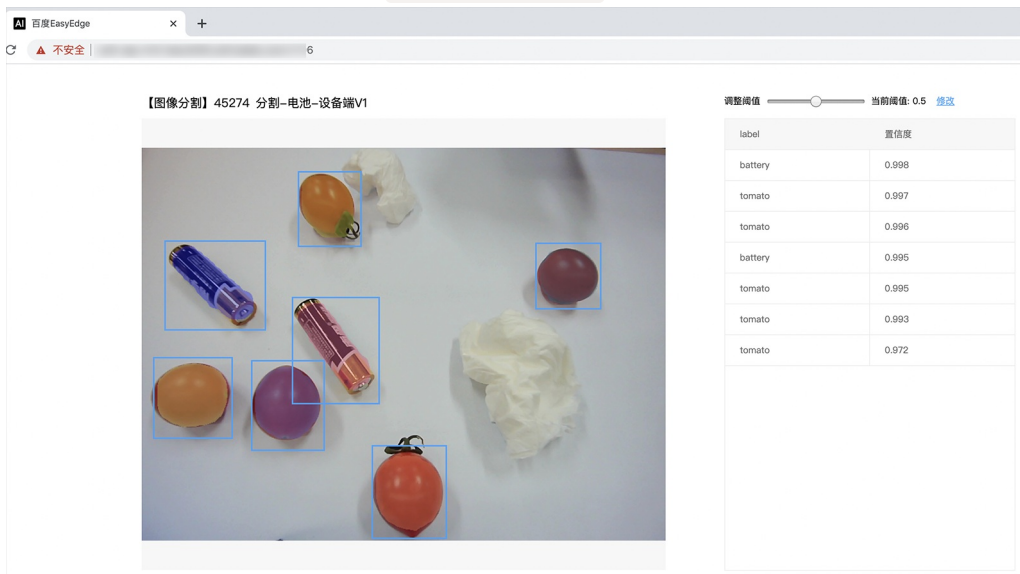


点击"启动服务"，等待数

秒即可启动成功，本地服务默认运行在

http://127.0.0.1:24401/

服务运行成功，此时可直接在浏览器中输入http://127.0.0.1:24401，在h5中测试模型效果。



### Http服务集成

服务运行成功后，除网页直接访问外，也可以通过http请求的方式执行模型的预测并获取预测结果。

### 图像服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

##### params 为GET参数 data 为POST Body
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl



```

##### include <sys/stat.h>
##### include <curl/curl.h>

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

#### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

**返回参数** | 字段 | 类型 | 取值 | 说明 | | ----- | ----- | ---- | ----- | | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 | | x1, y1 | float | 0~1 | 物体检测，矩形的左上角坐标（相对长宽的比例值） | | x2, y2 | float | 0~1 | 物体检测，矩形的右下角坐标（相对长宽的比例值） |

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

**图像分割** 返回结果格式参考API调用文档 代码参考 <https://github.com/Baidu-AIP/EasyDL-Segmentation-Demo>

## 声音服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.mp3', 'rb') as f:
    audio = f.read()

##### params 为GET参数 data 为POST Body
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=audio).json()
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./audio.mp3", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] audio = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(audio, 0, audio.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

##### include <sys/stat.h>
##### include <curl/curl.h>

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./audio.mp3";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

#### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送声音二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | | ----- | ----- | ---- | ----- | | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 |

## 服务器端Linux GPU 加速版

### 简介

Linux GPU加速版SDK是适用于EasyDL经典版、EasyDL专业版车型快速部署的工具包。SDK中包含了EasyDL训练的模型资源文件、SDK和demo文件。

### 测试前的准备

- Linux x86 GPU的硬件及开发环境
  - 详情参考下方文档

- EasyDL平台的Linux x86 GPU 加速版服务器端SDK&激活序列号
  - 以经典版图像分类为例，在[操作台](#)训练「私有服务器部署-服务器端SDK」下的模型后，前往[控制台](#)申请发布Linux x86 GPU的服务器端SDK，发布成功后即可同时获得加速版SDK和序列号
  - 首次使用SDK或者更换序列号、更换设备时，需要联网激活。激活成功之后，有效期内可离线使用

## 安装依赖

在使用SDK之前，首先要确认自己的硬件类型和相应的依赖库安装是否已经符合要求。

### 硬件要求：

- Linux GPU加速版SDK支持绝大部分Nvidia显卡。

### 软件要求：

- gcc 5.4+ (需包含GLIBCXX\_3.4.22)
- cmake 3+
- CUDA 9.0 + cuDNN 7.5 或 CUDA 10.0 + cuDNN 7.5
- TensorRT 7.0
- OpenCV 3.4

### 其他要求：

- 第一次使用SDK请确保联网

## SDK结构

获取到的SDK解压后的目录结构是：



其中：

- cpp文件夹下有两个压缩包分别为适配不同版本CUDA的SDK，解压缩后得到include头文件、lib库文件和demo示例代码文件。
- RES文件夹下是EasyDL训练得到的模型资源文件。
- tools文件夹下提供的是模型更新工具，用在迭代训练模型后，直接拉取新训练的模型，而不用重新下载SDK。

## 编译demo

解压SDK后，进入到demo目录可以直接编译demo。编译方法如下：

```
##### 1. 首先进入demo目录
##### 2. 创建build文件夹并进入
$ mkdir build && cd build
##### 3. 执行编译
$ cmake ..
$ make -j3
##### 4. 执行安装，把lib文件安装到系统路径，需要sudo权限；也可以选择不执行安装，把lib路径加为环境变量即可
$ sudo make install
##### 5. 这时候应该会产出demo编译的可执行文件，直接执行可以查看需要的参数
$ ./easyedge_batch_inference {res_dir} {image/image_dir}
##### 或
$ ./easyedge_multi_thread {res_dir} {image/image_dir}
##### 或
$ ./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

编译产出有三个可执行文件，分别为：

- easyedge\_batch\_inference：提供了对批量图片预测的能力，预测速度快，是最推荐使用的方式。

- `easymulti_thread`：提供了对多卡或多线程预测的支持。其中多线程的支持是针对一张显卡而言的，这种方式通常并不推荐使用，因为计算资源的相互竞争会拉慢单次预测的速度并且预测时间不稳定。
- `easymulti_serving`：提供了http服务的能力。启动服务后，可以轻松把http服务接口集成到自己的应用中。

其中前两个demo示例了SDK API的集成方式，第三个demo示例了如何使用SDK创建http服务。根据个人需求选择不同集成方式即可，接下来将分别介绍这两种集成方式。

## API集成

使用SDK的API接口方式能提供功能更丰富、预测速度更快的能力。API接口的设计尽可能的降低了调用复杂度，可以很方便的集成到自己的应用当中。

### 调用流程

```
// 1. 设置序列号
global_controller()->set_licence_key("ABCD-ABCD-ABCD-ABCD");
// 2. 配置运行选项
TensorRTConfig config;
config.model_dir = argv[1];
config.device = 0; // 设置需要使用的GPU
config.max_batch_size = 4; // 优化的模型可以支持的最大batch_size，实际单次推理的图片数不能大于此值
config.max_concurrency = 1; // 设置device对应的卡可以使用的最大线程数
config.fp16 = false; // 置true开启fp16模式推理会更快，精度会略微降低，但取决于硬件是否支持fp16，不是所有模型都支持fp16，参阅文档
// 3. 创建predictor并初始化
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
// 4. 执行预测
predictor->infer(imgs, result);
```

当机器上存在多张显卡时，可以创建多个predictor，每一个predictor通过config配置不同的显卡id，这样就可以同时在不同的显卡上执行预测。具体使用方式可以参考demo\_multi\_thread.cpp的示例代码。这里贴出：

```

// 配置要使用的显卡id
std::vector<int> devices(0, 1, 2, 3);
// 每个predictor对应一张显卡
std::vector<std::unique_ptr<EdgePredictor>> predictors;
// 配置运行选项
TensorRTConfig config;
config.model_dir = model_dir;
config.max_batch_size = 8; // 单次预测可以支持的最大batch_size, 多个predictor的此值要保持一致
config.max_concurrency = 1; // 单个device上的最大infer并发量, 建议保持为1
config.compile_level = 1; // 编译模型的策略, 如果当前设置的max_batch_size与历史编译存储的不同, 则重新编译模型
config.fp16 = false; // 置true开启fp16模式推理会更快, 精度会略微降低, 但取决于硬件是否支持fp16, 且不是所有模型都支持fp16,
参阅开发文档
//
for(int i = 0; i < predictors.size(); ++i) {
    config.device = devices[i]; // 设置GPU id
    // 创建predictor
    auto predictor = global_controller()->CreateEdgePredictor<TensorRTConfig>(config);
    if (predictor->init() != EDGE_OK) {
        exit(-1);
    }
    predictors.emplace_back(std::move(predictor));
}

// 创建与predictor数量等同线程数的infer_task
std::vector<std::thread> threads;
std::map<int /*predictor index*/, std::vector<std::vector<EdgeResultData>>> results;
for (int i = 0; i < predictors.size(); ++i) {
    assert(split_img_files[i].size() <= config.max_batch_size);
    // infer_task的实现详见demo_multi_thread.cpp, 主要功能为执行infer预测
    threads.emplace_back(std::thread(infer_task, std::ref(predictors[i]), std::ref(split_img_files[i]), std::ref(results), i));
}

for (auto& t : threads) {
    if (t.joinable()) {
        t.join();
    }
}
}

```

通过创建多个predictor可以将同一个模型同时部署到多张显卡上, 这样可以提升执行预测的吞吐率。当然也可以在不同的显卡上部署不同的模型, 通过在config中指定不同的model\_dir即可。

可以看到, 调用SDK的API流程很简单, 并可以通过config的不同配置实现不同的能力。比如支持批量图片预测、多卡部署、多线程预测、fp16加速等。实际集成的过程中, 最需要注意的就是config的配置, 一个适合自己应用场景的config参数配置才能带来最佳的预测速度。

**参数配置** config的定义可以参见头文件easyedge.h

```

struct TensorRTConfig : public EdgePredictorConfig {
    std::string model_filename{"model"};
    std::string params_filename{"params"};
    std::string cache_name{"m_cache"};
    /**
     * @brief GPU工作空间大小设置
     * workspace_size = workspace_prefix * (1 << workspace_offset)
     * workspace_offset: 10 = KB, 20 = MB, 30 = GB
     */
    int workspace_prefix{16};
    int workspace_offset{20};
    /**
     * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值
     */
    int max_batch_size{1};
    /**
     * @brief 设置使用哪张 GPU 卡
     */
    int device{0};
    /**
     * @brief 模型编译等级
     * 0：无论当前设置的max_batch_size是多少，仅使用历史编译产出（如果存在）
     * 1：如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）
     * 2：无论历史编译产出的max_batch_size为多少，均根据当前max_batch_size重新编译模型
     */
    int compile_level{1};
    /**
     * @brief 设置device对应的卡可以支持的最大并发量
     * 实际预测的时候对应卡的最大并发量不超过这里设置的范围
     */
    int max_concurrency{1};
    /**
     * @brief 是否开启fp16模式预测，需要硬件支持
     */
    bool fp16{false};
    /**
     * @brief 设置需要使用的DLA Core
     */
    int dla_core{-1};
}

```

**cache\_name**：GPU加速版SDK首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。比如，可以根据不同的配置来命名产出文件，从而可以在之后的运行中直接通过文件名就可以索引到编译产出，而不用再次执行优化过程。

**workspace\_size**：设置运行时可以被用来使用的最大临时显存。通常默认即可，但当执行预测失败时，可以适当调大此值。

**max\_batch\_size**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数不可大于此值，但可以不大于此值的任意图片数。建议设置为与实际需要批量预测的图片数量保持一致，以节省内存资源并可获得较高预测速度。

**device**：设置需要使用的 GPU 卡号。

**compile\_level**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 **max\_batch\_size** 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 **compile\_level** 来控制，当此值为 0 时，表示忽略当前设置的 **max\_batch\_size** 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 **max\_batch\_size** 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**max\_concurrency**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度且预测速度不稳定，建议优先考虑 batch inference。

**fp16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持fp16的模式包括：EasyDL经典版图像分类高精度模型。

运行demo 命令执行格式：



```
./easyedge_batch_inference {模型RES文件夹} {测试图片路径或图片文件夹路径}
./easyedge_multi_thread {模型RES文件夹} {测试图片路径或图片文件夹路径}
```

如：



预测效果展示：



## Http服务集成

使用SDK的http服务API可以创建一个http服务端口，从而可以在应用里或者网页中通过http请求的方式执行模型的预测并获取预测结果。

### 调用流程

```
// 1. 设置序列号
global_controller()->set_licence_key("ABCD-ABCD-ABCD-ABCD");
// 2. 配置运行选项
TensorRTConfig config;
config device = 0;
config model_dir = model_dir;
// 3. 启动服务
return global_controller()->start_http_server(config, host, port, service_id, 1);
```

编译并运行后就会在控制台启动一个服务，默认地址为：`http://{设备ip}:24401`，通过在浏览器访问此地址可以打开一个测试页面，上传图片即可获取识别结果。

### 运行demo demo命令执行格式

```
./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

如：



当出现提示：`HTTP is now serving at 0.0.0.0:24401, holding 1 instances`时，表示服务已经启动成功，这时候可以在浏览器里输入地址打开一个测试页面。如下所示：



上传图片，可以看到预测效果：



也可以通过在自己的项目中请求此地址做预测，并获取预测结果。URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

如使用Python的请求示例：

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

http请求的返回格式为：

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考开发文档
cost_ms	Number	预测耗时ms, 不含网络交互时间

## 专项适配硬件EdgeBoard(FZ)

### 简介

本文将为新手提供一个快速测试和集成EasyDL & EasyEdge的 Linux EdgeBoard-FZ SDK的图文教程。

### 测试前的准备

- EdgeBoard(FZ)硬件及开发环境
  - 详情参考下方文档
- EasyDL平台的EdgeBoard(FZ)专用SDK
  - 以图像分类为例, 前往[操作台](#)训练「专项硬件适配SDK-EdgeBoard(FZ)」下的模型并发布SDK后, 即可从平台下载
- 用于激活专用SDK的序列号
  - 前往[控制台](#)申请用于激活EdgeBoard(FZ)专用SDK的序列号
  - 首次使用SDK或者更换序列号、更换设备时, 需要联网激活。激活成功之后, 有效期内可离线使用

### 效果展示

```
0 build > ./easyedge_demo ~/lvxiangxiang/models/SqueezeNetV1.1-tf/ ~/lvxiangxiang/images/orange.jpg
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 The recommended threshold is 0.3
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Rescale mode is 0, target_size: 0, max_size: 0
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Config file read done
2020-07-16 16:52:33,636 DEBUG [EasyEdge] 548062597120 Local license is ok.
2020-07-16 16:52:33,855 DEBUG [EasyEdge] 548062597120 Image will be resized to 227,227
2020-07-16 16:52:33,986 DEBUG [EasyEdge] 548062597120 Inference costs 131.011(127.854) ms
950, orange, p:0.972394
Done
```

【图像分类】0 AlexNet-fluid



label	置信度
n07747607 orange	0.951
n07749582 lemon	0.030
n07716906 spaghetti squash	0.006
n03942813 ping-pong ball	0.004
n07717556 butternut squash	0.002
n03134739 croquet ball	0.001
n03929660 pick, plectrum, plectrum	0.001
n04332243 strainer	0.001
n03530642 honeycomb	0.000
n12144580 corn	0.000
n04409515 tennis ball	0.000
n04019541 puck, hockey puck	0.000

## 环境准备

**硬件环境** EdgeBoard-FZ系列计算盒，包括ZU3/ZU5/ZU9。

EdgeBoard-FZ系列硬件购买和详细硬件参数请前往[AI市场](#)。

使用SDK时，请保持EdgeBoard-FZ内核为最新，否则运行可能出现错误。

EdgeBoard-FZ计算盒使用手册：<https://ai.baidu.com/ai-doc/HWCE/Yk3b86gvp>

EdgeBoard-FZ内核更新地址：<https://ai.baidu.com/ai-doc/HWCE/Yk3b95s8o> **网络环境** 用户安装软件和测试SDK都需要联网，用户使用的EdgeBoard-FZ需确保有效的网络连接。 **软件环境** 需要在EdgeBoard-FZ计算盒上安装以下软件和第三方库以保证SDK正常编译和运行：

- cmake 3 +
- gcc 5.4 +
- opencv3.4 (可选)

可以使用以下方法确认cmake是否满足要求:

```
$ cmake --version
cmake version 3.13.3
```

若系统提示找不到cmake命令或者cmake version 低于3.x.x，则需要安装/升级cmake。

```
// apt安装cmake
sudo apt update
sudo apt install cmake
```

ubuntu官方的apt源update较慢，且可能访问不了，可以替换为国内的源：<https://www.cnblogs.com/yongy1030/p/10315569.html>

也可以使用源码编译的方式安装，参考[安装方法](#)。

```
// 升级cmake
sudo apt-get install software-properties-common
sudo add-apt-repository ppa:george-edison55/cmake-3.x
sudo apt-get update
sudo apt-get upgrade
```

安装/升级cmake后可再次执行cmake --version确认版本。

可以使用以下方法确认gcc是否满足要求:

```
$ gcc --version
gcc (Ubuntu/Linaro 5.4.0-6ubuntu1~16.04.12) 5.4.0 20160609
```

若系统提示找不到gcc命令或者gcc version 低于5.4.0，则需要安装/升级gcc。

```
// 安装gcc
sudo apt update
sudo apt install build-essential
// 升级gcc
sudo add-apt-repository ppa:ubuntu-toolchain-r/test # 如果找不到add-apt-repository命令，执行：apt-get install software-properties-common
sudo apt-get update
sudo apt-get install -y gcc-5 g++-5

cd /usr/bin # 升级gcc 5之后，还需要替换原来的软链接
sudo rm -r gcc # 移除之前的软连接
sudo ln -sf gcc-5 gcc # 建立gcc5的软连接
sudo rm -r g++ # 同gcc
sudo ln -sf g++-5 g++
```

安装/升级gcc后可再次执行gcc --version确认版本。

目前没有合适的方法确认系统中是否有SDK需要的OpenCV，若用户不确定是否安装OpenCV 3.4 +，并且可以被cmake find\_package到，可以手动编译安装OpenCV 3.4，也可以在之后编译SDK时自动编译OpenCV。

若选择在下一步编译EasyEdge SDK时自动编译OpenCV，则以下编译安装OpenCV的步骤可跳过。

下载OpenCV 3.4源代码包并解压：[下载地址](#)，然后编译安装：

```
// 编译安装OpenCV
cd opencv-3.4.6
mkdir build
cd build

cmake .. -DBUILD_DOCS=OFF -DBUILD_EXAMPLES=OFF -DBUILD_opencv_python2=OFF -DBUILD_opencv_python3=OFF -
DBUILD_WITH_DEBUG_INFO=OFF -DBUILD_PACKAGE=OFF -DBUILD_opencv_core=ON -DBUILD_opencv_imgproc=ON -
DBUILD_opencv_imgcodecs=ON -DBUILD_opencv_highgui=ON -DBUILD_opencv_video=OFF -DBUILD_opencv_videoio=OFF -
DBUILD_opencv_dnn=OFF -DBUILD_opencv_apps=OFF -DBUILD_opencv_flann=OFF -DBUILD_opencv_gpu=OFF -DBUILD_opencv_ml=OFF -
DBUILD_opencv_legacy=OFF -DBUILD_opencv_calib3d=OFF -DBUILD_opencv_features2d=OFF -DBUILD_opencv_java=OFF -
DBUILD_opencv_objdetect=OFF -DBUILD_opencv_photo=OFF -DBUILD_opencv_nonfree=OFF -DBUILD_opencvocl=OFF -
DBUILD_opencv_stitching=OFF -DBUILD_opencv_superres=OFF -DBUILD_opencv_ts=OFF -DBUILD_opencv_videostab=OFF -
DBUILD_opencv_contrib=OFF -DBUILD_SHARED_LIBS=ON -DBUILD_TESTS=OFF -DBUILD_PERF_TESTS=OFF -DBUILD_WITH_CAROTENE=OFF -
DCMAKE_BUILD_TYPE:STRING=Release -DWITH_FFMPEG=OFF -DWITH_IPP=OFF -DBUILD_PNG=ON -DBUILD_JPEG=ON -DBUILD_ZLIB=ON -
DBUILD_FAT_JAVA_LIB=OFF -DOPENCV_CXX11=OFF -DCMAKE_INSTALL_PREFIX:PATH=/usr/lib/aarch64-linux-gnu/

make # 如果有多个cpu可以用-j加快编译速度,如4个CPU用 make -j4
make install
```

## 启动EdgeBoard-FZ计算盒

### 一、将计算盒连接电源

指示灯亮起，等待约1分钟。

### 二、连接计算盒

参考[EdgeBoard-FZ使用手册](#)配置网口或串口连接，登录EdgeBoard-FZ计算盒。

### 三、加载驱动

开机加载一次即可。

```
insmod /home/root/workspace/driver/{zu9|zu5|zu3}/fpgadv.ko
```

根据计算盒的版本（zu9/zu5/zu3）选择驱动。若未加载驱动，SDK可能报错：

```
Failed to to fpga device: -1
```

### 四、设置系统时间

系统时间必须正确。

```
date --set "2019-5-18 20:48:00"
```

## 测试demo

**SDK介绍** 用户下载的Linux EdgeBoard-FZ SDK zip包中包含SDK动态库、模型等资源文件和测试demo.cpp。

需要将SDK zip包完整的放入EdgeBoard-FZ 硬件上再进行解压，否则可能会报错：

```
libeasyedge.so: file format not recognized; treating as linker script
```

```
Linux下解压命令：tar -xvf xxx.tar、unzip xxx.zip
```

SDK zip包的目录结构如下：

```
EasyEdge-Linux-mxxx-bxxx-edgeboard
├── cpp
│   ├── baidu_easyedge_linux_cpp_aarch64_PADDLEMOBILE_FPGA_gcc5.4_vx.x.x_xxxxxx.tar
│   └── RES # 模型、标签和配置文件
```

解压baidu\_easyedge\_linux\_cpp\_aarch64\_PADDLEMOBILE\_FPGA\_gcc5.4\_vx.x.x\_xxxxxx.tar之后的目录结构如下：

```
baidu_easyedge_linux_cpp_aarch64_PADDLEMOBILE_FPGA_gcc5.4_vx.x.x_xxxxxx
├── demo # 测试demo
│   ├── CMakeLists.txt
│   ├── demo.cpp
│   ├── opencv.cmake
│   └── easyedge_serving
├── include # SDK需要的头文件
│   ├── easyedge
│   └── easyedge.h
├── lib # SDK需要的库文件
│   ├── libeasyedge.so -> libeasyedge.so.x.x.x
│   ├── libeasyedge.so.x.x.x
│   ├── libeasyedge_static.a
│   ├── libpaddle-mobile.so -> libpaddle-mobile.so.x.x.x
│   ├── libpaddle-mobile.so.x.x.x
│   └── libverify.so
└── ReadMe.txt # 文档等其他说明
```

**编译demo** 前面安装了cmake、gcc等工具之后，可以编译SDK demo，生成测试的可执行文件。步骤如下。

一、将获取的序列号填入demo.cpp

```
cd demo # 进入demo文件夹
vi demo.cpp # 若vi未找到命令，执行 sudo apt install vim
```

在打开的代码编辑页面，找到

```
global_controller()->set_licence_key("set your license here");
```

将序列号填入引号内。如果想打印demo运行过程中的日志，找到

```
log_config.enable_debug = false;
```

将false改为true即可。

二、编译 在demo目录下执行

```
mkdir build # 创建build目录
cd build
cmake .. # 如果系统中安装了opencv3.4以上
##### 或者
cmake .. -DEDGE_BUILD_OPENCV=ON # 自动编译安装opencv
```

若用户需要自定义opencv library path、gcc路径等，修改CMakeList.txt即可。

当出现：

```
-- Configuring done -- Generating done -- Build files have been written to: /xxx/demo/build
```

表示cmake成功。然后执行编译

```
make # 如果有多个cpu可以用-j加快编译速度,如4个CPU用 make -j4
```

当出现：

```
[100%] Built target easyedge_serving [100%] Built target easyedge_demo
```

表示编译成功，在build目录下出现编译的产物：

- easyedge\_demo：测试的可执行文件
- easyedge\_serving：包含http server的测试的可执行文件
- thirdparty：编译安装的opencv

测试easyedge\_demo 在build目录下执行：

```
./easyedge_demo {模型RES文件夹} {测试图片路径}
```

第一个参数为包含模型的文件夹路径，第二个参数为测试的图片的路径。SDK中已经包含模型文件夹，如果用户有其他模型文件，可以指定为其路径。如：

```
./easyedge_demo ../.././RES /xxx/test.jpg
```

然后可以看到输出的结果：

```
0 build > ./easyedge_demo ~/lvxiangxiang/models/SqueezeNetV1.1-tf/ ~/lvxiangxiang/images/orange.jpg
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 The recommended threshold is 0.3
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Rescale mode is 0, target_size: 0, max_size: 0
2020-07-16 16:52:33,632 DEBUG [EasyEdge] 548062597120 Config file read done
2020-07-16 16:52:33,636 DEBUG [EasyEdge] 548062597120 Local license is ok.
2020-07-16 16:52:33,855 DEBUG [EasyEdge] 548062597120 Image will be resized to 227,227
2020-07-16 16:52:33,986 DEBUG [EasyEdge] 548062597120 Inference costs 131.011(127.854) ms
950, orange, p:0.972394
Done
```

如果是物体检测或者图像

分割模型，可以打开生成的/xxx/test.result.cpp.jpg图片，查看检测框的效果。

如果用户使用的是ZU5，且执行过程中出现内存不足：Killed。这是因为FZ5A带vcu，给它预留的内存过大导致，如果用不到VCU可以把这部分改小。修改/run/media/mmcbk1p1/uEnv.txt：

```
ethaddr=00:0a:35:00:00:09
uenvcmd=fatload mmc 1 0x3000000 image.ub && bootm 0x3000000

bootargs=earlycon console=ttyPS0,115200 clk_ignore_unused cpuidle.off=1 root=/dev/mmcbk1p2 rw rootwait cma=128M
```

注意中间空行要保留。

如果预测结果明显错误或者执行过程报错，请检查内核是否为最新版本。测试easyedge\_serving easyedge\_serving会开启一个http server服务，并实现了一个简单的网页，用户可以在网页上上传图片并查看预测结果。

在build目录下执行：

```
./easyedge_serving {模型RES文件夹} {序列号} {主机ip, 默认 0.0.0.0} {端口, 默认 24401}
```

如：

```
./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401
```

若日志显示：

```
HTTP is now serving at 0.0.0.0:24401
```

表示http server启动成功。此时可以打开浏览器，输入网址http://{设备ip}:24401，上传图片来进行测试。

查看设备ip的方法：

```
ifconfig # 如果没有ifconfig命令，执行 sudo apt install net-tools
```

找到eth0一栏。

注意：只有本机电脑和硬件设备的网络ip在同一网段之下，才可以透过网址访问。

【图像分类】0 AlexNet-fluid



效果如下：

label	置信度
n07747607 orange	0.951
n07749582 lemon	0.030
n07716906 spaghetti squash	0.006
n03942813 ping-pong ball	0.004
n07717556 butternut squash	0.002
n03134739 croquet ball	0.001
n03929660 pick, plectrum, plectron	0.001
n04332243 strainer	0.001
n03530642 honeycomb	0.000
n12144580 corn	0.000
n04409515 tennis ball	0.000
n04019541 puck, hockey puck	0.000

## 集成SDK

SDK提供了一系列模型加载、预测等接口，用户可以方便的集成进自己的程序之中。

接口说明、数据格式说明以及常见错误请参考SDK技术文档。

建议先测试Demo，以及参考demo.cpp和demo的CMakeLists.txt调用流程。如果遇到错误，优先参考文件中的注释以及日志说明。

### 一、导入SDK头文件和库文件

在baidu\_easyedge\_linux\_cpp\_aarch64\_PADDLEMOBILE\_FPGA\_gcc5.4\_vx.x.x\_xxxxxx/include下有SDK的头文件。在baidu\_easyedge\_linux\_cpp\_aarch64\_PADDLEMOBILE\_FPGA\_gcc5.4\_vx.x.x\_xxxxxx/lib下有SDK的库文件，包含动态库libeasyedge.so和静态库libeasyedge\_static.a。用户可选择合适的导入方式。

用户将头文件和库文件拷贝至自己的项目中，并在自己的CMakeLists.txt中引用：

```
find_package(OpenCV REQUIRED)

// 导入头文件
include_directories(
    ${OpenCV_INCLUDE_DIRS}
    ${CMAKE_SOURCE_DIR}/../include/
)
// 导入库文件
link_directories(
    ${CMAKE_SOURCE_DIR}/../lib/
)
// 链接库文件
target_link_libraries(your_executable_file ${OpenCV_LIBS} easyedge paddle-mobile)
```

### 二、在程序中调用SDK接口



```
// 引入SDK头文件
##### include "easyedge/easyedge.h"
// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");
// step 1: 配置模型资源目录
PaddleFluidConfig config;
config.model_dir = (模型文件目录);
// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor(config);
auto img = cv::imread((测试图片路径));
// step 3: 预测图像
std::vector<EdgeResultData> result;
predictor->infer(img, result);
```

目前EdgeBoard暂不支持并行多模型计算。

## 专项适配硬件Jetson

### 简介

Jetson SDK是适用于EasyDL图像模型快速部署的工具包。SDK中包含了EasyDL训练的模型资源文件、SDK和demo文件。

### 测试前的准备

- Jetson(Nano/TX2/Xavier)硬件及开发环境
  - 详情参考下方文档
- EasyDL平台的Jetson专用SDK
  - 以图像分类为例，前往[操作台](#)训练「专项硬件适配SDK-Jetson」下的模型并发布SDK后，即可从平台下载
- 用于激活专用SDK的序列号
  - 前往[控制台](#)申请用于激活Jetson专用SDK的序列号
  - 首次使用SDK或者更换序列号、更换设备时，需要联网激活。激活成功之后，有效期内可离线使用

### 安装依赖

在使用SDK之前，首先要确认自己的硬件类型和相应的依赖库安装是否已经符合要求。

#### 硬件要求：

Jetson系列开发板：

- Jetson Nano
- Jetson TX2
- Jetson Xavier NX
- Jetson Xavier

#### 软件要求：

- JetPack4.2.2
- JetPack4.4

#### 其他要求：

- 第一次使用SDK请确保联网

JetPack的安装需要借助SDK Manager，安装过程参考[Install Jetson Software with SDK Manager](#)。

对于Jetson Nano和Xavier NX还可以使用Etcher将系统镜像烧录到micro SD Card的形式，这种方式更简单一些。

### SDK结构

获取到的SDK解压后的目录结构是：



其中：

- cpp文件夹下有两个压缩包分别为适配不同版本JetPack的SDK，解压缩后得到include头文件、lib库文件和demo示例代码文件。
- RES文件夹下是EasyDL训练得到的模型资源文件。

## 编译demo

解压SDK后，进入到demo目录可以直接编译demo。编译方法如下：

```
##### 1. 首先进入demo目录
##### 2. 创建build文件夹并进入
$ mkdir build && cd build
##### 3. 执行编译
$ cmake ..
$ make -j3
##### 4. 执行安装，把lib文件安装到系统路径，需要sudo权限；也可以选择不执行安装，把lib路径加为环境变量即可
$ sudo make install
##### 5. 这时候应该会产出demo编译的可执行文件，直接执行可以查看需要的参数
$ ./easyedge_batch_inference {res_dir} {image/image_dir}
##### 或
$ ./easyedge_multi_thread {res_dir} {image/image_dir}
##### 或
$ ./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

编译产出有三个可执行文件，分别为：

- easyedge\_batch\_inference：提供了对批量图片预测的能力，预测速度快，是最推荐使用的方式。
- easyedge\_multi\_thread：提供了对多线程预测的支持。但是这种方式通常并不推荐使用，因为多线程下计算资源的相互竞争会拉慢单次预测的速度并且预测时间不稳定，建议优先考虑使用batch inference的方式。
- easyedge\_serving：提供了http服务的能力。启动服务后，可以在浏览器访问测试页面，或轻松把http服务接口集成到自己的应用中。

其中前两个demo示例了SDK API的集成方式，第三个demo示例了如何使用SDK创建http服务。根据个人需求选择不同集成方式即可，接下来将分别介绍这两种集成方式。

## API集成

使用SDK的API接口方式能提供功能丰富、预测速度快的能力。API接口的设计尽可能的降低了调用复杂度，可以很方便的集成到自己的应用中。

### 调用流程

```
// 1. 设置序列号
global_controller()->set_licence_key("ABCD-ABCD-ABCD-ABCD");
// 2. 配置运行选项
TensorRTConfig config;
config.model_dir = argv[1];
config.max_batch_size = 4; // 优化的模型可以支持的最大batch_size，实际单次推理的图片数不能大于此值
config.max_concurrency = 1; // 设置device对应的卡可以使用的最大线程数
config.fp16 = false; // 置true开启fp16模式推理会更快，精度会略微降低，但取决于硬件是否支持fp16，不是所有模型都支持fp16，参阅文档
// 3. 创建predictor并初始化
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
// 4. 执行预测
predictor->infer(imgs, result);
```

可以看到，调用SDK的API流程很简单，并可以通过config的不同配置实现不同的能力。比如支持批量图片预测、多线程预测、fp16加速等。实际集成的过程中，最需要注意的就是config的配置，一个适合自己应用场景的config参数配置才能带来最佳的预测速度。

**参数配置** config的定义可以参见头文件easyedge.h

```

struct TensorRTConfig : public EdgePredictorConfig {
    std::string model_filename{"model"};
    std::string params_filename{"params"};
    std::string cache_name{"m_cache"};
    /**
     * @brief GPU工作空间大小设置
     * workspace_size = workspace_prefix * (1 << workspace_offset)
     * workspace_offset: 10 = KB, 20 = MB, 30 = GB
     */
    int workspace_prefix{16};
    int workspace_offset{20};
    /**
     * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值
     */
    int max_batch_size{1};
    /**
     * @brief 设置使用哪张 GPU 卡
     */
    int device{0};
    /**
     * @brief 模型编译等级
     * 0：无论当前设置的max_batch_size是多少，仅使用历史编译产出（如果存在）
     * 1：如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）
     * 2：无论历史编译产出的max_batch_size为多少，均根据当前max_batch_size重新编译模型
     */
    int compile_level{1};
    /**
     * @brief 设置device对应的卡可以支持的最大并发量
     * 实际预测的时候对应卡的最大并发量不超过这里设置的范围
     */
    int max_concurrency{1};
    /**
     * @brief 是否开启fp16模式预测，需要硬件支持
     */
    bool fp16{false};
    /**
     * @brief 设置需要使用的DLA Core
     */
    int dla_core{-1};
}

```

**cache\_name**：GPU加速版SDK首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。比如，可以根据不同的配置来命名产出文件，从而可以在之后的运行中直接通过文件名就可以索引到编译产出，而不用再次执行优化过程。

**workspace\_size**：设置运行时可以被用来使用的最大临时显存。通常默认即可，但当执行预测失败时，可以适当调大此值。

**max\_batch\_size**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数不可大于此值，但可以是小于此值的任意图片数。建议设置为与实际需要批量预测的图片数量保持一致，以节省内存资源并可获得较高预测速度。

**device**：设置需要使用的 GPU 卡号。

**compile\_level**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 max\_batch\_size 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 compile\_level 来控制，当此值为 0 时，表示忽略当前设置的 max\_batch\_size 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 max\_batch\_size 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**max\_concurrency**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度且预测速度不稳定，建议优先考虑 batch inference。

**fp16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持fp16的模式包括：EasyDL图像分类高精度模型。

**运行demo** 命令执行格式：

```
./easyedge_batch_inference {模型RES文件夹} {测试图片路径或图片文件夹路径}
./easyedge_multi_thread {模型RES文件夹} {测试图片路径或图片文件夹路径}
```

如：



预测效果展示：



## Http服务集成

使用SDK的http服务API可以创建一个http服务端口，从而可以在应用里或者网页中通过http请求的方式执行模型的预测并获取预测结果。

### 调用流程

```
// 1. 设置序列号
global_controller()->set_licence_key("ABCD-ABCD-ABCD-ABCD");
// 2. 配置运行选项
TensorRTConfig config;
config.model_dir = model_dir;
// 3. 启动服务
return global_controller()->start_http_server(config, host, port, service_id, 1);
```

编译并运行后就会在控制台启动一个服务，默认地址为：`http://{设备ip}:24401`，通过在浏览器访问此地址可以打开一个测试页面，上传图片即可获取识别结果。

### 运行demo demo命令执行格式

```
./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

如：



当出现提示：`HTTP is now serving at 0.0.0.0:24401, holding 1 instances`时，表示服务已经启动成功，这时候可以在浏览器里输入地址打开一个测试页面。如下所示：



上传图片，可以看到预测效果：



也可以通过在自己的项目中请求此地址做预测，并获取预测结果。URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

如使用Python的请求示例：

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

http请求的返回格式为：

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考开发文档
cost_ms	Number	预测耗时ms, 不含网络交互时间

## 开发板使用技巧

**查询L4T或JetPack版本** 有时候我们可能不记得自己的板子刷的是哪个JetPack版本了，可以通过下面这条命令查询L4T的版本。

```
##### 在终端输入如下命令并回车
$ head -n 1 /etc/nv_tegra_release
##### 就会输出类似如下结果
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看，我的板子当前的L4T版本为R32.4.3，对应JetPack4.4。但是，L4T的版本不是JetPack的版本，不过一般可以从L4T的版本唯一对应到JetPack的版本，下面列出了最近几个版本的对应关系：

```
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下，执行AI推理的速度是不一样的，如果对速度需求很高，可以把功率开到最大，但记得加上小风扇散热~

```
##### 1. 运行下面这条命令可以查询开发板当前的运行功率模式
$ sudo nvpmode -q verbose
##### $ NV Power Mode: MAXN
##### $ 0
##### 如果输出为MAXN代表是最大功率模式

##### 2. 若需要把功率调到最大，运行下面这条命令
$ sudo nvpmode -m 0

##### 如果你进入了桌面系统，也可以在桌面右上角有个按钮可以切换模式

##### 3. 查询资源利用率
$ sudo tegrastats
```

## 价格说明

### EasyDL图像价格说明

#### EasyDL图像本地服务器部署价格说明

EasyDL图像支持本地服务器部署的任务类型包括：图像分类、物体检测、图像分割，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

如免费试用期结束后希望购买可在[控制台](#)在线按设备使用年限购买授权。

## 🔗 价格参考

部署类型	按年授权价格（每个模型每年在每个设备）	永久授权价格（每个模型在每个设备）
本地服务器SDK	10000元	50000元
本地服务器API	10000元	50000元

**说明1：**本地服务器SDK与本地服务器API在适配硬件上的评测信息已上线，点击查看性能对比：[图像分类](#)、[物体检测](#)、[图像分割](#)

**说明2：**本地服务器部署的授权粒度为每个模型每年每台设备，例如购买了2个模型3年在1台设备的授权，价格为2（个模型）x3（年）x1（台）x10000元=60000元。可在一台设备上激活2个有效期三年的模型。

## EasyDL图像软硬一体方案价格说明

目前EasyDL图像提供基于图像分类、物体检测任务的多种软硬一体方案，请前往[专题页面](#)对比不同方案的性能与价格，选择与业务场景最匹配的方案。

## 🔗 方案获取流程

- Step 1：在EasyDL训练专项适配硬件的图像分类/物体检测模型，迭代模型至效果满足业务需求
- Step 2：发布模型时选择专项适配硬件的专用SDK，并在[AI市场](#)购买方案
- Step 3：获得硬件和用于激活专用SDK的专用序列号，参考文档集成后，即可实现离线AI预测

如有其他硬件方案需求，请[提交工单](#)咨询。

## EasyDL图像通用小型设备部署价格说明

EasyDL图像支持通用小型设备部署的任务类型包括：图像分类、物体检测、图像分割，每个模型发布设备端SDK后需通过序列号激活使用，每发布一个模型即可申请2个测试序列号供3个月内免费试用。

如免费试用期结束后希望购买可在[控制台](#)在线按设备购买授权或按产品线购买。

## 🔗 价格参考

### 按设备购买永久授权

授权终端设备数量	单设备授权价格（基础版SDK）	单设备授权价格（加速版SDK）
1≤设备数≤10	300元	2000元
11≤设备数≤100	250元	1500元
101≤设备数	200元	1000元

**说明1：**加速版SDK目前已支持部分模型，点击查看基础版与加速版的性能对比：[图像分类](#)、[物体检测](#)、[图像分割](#)

**说明2：**设备授权的购买是「累计阶梯计费」，例如您购买了9个授权，因为这9个是在1~10的区间，所以付费300元/个×9个；当您继续购买2个时，累计购买量已经达到了11个，则付费1×300+1×250，因为后1个已经在11~100的区间内

### 按产品线购买永久授权

单个序列号可激活设备数上限为1万台，目前已支持按年购买授权：基础版SDK每年10万元，加速版SDK每年30万元。

如有授权年限、设备上限数的更多需求，请[提交工单](#)联系我们。

**说明1：**按产品线授权的序列号适用于开发手机APP，序列号仅限在绑定的包名下使用

**说明2：**如同一模型需同时购买iOS和Android包名的授权，需按2个产品线购买

## EasyDL图像价格常见问题

### 🔗 EasyDL图像常见问题

图像分类、物体检测、图像分割API如何收费？调用量不够怎么办？

- 每个图像API有累计10000点的免费调用额度，如需付费使用，请在[控制台](#)进行线上购买

## EasyDL图像公有云API价格说明

EasyDL图像支持发布为在线API的任务类型包括：图像分类、物体检测、图像分割

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发需求。具体介绍见下方。

### 按量后付费

只需在智能云控制台「EasyDL图像」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：物体检测-高性能模型对应发布的公有云服务，每次调用消耗5点，调用1000次对应消耗5000点，所以应付费5000\*0.001=5元

### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

**说明：**成功调用与失败调用均消耗免费额度。

### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	图像分类-高性能API服务保证10次并发；其余API保证4次并发

### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。每个月第一天清零上月累积的调用量，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每"点"对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

不同模型单次调用消耗的点数说明：

**注意：**公有云服务价格由实际服务压测结果计算得出，同一种模型的不同训练配置可能导致最终价格不同。下表单价仅供参考，请以实际公有云服务发布价格为准

模型名称	单次调用消耗点数 (点/次)	对应实际价格 (元/次)
图像分类-AutoDL Transfer	6	0.006
图像分类-高性能	4	0.004
图像分类-高精度	4	0.004
物体检测-高性能	5	0.005
物体检测-高精度	16	0.016
物体检测-超高精度	17	0.017
图像分割-实例分割-高精度	14	0.014
图像分割-实例分割-高性能	60	0.060
图像分割-语义分割-高精度	32	0.032
图像分割-语义分割-高性能	8	0.008
图像分类-精度提升配置包-0-50ms	4	0.004
图像分类-精度提升配置包-50-100ms	4	0.004
图像分类-精度提升配置包-100-200ms	4	0.004
图像分类-精度提升配置包-200-300ms	4	0.005
图像分类-精度提升配置包-300-500ms	6	0.006
图像分类-精度提升配置包-500-1000ms	6	0.006
图像分类-精度提升配置包-1000+ms	10	0.01
物体检测-精度提升配置包-0-200ms	4	0.004
物体检测-精度提升配置包-200-300ms	5	0.005
物体检测-精度提升配置包-300-500ms	8	0.008
物体检测-精度提升配置包-500-1000ms	8	0.008
物体检测-精度提升配置包-1000-1500ms	12	0.012
物体检测-精度提升配置包-1500-2000ms	17	0.017
物体检测-精度提升配置包-2000-3000ms	26	0.026
物体检测-精度提升配置包-5000+ms	60	0.06

**说明：**调用失败不计费

#### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元

中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元

本月费用共计：1800元

#### 余额不足提醒与欠费处理

##### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

##### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

#### 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)



调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗 $200000 \times 0.001 = 200$ 元，但「20万点」调用点包的目录价为180元。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数 (万点)	价格 (元)	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

#### ☞ QPS叠加包

已开通付费（或购买调用点包）后，若您有临时性的QPS高并发要求，可选择在某个时间段内叠加购买QPS。[立即购买](#)

QPS叠加包分为两种：

QPS叠加包种类	价格	说明
QPS叠加包	因公有云服务模型算法不同而不同	调用消耗仍将计算点数，对应计算费用
QPS叠加包（不限调用量）	对比同类算法的普通QPS叠加包，价格会较高一些	调用消耗不再计算点数，即无论调用量多少，公有云服务扣费不变

具体计费标准可在公有云服务开通「按量后付费」后，在对应的QPS叠加包购买页面查看，不同公有云服务的QPS叠加包价格不一样。

## EasyDL图像价格整体说明

本文档介绍EasyDL图像各项服务的价格

EasyDL旨在为开发者提供一站式AI开发体验，仅针对训练算力及部署两项内容收费。

### 算力收费

EasyDL图像提供付费算力，付费算力可用于模型训练以及批量预测功能，可根据实际需求购买算力使用时长。

价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 部署收费

EasyDL图像在模型部署方面支持公有云API部署、私有服务器部署、设备端离线SDK部署、软硬一体方案四种方式，可根据业务场景需求选择具体部署方式，不同部署方式的计费说明如下：

- 公有云API计费
  - 目前图像分类、物体检测、图像分割任务支持公有云API部署，API可在平台自助付费使用
- 私有服务器部署计费
  - 目前服务器部署包支持线上按设备数量和模型数量及有效期购买使用授权
- 设备端离线SDK计费
  - 目前设备端SDK已支持在线购买按设备使用授权、按产品线使用授权
- 软硬一体方案计费

目前已推出多种软硬一体方案，可在AI市场咨询购买

具体计费详情，请参考不同部署方式的计费文档。

### EasyDL图像算力资源价格说明

#### 算力资源价格说明

EasyDL提供了丰富的模型训练算法，同时在训练任务和批量预测任务上也提供多种机型自由选择。

#### 说明：

为更好支持您的模型付费训练，平台针对您的模型创建数量、任务并行数均进行了权益升级，每位用户创建模型数量从30个提升至100个；单模型组仅支持运行 1个训练任务提升至同时运行5个训练任务。

#### 计费方式

计费规则如下：

- 按分钟计费，不足1分钟按百分比计算。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后 1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 使用 EasyDL 前需保证账户无欠款。计费公式 费用=计算设备单价×计算设备数×使用时长 时长计量方法：只包括模型训练时的统计时间，数据预处理等不包括在计费时长内。

#### 产品单价

模型训练 在EasyDL图像方向的任务配置过程中，您可以选择训练的设备以及设备数量。目前图像分类的高性能和AutoDL算法只支持单设备训练。

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

说明1：扣费发生的时间点为任务训练结束（包含手动暂停训练或自动停止训练）后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

说明2：为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

批量预测 | 技术方向 | 计算设备 | 定价 | |-----| |-----| |-----| |-----| | EasyDL图像 | CPU\_4核\_16G | 0.025元/分钟/设备 (2小时免费额度) | EasyDL图像 | TeslaGPU\_P4\_8G显存单卡\_12核CPU\_40G内存 | 0.28元/分钟/设备 | EasyDL图像 | TeslaGPU\_P40\_24G显存单卡\_12核CPU\_40G内存 | 0.36元/分钟/设备 | EasyDL图像 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 0.45元/分钟/设备 |

说明1：扣费发生的时间点为任务结束后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

说明2：为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

算力资源包 | 可用模块 (图像) | 计算设备 | 时长 | 定价 | |-----| |-----| |-----| |-----| | 批量预测 | CPU\_4核\_16G | 50小时 | 70元/个 | / 批量预测 | CPU\_4核\_16G | 100小时 | 135元/个 | / 批量预测 | CPU\_4核\_16G | 300小时 | 380元/个 | / 模型训练、批量预测 | TeslaGPU\_P40\_24G显存单卡\_12核CPU\_40G内存 | 50小时 | 1000元/个 | / 模型训练、批量预测 | TeslaGPU\_P40\_24G显存单卡\_12核CPU\_40G内存 | 100小时 | 1900元/个 | / 模型训练、批量预测 | TeslaGPU\_P40\_24G显存单卡\_12核CPU\_40G内存 | 300小时 | 5300元/个 | / 模型训练、批量预测 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 50小时 | 1300元/个 | / 模型训练、批量预测 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 100小时 | 2400元/个 | / 模型训练、批量预测 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 300小时 | 6700元/个 | [点击此处](https://console.bce.baidu.com/ai/?=1612251100620&fromai=1#/ai/easydlLitelImage/buyHourPackage/index)立即前往购买

#### 余额不足提醒与欠费处理

### 余额不足提醒

根据您历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用免费额度，并且无法发起新的训练任务。

## EasyDL文本价格说明

### 文本私有服务器部署价格说明

EasyDL文本支持本地服务器部署的任务类型包括：文本分类-单标签、文本分类-多标签、文本实体抽取、文本实体关系抽取、情感倾向分析以及评论观点抽取，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用

如免费试用期结束后希望购买可在[控制台](#)在线按设备使用年限购买授权。

### 🔗 价格参考

部署类型	按年授权价格（每个模型每年在每个设备）	永久授权价格（每个模型在每个设备）
本地服务器SDK	10000元	50000元
本地服务器API	10000元	50000元

**说明：**本地服务器部署的授权粒度为每个模型每年每台设备，例如购买了2个模型3年在1台设备的授权，价格为2（个模型）x3（年）x1（台）x10000元=60000元。可在一台设备上激活2个有效期三年的模型。

### EasyDL文本公有云API价格说明

EasyDL文本支持发布为在线API的任务类型包括：文本分类-单标签、文本分类-多标签、文本实体抽取、文本实体关系抽取、情感倾向分析、短文本相似度、评论观点抽取、ERNIE大模型创作

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发需求。具体介绍见下方。

#### 按量后付费

只需在智能云控制台「EasyDL文本」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：文本实体抽取-高精度模型对应发布的公有云服务，每次调用消耗4点，调用1000次对应消耗4000点，所以应付费4000\*0.001=4元

#### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

**说明：**成功调用与失败调用均消耗免费额度。

#### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	文本分类API服务保证10次并发；其余文本API不保证调用并发

#### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每“点”对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

不同模型单次调用消耗的点数说明：

模型名称	单次调用消耗点数（点/次）	对应实际价格（元/次）
文本分类-单标签（短文本）-高精度	8	0.008
文本分类-单标签（短文本）-高性能	4	0.004
文本分类-单标签（多语种）-高精度	8	0.008
文本分类-多标签-高精度	8	0.008
文本分类-多标签-高性能	4	0.004
文本实体抽取-高精度	4	0.004
文本实体抽取-高性能	4	0.004
文本实体关系抽取-高精度	4	0.004
情感倾向分析-高精度	8	0.008
情感倾向分析-高性能	4	0.004
短文本相似度-高精度	8	0.008
短文本相似度-高性能	4	0.004
评论观点抽取-高精度	9	0.009
评论观点抽取-高性能	4	0.004
文本创作	2500	2.5

说明1：调用失败不计费

说明2：文本创作的公有云服务在邀测期间单独提供200万点的免费额度

#### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元

中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元

本月费用共计：1800元

#### 余额不足提醒与欠费处理

##### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

##### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

#### 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)

调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗200000\*0.001=200元，但「20万点」调用点

包的目录价为180元，相当于给客户打了9折。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数 (万点)	价格 (元)	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

## QPS叠加包

已开通付费（或购买调用点包）后，若您有临时性的QPS高并发要求，可选择在某个时间段内叠加购买QPS。[立即购买](#)

QPS叠加包分为两种：

QPS叠加包种类	价格	说明
QPS叠加包	因公有云服务模型算法不同而不同	调用消耗仍将计算点数，对应计算费用
QPS叠加包（不限调用量）	对比同类算法的普通QPS叠加包，价格会较高一些	调用消耗不在计算点数，即无论调用量多少，公有云服务扣费不变

具体计费标准可在公有云服务开通「按量后付费」后，在对应的QPS叠加包购买页面查看，不同公有云服务的QPS叠加包价格不一样。

## EasyDL文本价格整体说明

本文档介绍EasyDL文本各项服务的价格

EasyDL旨在为开发者提供一站式AI开发体验，仅针对训练算力及部署两项内容收费。

### 算力收费

EasyDL文本提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。（其中文本创作操作台作为新推出任务方向，会持续为开发者提供免费训练体验）

其余操作台算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 部署收费

EasyDL文本提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长

EasyDL文本在模型部署方面支持公有云API部署、私有服务器部署两种方式，可根据业务场景需求选择具体部署方式，不同部署方式的计费说明如下：

#### • 公有云API部署计费

文本分类-单标签、文本实体抽取、文本实体关系抽取、情感倾向分析、短文相似度、评论观点抽取任务支持公有云API部署，可通过公有云API可在平台自助付费使用

#### • 私有服务器部署计费

目前服务器部署包支持线上按设备数量和模型数量及有效期购买使用授权

具体计费详情，请参考不同部署方式的计费文档。

## EasyDL文本算力价格说明

### 算力资源价格说明

EasyDL提供了丰富的模型训练算法，同时在训练任务上也提供多种机型自由选择。

#### 说明：

为更好支持您的模型付费训练，平台针对您的模型创建数量、任务并行数均进行了权益升级，每位用户创建模型数量从30个提升至100个；单模型组仅支持运行 1个训练任务提升至同时运行5个训练任务。

### 计费方式

计费规则如下：

- 按分钟计费，不足1分钟按百分比计算。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后 1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 使用 EasyDL 前需保证账户无欠款。 计费公式 费用=计算设备单价×计算设备数×使用时长 时长计量方法：只包括模型训练时的统计时间，数据预处理等不包括在计费时长内。

### 产品单价

在EasyDL文本方向的任务配置过程中，您可以选择训练的设备以及设备数量。

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

**说明1：**扣费发生的时间点为任务训练结束（包含手动暂停训练或自动停止训练）后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

**说明2：**为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

算力资源包 | 可用模块（文本） | 计算设备 | 时长 | 定价 | | ----- | ----- | ----- | ----- | | 模型训练 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 50小时 | 1300元/个 | | 模型训练 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 100小时 | 2400元/个 | | 模型训练 | TeslaGPU\_V100\_16G显存单卡\_12核CPU\_56G内存 | 300小时 | 6700元/个 |

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用免费额度，并且无法发起新的训练任务。

## EasyDL结构化数据价格说明

### EasyDL结构化数据公有云API价格说明

EasyDL结构化数据支持发布为在线API的任务类型包括：表格数据预测、时序预测

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发

需求。具体介绍见下方。

### 按量后付费

只需在智能云控制台「EasyDL结构化数据」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：结构化数据-时序预测对应发布的公有云服务，每次调用消耗5点，调用1000次对应消耗5000点，所以应付费5000\*0.001=5元

### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

注：成功调用与失败调用均消耗免费额度。

### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	时序预测API服务保证4次并发；其余API不保证并发

### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每"点"对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

不同模型单次调用消耗的点数说明

模型名称	单次调用消耗点数（点/次）	对应实际价格（元/次）
结构化数据-表格数据预测	4	0.004
结构化数据-时序预测	5	0.005

说明：调用失败不计费

### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元

中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元

本月费用共计：1800元

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。



- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

## 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)

调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗 $200000 \times 0.001 = 200$ 元，但「20万点」调用点包的目录价为180元，相当于给客户打了9折。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数 (万点)	价格 (元)	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

## QPS叠加包

已开通付费（或购买调用点包）后，若您有临时性的QPS高并发要求，可选择在某个时间段内叠加购买QPS。[立即购买](#)

QPS叠加包分为两种：

QPS叠加包种类	价格	说明
QPS叠加包	因公有云服务模型算法不同而不同	调用消耗仍将计算点数，对应计算费用
QPS叠加包（不限调用量）	对比同类算法的普通QPS叠加包，价格会较高一些	调用消耗不在计算点数，即无论调用量多少，公有云服务扣费不变

具体计费标准可在公有云服务开通「按量后付费」后，在对应的QPS叠加包购买页面查看，不同公有云服务的QPS叠加包价格不一样

## 表格预测算力资源价格说明

### EasyDL结构化数据价格整体说明

本文档介绍EasyDL结构化数据服务的价格

EasyDL旨在为开发者提供一站式AI开发体验，仅针对训练算力及部署两项内容收费。

### 算力收费

EasyDL结构化数据提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 部署收费

EasyDL结构化数据在模型部署上支持公有云API部署

模型效果满意后，可将模型部署在公有云服务器上，公有云部署免费调用次数为10000点/接口，调用点数可在平台自主付费购买使用

EasyDL表格预测在模型部署上还支持服务器、通用小型设备离线部署

- 私有服务器部署计费



目前服务器部署包支持线上按设备数量和模型数量及有效期购买使用授权

- 设备端离线SDK计费

目前设备端SDK已支持在线购买按设备使用授权、按产品线使用授权

具体计费详情，请参考不同部署方式的计费文档

## EasyDL视频价格说明

### EasyDL视频公有云API价格说明

EasyDL视频-视频分类支持发布为在线API

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发需求。具体介绍见下方。

#### 按量后付费

只需在智能云控制台「EasyDL视频」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：视频分类对应发布的公有云服务，每次调用消耗15点，调用1000次对应消耗15000点，所以应付费15000\*0.001=15元

#### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

**说明：**成功调用与失败调用均消耗免费额度。

#### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	视频分类API服务保证4次并发

#### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。每个月第一天上月累积的调用量清零，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每“点”对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

不同模型单次调用消耗的点数说明

模型名称	单次调用消耗点数（点/次）	对应实际价格（元/次）
视频分类	15	0.015

**说明：**调用失败不计费

#### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元  
 中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元  
 本月费用共计：1800元

#### 余额不足提醒与欠费处理

##### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

##### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

#### 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)

调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗200000\*0.001=200元，但「20万点」调用点包的目录价为180元，相当于给客户打了9折。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数（万点）	价格（元）	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

#### QPS叠加包

已开通付费（或购买调用点包）后，若您有临时性的QPS高并发要求，可选择在某个时间段内叠加购买QPS。[立即购买](#)

QPS叠加包分为两种：

QPS叠加包种类	价格	说明
QPS叠加包	因公有云服务模型算法不同而不同	调用消耗仍将计算点数，对应计算费用
QPS叠加包（不限调用量）	对比同类算法的普通QPS叠加包，价格会较高一些	调用消耗不在计算点数，即无论调用量多少，公有云服务扣费不变

具体计费标准可在公有云服务开通「按量后付费」后，在对应的QPS叠加包购买页面查看，不同公有云服务的QPS叠加包价格不一样。

#### EasyDL视频本地服务器部署价格说明

EasyDL视频-目标跟踪支持本地服务器部署，在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

如免费试用期结束后希望购买可在[控制台](#)在线按设备使用年限购买授权。

#### 价格参考

部署类型	按年授权价格（每个模型每年在每个设备）	永久授权价格（每个模型在每个设备）
本地服务器SDK	10000元	50000元

**说明：**本地服务器部署的授权粒度为每个模型每年每台设备，例如购买了2个模型3年在1台设备的授权，价格为2（个模型）x3（年）x1（台）x10000元=60000元。可在一台设备上激活2个有效期三年的模型。

## EasyDL视频设备端部署价格说明

EasyDL视频-目标跟踪支持将定制模型部署在设备端上，每个模型发布设备端SDK后需通过序列号激活使用，每发布一个模型即可申请2个测试序列号供3个月内免费试用。

目前已支持在[控制台](#)在线按设备购买授权或按产品线购买。

### 价格参考

#### 按设备购买永久授权

授权终端设备数量	单设备授权价格（基础版SDK）	单设备授权价格（加速版SDK）
1≤设备数≤10	300元	2000元
11≤设备数≤100	250元	1500元
101≤设备数	200元	1000元

**说明：**设备授权的购买是「累计阶梯计费」，例如您购买了9个授权，因为这9个是在1~10的区间，所以付费300元/个×9个；当您继续购买2个时，累计购买量已经达到了11个，则付费1×300+1×250，因为后1个已经在11~100的区间内

#### 按产品线购买永久授权

单个序列号可激活设备数上限为1万台，目前已支持按年购买授权：基础版SDK每年10万元，加速版SDK每年30万元。

如有授权年限、设备上限数的更多需求，请[提交工单](#)联系我们。

**说明：**按产品线授权的序列号适用于开发手机APP，序列号仅限在绑定的包名下使用

## EasyDL视频软硬一体方案价格说明

目前EasyDL视频提供基于目标跟踪的软硬一体方案，请前往[专题页面](#)对比不同方案的性能与价格，选择与业务场景最匹配的方案。

### 方案获取流程

- Step 1：在EasyDL训练专项适配硬件的目标跟踪模型，迭代模型至效果满足业务需求
- Step 2：发布模型时选择专项适配硬件的专用SDK，并在[AI市场](#)购买方案
- Step 3：获得硬件和用于激活专用SDK的专用序列号，参考文档集成后，即可实现离线AI预测

如有其他硬件方案需求，请[提交工单](#)咨询。

## EasyDL视频价格整体说明

本文档介绍EasyDL视频各项服务的价格

EasyDL旨在为开发者提供一站式AI开发体验，仅针对训练算力及部署两项内容收费。

### 算力收费

EasyDL视频提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 部署收费

EasyDL视频模型部署上支持公有云API部署、私有服务器部署、设备端离线SDK部署、软硬一体方案四种方式，可根据业务场景需求选择具体部署方式，不同部署方式的计费说明如下：

- 公有云API计费

EasyDL视频-视频分类任务支持公有云API部署，可在平台自助付费使用

- 私有服务器部署计费

EasyDL视频-目标跟踪任务支持服务器部署，目前服务器部署包支持线上按设备数量和模型数量及有效期购买使用授权

- 设备端离线SDK计费

EasyDL视频-目标跟踪任务支持设备端SDK部署，目前设备端SDK已支持在线购买按设备使用授权、按产品线使用授权

- 软硬一体方案计费

EasyDL视频-目标跟踪任务目前已推出多种软硬一体方案，可在[AI市场](#)咨询购买

具体计费详情，请参考不同部署方式的计费文档。

## EasyDL视频算力资源价格说明

### 算力资源价格说明

EasyDL提供了丰富的模型训练算法，同时在训练任务上也提供多种机型自由选择。

#### 说明：

为更好支持您的模型付费训练，平台针对您的模型创建数量、任务并行数均进行了权益升级，每位用户创建模型数量从30个提升至100个；单模型组仅支持运行 1个训练任务提升至同时运行5个训练任务。

### 计费方式

具体计费规则如下：

- 按分钟计费，不足1分钟按百分比计算。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后 1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 使用 EasyDL 前需保证账户无欠款。计费公式 费用=计算设备单价×计算设备数×使用时长 时长计量方法：只包括模型训练时的统计时间，数据预处理等不包括在计费时长内。

### 产品单价

**模型训练** 在EasyDL视频方向的任务配置过程中，您可以选择训练的设备以及设备数量。

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

**说明1：**扣费发生的时间点为任务训练结束（包含手动暂停训练或自动停止训练）后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

**说明2：**为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您的历史账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

## 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用免费额度，并且无法发起新的训练任务。

## EasyDL语音价格说明

### EasyDL语音公有云API价格说明

EasyDL语音-声音分类支持发布为在线API

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发需求。具体介绍见下方。

### 按量后付费

只需在智能云控制台「EasyDL语音」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：声音分类对应发布的公有云服务，每次调用消耗4点，调用1000次对应消耗4000点，所以应付费4000\*0.001=4元

### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

**说明：**成功调用与失败调用均消耗免费额度。

### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	声音分类API服务保证4次并发

### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每“点”对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

不同模型单次调用消耗的点数说明：

模型名称	单次调用消耗点数（点/次）	对应实际价格（元/次）
声音分类	4	0.004

**说明：**调用失败不计费

### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元

中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元

本月费用共计：1800元

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

### 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)

调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗 $200000 \times 0.001 = 200$ 元，但「20万点」调用点包的目录价为180元，相当于给客户打了9折。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数（万点）	价格（元）	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

### QPS叠加包

已开通付费（或购买调用点包）后，若您有临时性的QPS高并发要求，可选择在某个时间段内叠加购买QPS。[立即购买](#)

QPS叠加包分为两种：

QPS叠加包种类	价格	说明
QPS叠加包	因公有云服务模型算法不同而不同	调用消耗仍将计算点数，对应计算费用
QPS叠加包（不限调用量）	对比同类算法的普通QPS叠加包，价格会较高一些	调用消耗不在计算点数，即无论调用量多少，公有云服务扣费不变

具体计费标准可在公有云服务开通「按量后付费」后，在对应的QPS叠加包购买页面查看，不同公有云服务的QPS叠加包价格不一样。

### EasyDL语音本地服务器部署价格说明

EasyDL语音-声音分类支持本地服务器部署，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

目前已支持在[控制台](#)在线按设备使用年限购买授权。

### 价格参考

部署类型	按年授权价格（每个模型每年在每个设备）	永久授权价格（每个模型在每个设备）
本地服务器SDK	10000元	50000元
本地服务器API	10000元	50000元

**说明1：**本地服务器SDK与本地服务器API在适配硬件上的评测信息已上线，点击查看性能对比：[声音分类](#)

**说明2：**本地服务器部署的授权粒度为每个模型每年每台设备，例如购买了2个模型3年在1台设备的授权，价格为2（个模型）x3（年）x1（台）x10000元=60000元。可在一台设备上激活2个有效期三年的模型。

## EasyDL语音本地设备端部署价格说明

EasyDL语音-声音分类支持通用小型设备部署，每个模型发布设备端SDK后需通过序列号激活使用，每发布一个模型即可申请2个测试序列号供3个月内免费试用。

如免费试用期结束后希望购买可在[控制台](#)在线按设备购买授权或按产品线购买。

### 🔗 价格参考

#### 按设备购买永久授权

授权终端设备数量	单设备授权价格（基础版SDK）	单设备授权价格（加速版SDK）
1≤设备数≤10	300元	2000元
11≤设备数≤100	250元	1500元
101≤设备数	200元	1000元

**说明1：**加速版SDK目前已支持部分模型，点击查看基础版与加速版的性能对比：[声音分类](#)

**说明2：**设备授权的购买是「累计阶梯计费」，例如您购买了9个授权，因为这9个是在1~10的区间，所以付费300元/个×9个；当您继续购买2个时，累计购买量已经达到了11个，则付费1×300+1×250，因为后1个已经在11~100的区间内

#### 按产品线购买永久授权

单个序列号可激活设备数上限为1万台，目前已支持按年购买授权：基础版SDK每年10万元，加速版SDK每年30万元。

如有授权年限、设备上限数的更多需求，请[提交工单](#)联系我们。

**说明1：**按产品线授权的序列号适用于开发手机APP，序列号仅限在绑定的包名下使用

**说明2：**如同一模型需同时购买iOS和Android包名的授权，需按2个产品线购买

## EasyDL语音价格整体说明

本文档介绍EasyDL语音各项服务的价格

EasyDL旨在为开发者提供一站式AI开发体验，仅针对训练算力及部署两项内容收费。

### 算力收费

EasyDL图像提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。（其中，语音识别由于任务特殊性，将持续为开发者提供免费训练体验）

声音分类价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡¥4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡¥4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡¥21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡¥27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 部署收费

EasyDL语音在模型部署方面支持公有云API部署、私有服务器部署、设备端离线SDK部署三种方式，可根据业务场景需求选择具体部署方式，不同部署方式的计费说明如下：

- 公有云API计费

目前声音分类API可在平台自助付费使用，语音识别在语音技术下任一接口购买量包、或者开通后付费的用户，可免费使用EasyDL语音-语音识别训练模型



- 私有服务器部署计费

目前服务器部署包支持线上按设备数量和模型数量及有效期购买使用授权

- 设备端离线SDK计费

目前设备端SDK已支持在线购买按设备使用授权、按产品线使用授权

具体计费详情，请参考不同部署方式的计费文档

## EasyDL语音算力资源价格说明

### 算力资源价格说明

EasyDL提供了丰富的模型训练算法，同时在训练任务上也提供多种机型自由选择。（其中，语音识别由于任务特殊性，将持续为开发者提供免费训练体验）

#### 说明：

为更好支持您的模型付费训练，平台针对您的模型创建数量、任务并行数均进行了权益升级，每位用户创建模型数量从30个提升至100个；单模型组仅支持运行 1个训练任务提升至同时运行5个训练任务。

### 计费方式

计费规则如下：

- 按分钟计费，不足1分钟按百分比计算。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后 1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 使用 EasyDL 前需保证账户无欠款。计费公式 费用=计算设备单价×计算设备数×使用时长 时长计量方法：只包括模型训练时的统计时间，数据预处理等不包括在计费时长内。

### 产品单价

**模型训练** 在EasyDL声音方向的任务配置过程中，您可以选择训练的设备以及设备数量。

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

**说明1：**扣费发生的时间点为任务训练结束（包含手动暂停训练或自动停止训练）后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

**说明2：**为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用免费额度，并且无法发起新的训练任务。

## EasyDL零售行业版价格说明

### 价格整体说明



## 简介

本文档介绍EasyDL零售版各项服务的价格，EasyDL零售版提供两种服务，分别为定制模型服务、货架拼接服务。其中定制模型服务支持云服务API和本地服务器部署两种服务方式，货架拼接服务支持云服务API。

云服务的计费方式包括：**按调用量后付费**、**调用量次数包预付费**和**QPS叠加包预付费**三种，根据实际购买的项目进行付费。其中**按调用量后付费**方式，系统根据实际调用的次数，每小时对您的百度云账户进行扣费。如果需要本地服务器部署，请加入EasyDL零售版官方QQ群（群号：1009661589）联系群管咨询。

## 计费方式介绍

EasyDL零售版的相关服务和接口计费方式如下：

服务名称	服务方式	是否计费	计费方式	免费额度
定制模型服务	云服务API	计费	按调用量后付费和QPS叠加包预付费	每个模型发布的云服务API享有1000次免费调用量，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>
翻拍识别服务	云服务API	计费	按调用量后付费、调用量次数包和QPS叠加包	每账号享有累计免费1000次，超出部分按调用量计费，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>
货架拼接服务	云服务API	计费	按任务数后付费、任务次数包和并发任务叠加包	累计免费200次拼接任务，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>

免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用，计费方式请参考[公有云API价格说明文档](#)。

上述服务中，定制商品检测服务支持本地服务器部署，如需要这种方式，请加入EasyDL零售版官方QQ群（群号：1009661589）联系群管咨询。

## 公有云API价格说明

### 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。

API	模型ID	模型类型	模型名称	模型版本	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
111	30443	商品检测	1111	V1	● 免费使用	剩余免费1000次	不保证并发	开通	购买   配账详情
silenceesapi	27185	商品检测	silencesecond	V3	● 付费使用	500次/天免费 + 超出按量计费	4	终止付费	购买   配账详情
silencefirst0829	27135	商品检测	silencefirst	V3	● 免费使用	500次/天免费	不保证并发	开通	购买   配账详情

API	状态	调用量限制	QPS限制	开通按量后付费
饮品检测	● 免费使用	500次/天免费	不保证并发	免费试用

API	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
日化品检测	● 免费使用	剩余免费1000次	不保证并发	免费试用	

API	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
商品陈列翻拍识别	● 付费使用	剩余免费999次 + 超出按量计费	4	终止付费	购买   配账详情

### 定制商品检测服务

#### 价目表 - 按调用量后付费

定制商品检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

#### 1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

#### 2. 商品陈列层数识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

#### 3. 商品陈列场景识别（可选）

接口支持识别商品陈列的场景，场景类型支持：货架、端架和立式冰柜

#### 4. 商品排面占比统计（可选）

接口支持统计商品排面数/占比、未识别商品数、空位数及货架利用率

#### 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用三项服务，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列层数识别和商品陈列场景识别两项服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见[服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

月调用量（万次）	单次调用价格（元）	QPS限制	说明
0<月调用量<=15	0.009	4	服务器支持每秒处理4次查询
15<月调用量<=150	0.008	4	服务器支持每秒处理4次查询
150<月调用量	0.007	4	服务器支持每秒处理4次查询

- 商品陈列层数识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.04	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品排面占比统计（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.02	4	服务器支持每秒处理4次查询

注：调用失败不计费

#### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制商品检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制商品检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

#### 费用举例

从2019-3-1至2019-3-31，定制商品检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

- 商品基本信息识别的费用为43,650元，明细如下：

前15万次落入0~15w阶梯，单次调用0.009元/次，费用为1,350元；

中间15万~150万次落入15~150w阶梯，单次调用0.008元/次，费用为10,800元；

最后150万~600万次落入大于150w阶梯，单次调用0.007元/次，费用为31,500元；

共计43,650元

2. 商品陈列层数识别的费用为360,000元，明细如下：

月调用量为600万次，单次调用0.04元/次，费用为240,000元

3. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计319,650元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1050元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制商品检测API的所有服务功能均有效

## 定制地堆检测服务

### 价目表 - 按调用量后付费

定制地堆检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

2. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

### 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用商品陈列场景识别服务功能，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列场景识别服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

单次调用价格（元）	QPS限制	说明
0.016	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制地堆检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制地堆检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制地堆检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

1. 商品基本信息识别的费用为96,000元，明细如下：

月调用量为600万次，单次调用0.016元/次，费用为96,000元

2. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计132,000元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	60元/天
按月购买	1200元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制地堆检测API的所有服务功能均有效

## 翻拍识别服务

### 价目表 - 按调用量后付费

#### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 价目表 - 调用量次数包

如果对调用次数有预估，可以选择购买**单次调用价格更低**的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	490 元	4	1年
10万次	4,800 元	4	1年
100万次	45,000 元	4	1年
500万次	212,500 元	4	1年
1000万次	420,000 元	4	1年
2000万次	800,000 元	4	1年

**购买后不可退款**，次数包使用完后，开始按调用量每次0.05元收取费用

**特殊说明**，此计费方式仅限于单独调用翻拍模型接口，定制商品检测服务接口中的翻拍服务的计费不适用

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

### 🔗 货架拼接服务

货架拼接服务支持按任务数后付费、任务次数包预付费和并发任务叠加包预付费三种计费方式。

#### 价目表 - 按任务数后付费

#### 付费调用

每个账户享有累计200次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

任务数	价格（元）	并发任务数限制	说明
每次拼接任务	0.2	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务启动后失败和运行前终止不计费，任务成功和运行后终止会计费用

### 免费额度

每个账号享有一定量免费调用额度，如下表：

服务	免费任务额度	并发任务数限制	说明
货架拼接	累计200次	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务成功与失败调用均消耗免费额度

### 价目表 - 任务次数包

如果对拼接任务次数有预估，可以选择购买**单次任务价格更低**的次数包，价格如下：

规格	价格	并发任务数限制	有效期
1千次	200 元	1	1年
1万次	1,900 元	1	1年
10万次	18,000 元	1	1年
100万次	150,000 元	1	1年
500万次	600,000 元	1	1年

购买后不可退款，任务次数包使用完后，开始按调用量每个任务0.2元收取费用

### 价目表 - 并发任务叠加包

开通付费后，并发任务数限制为1，如果有更多的并发请求需要，可以根据业务需求按天或按月购买并发任务叠加包，价格如下：

购买方式	每并发任务价格
按天购买	2元/天
按月购买	40元/月

购买 并发任务叠加包需保证已开通按量后付费或购买任务次数包

购买的并发任务叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## ☞ 余额不足提醒与欠费处理

### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

## 本地部署价格说明

对于调用量稳定且巨大的企业，可以选择将定制化商品检测AI模型私有化部署在企业本地服务器上，如需要这种方式，请加入EasyDL零售版官方QQ群（群号：1009661589）联系群管咨询。

## 公有云API价格说明

## ☞ 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。

零售版概览

产品介绍: (展开查看服务功能介绍) [展开](#)

可用接口列表

已上线的定制接口

API	模型ID	模型类型	模型名称	模型版本	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
	111	商品检测	1111	V1	● 免费使用	剩余免费1000次	不保证并发	<a href="#">开通</a>	<a href="#">购买</a>   <a href="#">配账详情</a>
	silenceapi	商品检测	silencesecond	V3	● 付费使用	500次/天免费 + 超出按量计费	4	<a href="#">终止付费</a>	<a href="#">购买</a>   <a href="#">配账详情</a>
	silencefirst0820	商品检测	silencefirst	V3	● 免费使用	500次/天免费	不保证并发	<a href="#">开通</a>	<a href="#">购买</a>   <a href="#">配账详情</a>

饮品检测

API	状态	调用量限制	QPS限制	开通按量后付费
饮品检测	● 免费使用	500次/天免费	不保证并发	<a href="#">免费试用</a>

日化品检测

API	状态	调用量限制	QPS限制	开通按量后付费
日化品检测	● 免费使用	剩余免费1000次	不保证并发	<a href="#">免费试用</a>

商品陈列翻拍识别

API	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
商品陈列翻拍识别	● 付费使用	剩余免费999次 + 超出按量计费	4	<a href="#">终止付费</a>	<a href="#">购买</a>   <a href="#">配账详情</a>

定制商品检测服务

价目表 - 按调用量后付费

定制商品检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

1. 商品基本信息识别 (必选)

接口支持识别商品信息 (商品名称、品牌、规格)、编号和置信度

2. 商品陈列层数识别 (可选)

货架场景：货架、端架 (小方货架)；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

3. 商品陈列场景识别 (可选)

接口支持识别商品陈列的场景，场景类型支持：货架、端架和立式冰柜

4. 商品排面占比统计 (可选)

接口支持统计商品排面数/占比、未识别商品数、空位数及货架利用率

付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用三项服务，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列层数识别和商品陈列场景识别两项服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别 (必选)，按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

月调用量 (万次)	单次调用价格 (元)	QPS限制	说明
0<月调用量<=15	0.009	4	服务器支持每秒处理4次查询
15<月调用量<=150	0.008	4	服务器支持每秒处理4次查询
150<月调用量	0.007	4	服务器支持每秒处理4次查询

- 商品陈列层数识别 (可选)，单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.04	4	服务器支持每秒处理4次查询

- 商品陈列场景识别 (可选)，单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品排面占比统计 (可选) , 单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.02	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制商品检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制商品检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制商品检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

1. 商品基本信息识别的费用为43,650元，明细如下：

前15万次落入0~15w阶梯，单次调用0.009元/次，费用为1,350元；

中间15万~150万次落入15~150w阶梯，单次调用0.008元/次，费用为10,800元；

最后150万~600万次落入大于150w阶梯，单次调用0.007元/次，费用为31,500元；

共计43,650元

2. 商品陈列层数识别的费用为360,000元，明细如下：

月调用量为600万次，单次调用0.04元/次，费用为240,000元

3. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计319,650元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1050元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制商品检测API的所有服务功能均有效



## 价目表 - 按调用量后付费

定制地堆检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

### 1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

### 2. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

## 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用商品陈列场景识别服务功能，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列场景识别服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

单次调用价格（元）	QPS限制	说明
0.016	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

注：调用失败不计费

## 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制地堆检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制地堆检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

## 费用举例

从2019-3-1至2019-3-31，定制地堆检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

- 商品基本信息识别的费用为96,000元，明细如下：

月调用量为600万次，单次调用0.016元/次，费用为96,000元

- 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计132,000元。

## 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	60元/天
按月购买	1200元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制地堆检测API的所有服务功能均有效

## 🔗 翻拍识别服务

### 价目表 - 按调用量后付费

#### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

#### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 价目表 - 调用量次数包

如果对调用次数有预估，可以选择购买**单次调用价格更低**的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	490 元	4	1年
10万次	4,800 元	4	1年
100万次	45,000 元	4	1年
500万次	212,500 元	4	1年
1000万次	420,000 元	4	1年
2000万次	800,000 元	4	1年

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

**特殊说明**，此计费方式仅限于单独调用翻拍模型接口，定制商品检测服务接口中的翻拍服务的计费不适用

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## 🔗 货架拼接服务

货架拼接服务支持按任务数后付费、任务次数包预付费和并发任务叠加包预付费三种计费方式。

### 价目表 - 按任务数后付费

#### 付费调用

每个账户享有累计200次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

任务数	价格（元）	并发任务数限制	说明
每次拼接任务	0.2	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务启动后失败和运行前终止不计费，任务成功和运行后终止会计费用

#### 免费额度

每个账号享有一定量免费调用额度，如下表：

服务	免费任务额度	并发任务数限制	说明
货架拼接	累计200次	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务成功与失败调用均消耗免费额度

### 价目表 - 任务次数包

如果对拼接任务次数有预估，可以选择购买**单次任务价格更低**的次数包，价格如下：

规格	价格	并发任务数限制	有效期
1千次	200 元	1	1年
1万次	1,900 元	1	1年
10万次	18,000 元	1	1年
100万次	150,000 元	1	1年
500万次	600,000 元	1	1年

购买后不可退款，任务次数包使用完后，开始按调用量每个任务0.2元收取费用

### 价目表 - 并发任务叠加包

开通付费后，并发任务数限制为1，如果有更多的并发请求需要，可以根据业务需求按天或按月购买并发任务叠加包，价格如下：

购买方式	每并发任务价格
按天购买	2元/天
按月购买	40元/月

购买 并发任务叠加包需保证已开通按量后付费或购买任务次数包

购买的并发任务叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## 余额不足提醒与欠费处理

### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

## 价格整体说明

### 简介

本文档介绍EasyDL零售版各项服务的价格，EasyDL零售版提供两种服务，分别为定制模型服务、货架拼接服务。其中定制模型服务支持云服务API和本地服务器部署两种服务方式，货架拼接服务支持云服务API。

云服务的计费方式包括：**按调用量后付费**、**调用量次数包预付费**和**QPS叠加包预付费**三种，根据实际购买的项目进行付费。其中**按调用量后付费**方式，系统根据实际调用的次数，每小时对您的百度云账户进行扣费。如果需要本地服务器部署，请加入EasyDL零售版官方QQ群（群号：1009661589）联系群管咨询。

### 计费方式介绍

EasyDL零售版的相关服务和接口计费方式如下：

服务名称	服务方式	是否计费	计费方式	免费额度
定制模型服务	云服务API	计费	按调用量后付费和QPS叠加包预付费	每个模型发布的云服务API享有1000次免费调用量，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>
翻拍识别服务	云服务API	计费	按调用量后付费、调用量次数包和QPS叠加包	每账号享有累计免费1000次，超出部分按调用量计费，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>
货架拼接服务	云服务API	计费	按任务数后付费、任务次数包和并发任务叠加包	累计免费200次拼接任务，超出免费额度的每次调用量根据选择的模型服务功能收取不同的费用，详情请见 <a href="#">价格说明文档</a>

免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用，计费方式请参考[公有云API价格说明文档](#)。

上述服务中，定制商品检测服务支持本地服务器部署，如需要这种方式，请加入EasyDL零售版官方QQ群（群号：1009661589）联系群管咨询。

## EasyDL跨模态价格说明

### EasyDL跨模态公有云API价格说明

EasyDL跨模态支持发布为在线API的任务类型包括：图文匹配

模型训练并发布为API后，可以在[控制台](#)看到已发布上线的所有公有云服务。

可以根据实际需求，开通「按量后付费」后，购买「调用点包」，高调用量下的优惠方案。也可以购买「QPS叠加包」，满足业务场景的高并发需求。具体介绍见下方。

### 按量后付费

只需在智能云控制台「EasyDL」-「公有云服务」中找到需要付费使用的接口，点击开通付费，即可完成付费开通。[立即开通](#)

根据实际调用消耗的点数，系统每小时会对您的百度智能云账户进行扣费。1点=0.001元。

例如：图文匹配-高精度模型对应发布的公有云服务，每次调用消耗32点，调用100次对应消耗3200点，所以应付费 $3200 \times 0.001 = 3.2$ 元

### 免费额度

EasyDL定制化API服务都具有免费调用额度，开通付费后，免费调用额度仍保留，所有技术方向的API接口均有10000点免费额度

说明：成功调用与失败调用均消耗免费额度。

### 免费/付费对比

对于EasyDL各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	图像分类-高性能API服务保证10次并发；其余API保证4次并发

### 价目表

产品采用分段阶梯定价方式，调用单价按照自然月累积调用量所落阶梯区间而变化。每个月第一天清零上月累积的调用量，重新开始累积本月调用量。

未购买优惠商品的调用接口享受阶梯价格，按月计算：

月调用量（万点）	每“点”对应换算（元/点）
0<月消耗调用点量<=100	0.001
100<月消耗调用点量<=500	0.0008
500<月消耗调用点量	0.00064

模型单次调用消耗的点数说明：

模型名称	单次调用消耗点数（点/次）	对应实际价格（元/次）
图文匹配-高精度	32	0.032

说明：调用失败不计费

### 费用举例

从2021-8-1至2021-8-31，本月某个公有云服务API接口的月消耗调用点量为200万点（已除去免费额度），费用如下：

前100万点落入0~100w阶梯，单价0.001元/点，费用为1000元

中间100万-200万点落入100w以上阶梯，单价0.0008元/点，费用为800元

本月费用共计：1800元

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

### 调用点包

开通按量后付费后即可购买调用点包。[立即购买](#)

调用点包的本质可理解为“充值折扣”，像「20万点」调用点包，按点与金钱换算规则，需要消耗 $200000 \times 0.001 = 200$ 元，但「20万点」调用点包的目录价为180元。

调用点包有效期为12个月。计费调用优先抵扣调用点包额度，额度耗尽自动切换为按量后付费形式，具体计费标准如下：

调用点包点数 (万点)	价格 (元)	对等折扣
20	180.00	0.90
50	430.00	0.86
100	800.00	0.80
300	2250.00	0.75
500	3500.00	0.70
1000	6500.00	0.65

## EasyDL跨模态算力资源价格说明

### 算力资源价格说明

EasyDL提供了丰富的模型训练算法，同时在训练任务上也提供多种机型自由选择。（其中，语音识别由于任务特殊性，将持续为开发者提供免费训练体验）

#### 说明：

为更好支持您的模型付费训练，平台针对您的模型创建数量、任务并行数均进行了权益升级，每位用户创建模型数量从30个提升至100个；单模型组仅支持运行 1个训练任务提升至同时运行5个训练任务。

### 计费方式

计费规则如下：

- 按分钟计费，不足1分钟按百分比计算。
- 按小时扣费，即北京时间整点扣费并生成账单。出账单时间是当前计费周期结束后 1小时内。例如，10:00-11:00的账单会在12:00之前生成，具体以系统出账时间为准。
- 使用 EasyDL 前需保证账户无欠款。 计费公式 费用=计算设备单价×计算设备数×使用时长 时长计量方法：只包括模型训练时的统计时间，数据预处理等不包括在计费时长内。

### 产品单价

#### 模型训练

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

**说明1：**扣费发生的时间点为任务训练结束（包含手动暂停训练或自动停止训练）后，如果因EasyDL系统异常导致训练任务运行失败，则相应训练任务的全部耗时在账单中会做扣减，不会参与计费。

**说明2：**为确保训练任务的正常进行，建议您在开通付费后确保账户余额不低于100元。

### 余额不足提醒与欠费处理

#### 余额不足提醒

根据您历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

#### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。

- 欠费后您开通付费的产品将进入欠费状态，只能使用免费额度，并且无法发起新的训练任务。

# EasyDL 图像使用说明

## EasyDL图像介绍

### 🔗 任务简介

Hi，欢迎来到[百度EasyDL图像](#)

目前EasyDL图像共支持训练3种不同应用场景的模型：

- 图像分类

识别一张图中是否是某类物体/状态/场景。可以识别图片中主体单一的场景

- 物体检测

在一张图包含多个物体的情况下，定制识别出每个物体的位置、数量、名称。可以识别图片中有多个主体的场景

- 图像分割

对比物体检测，支持用多边形标注训练数据，模型可像素级识别目标。适合图中有多个主体、需识别其位置或轮廓的场景

### 🔗 产品优势

#### 🔗 可视化操作

无需机器学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

#### 🔗 操作步骤

##### Step 1 创建模型

确定模型名称，记录希望模型实现的功能

##### Step 2 上传并标注数据

分类功能的模型：只需按分类（如合格图片vs不合格图片）上传图片即可

检测功能的模型：上传数据后，需要在数据中标注出需要检测的具体目标

分割功能的模型：上传数据后，需要在数据中标注出需要识别物体的轮廓

##### Step 3 训练模型并校验效果

选择部署方式与算法，用上传的数据一键训练模型

模型训练完成后，可在线校验模型效果

##### Step 4 发布模型

根据训练时选择的部署方式，将模型以云端API、本地部署SDK、端云协同部署包等多种方式发布使用

更详细的操作指导，请参考[各类模型的技术文档](#)

#### 🔗 高精度效果

- EasyDL图像底层结合百度 AutoDL/AutoML技术，针对用户数据能够自动获得最优模型和最优超参组合，进而基于少量数据就能获得出色性能和模型效果
- EasyDL图像以百度独有超大规模预训练模型为基座，小量级数据进行训练也可获得高精度模型

#### 🔗 高精算法

- 采用PaddlePaddle深度学习框架结合Auto Model Search，保证模型效果领先
- 训练图像分类和物体检测模型时，均支持选择多种算法，满足不同场景对性能、效果的不同需求；还有专项精度提升配置包，包含自动超参搜索、小目标检测等精度优化功能，针对优化模型效果

#### 🔗 AutoDL

训练图像分类模型时，支持选择AutoDL Transfer

AutoDL Transfer模型是百度研发的AutoDL技术之一，结合了模型网络结构搜索、迁移学习技术、并针对用户数据进行自动优化。与通用算法相比，训练时间较长，但更适用于细分类场景。例如，通用算法可用于区分猫和狗，但如果要区分不同品种的猫，则AutoDL效果会更好

#### 免训练极速迭代

训练图像分类模型之后，支持开启免训练极速迭代模式。该模式基于深度度量学习技术（Deep Metric Learning），模式开启后，模型的迭代添加数据仅需等待几分钟即可获得效果不错的模型，无需训练。适用于数据量大，模型迭代频繁的用户需求场景。

#### 丰富的部署方案

- 训练完成后，可将模型部署在公有云服务器、私有服务器，封装成可离线运行的设备端SDK，或直接购买软硬一体方案，灵活适配各种使用场景及运行环境，也可直接发布为端云协同部署包，下发至边缘设备进行应用
- 本地部署的性能评测详细信息可见[模型算法推理性能大表](#)

部署方式	支持的硬件	支持的系统	技术文档
公有云API	可集成公有云API即可	不限制	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
私有服务器部署 [私有API]	x86-64 CPU	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	Nvidia GPU	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
私有服务器部署 [服务器端SDK]	x86-64 CPU	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	Nvidia GPU	Linux/Windows	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	HUAWEI Atlas 300	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a>
通用设备端SDK	ARM (AArch64, ARMv7I)	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	Hisilicon NNIE	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a>
	HUAWEI Atlas 200	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a>
	ARM	Android/iOS	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	Qualcomm Snapdragon GPU/DSP	Android	<a href="#">图像分类</a> <a href="#">物体检测</a>
	Hisilicon Kirin NPU	Android	<a href="#">图像分类</a> <a href="#">物体检测</a>
专项硬件适配SDK [软硬一体方案]	Apple A-Bionic	iOS	<a href="#">图像分类</a> <a href="#">物体检测</a>
	x86-64 CPU	Windows	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	Intel Movidius NCS (MyRIAD 2/MyRIAD X)	Linux/Windows	<a href="#">图像分类</a> <a href="#">物体检测</a>
	Baidu-EdgeBoard(FZ)	Linux	<a href="#">方案介绍及对比</a>
端云协同部署	Baidu-EdgeBoard(VMX)	Linux/Windows	
	Nvidia-Jetson(Nano/TX2/Xavier)	Linux	
	x86-64 CPU	Linux	<a href="#">图像分类</a> <a href="#">物体检测</a> <a href="#">图像分割</a>
	ARM (AArch64, ARMv7I)	Linux	

#### 公有云API

支持图像分类、物体检测、图像分割模型

训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合

具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

支持查找云端模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，不断优化模型效果

#### 私有服务器部署

支持图像分类、物体检测、图像分割模型



将训练完成的模型部署在私有CPU/GPU服务器上，支持私有API和服务器端SDK两种集成方式，可在内网/无网环境下使用模型，确保数据隐私

- 私有API：将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷
- 服务器端SDK：将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活

#### 🔗 设备端SDK

##### 支持图像分类、物体检测、图像分割模型

训练完成的模型被打包成适配智能硬件（不含服务器）的SDK，可进行设备端离线计算。满足推理阶段数据敏感性要求、更快的响应速度要求

支持iOS、Android、Linux、Windows四种操作系统，基础接口封装完善，满足灵活的应用侧二次开发

提供基础版、加速版（已支持通用x86、通用ARM芯片）两种版本，可根据业务场景需求选择。了解加速版性能：[图像分类 物体检测](#)

#### 🔗 软硬一体方案

##### 支持图像分类、物体检测模型，[了解更多](#)

提供与模型深度适配的高性能硬件方案，多种算力、价位可选

可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松

#### 🔗 智能数据服务

全方位支持训练数据的采集、标注、质检、增强，助力提升模型效果

#### 🔗 数据采集

在云服务调用数据管理中，可查找云端模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集

可实现训练数据的持续丰富和模型效果的持续优化

点击了解功能说明：[图像分类、物体检测](#)

#### 🔗 智能标注

针对物体检测模型，可通过[智能标注](#)降低标注成本

启动后，只需标注数据集30%左右的数据即可训练出同等效果的模型

在图像分割任务中还提供“自动识别轮廓标注”来自动标注目标轮廓，降低标注成本

#### 🔗 多人标注

训练物体检测模型前，可与其他用户共享数据集，实现[多人分工标注](#)数据后再集中训练模型

#### 🔗 采集/标注支持

联合第三方数据标注合作伙伴，提供全面且高质量的训练数据采集、标注服务

可在AI市场选择合适的[数据服务商](#)

## 图像分类

### 整体介绍

#### 🔗 简介

Hi，您好，欢迎使用百度EasyDL图像-图像分类，请您根据实际应用场景选择模型类型。

EasyDL图像支持定制图像分类、物体检测、图像分割三类模型，三类模型的功能区别如下：

- 图像分类：识别一张图中是否是某类物体/状态/场景，适用于图片内容单一、需要给整张图片分类的场景
- 物体检测：检测图中每个物体的位置、名称。适合图中有多个主体要识别、或要识别主体位置及数量的场景
- 图像分割：对比物体检测，模型可像素级识别目标。适合图中有多个主体、需识别其位置或轮廓的场景

## 应用场景

图像分类是AI视觉应用中最经典的能力，常见的应用场景如下：

- 图片内容检索：定制训练需要识别的各种物体，并结合业务信息展现更丰富识别结果
- 图片审核：定制图像审核规则，如训练直播场景中抽烟等违规现象
- 制造业分拣或质检：定制生产线上各种产品识别，进而实现自动分拣或者质检
- 医疗诊断：定制识别医疗图像，辅助医生肉眼诊断

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作。在数据已经准备好的情况下，最快15分钟即可获得定制模型。

下面将详细介绍每一步的操作方式和注意事项。如果文档没有解决您的问题，请在百度智能云控制台内[提交工单](#)反馈。

□

## 数据准备

### 创建数据集

在训练之前需要在数据中心【创建数据集】，添加并标注数据



### 设计分类

首先想好分类如何设计，每个分类为你希望识别出的一种结果，如要识别水果，则可以以“apple”、“pear”等分别作为一个分类；如果是审核的场景判断合规性，可以以“qualified”、“unqualified”设计为两类，或者“qualified”、“unqualified1”、“unqualified2”、“unqualified3”……设计为多类。

注意：目前单个模型的分类型上限为1000类

### 准备数据

基于设计好的分类准备图片：

- 每个分类需要准备20张以上
- 如果想要较好的效果，建议每个分类准备不少于100张图片
- 如果不同分类的图片具有相似性，需要增加更多图片，尽量提升图片数据的丰富度
- 一个模型的图片总量限制10万张（每个账户的图片数量上限为10万张）

#### 图片格式要求：

- 1、目前支持图片类型为png、jpg、bmp、jpeg，图片大小限制在14M以内
- 2、图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px

#### 图片内容要求：

- 1、训练图片和实际场景要识别的图片拍摄环境一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片；如果是需要识别白天光照下的物体，就不能使用夜晚拍摄的图片数据
- 2、每个分类的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

如果训练图片场景无法全部覆盖实际场景要识别的图片：

- 如果要识别的主体在图片中占比较大，模型本身的泛化能力可以保证模型的效果不受很大影响

- 如果识别的主体在图片中占比较小，且实际环境很复杂无法覆盖全部的场景，建议用物体检测的模型来解决问题（物体检测可以支持将要识别的主体从训练图片中框出的方式来标注，所以能适应更泛化的场景和环境）

如果需要寻求第三方数据采集团队协助数据采集，可以在百度智能云控制台内[提交工单](#)

### 上传数据集并在线标注

在完成了设计分类与准备数据后，可以通过以下方式导入数据：

- 导入未标注/分类的数据，在线进行数据标注
- 直接导入标注/分类好的数据

### 导入未标注数据

#### 本地数据

支持上传图片、压缩包，或通过[API导入](#)

#### 已有数据集

支持选择百度云BOS导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他图像类模型的数据集



### 在线标注

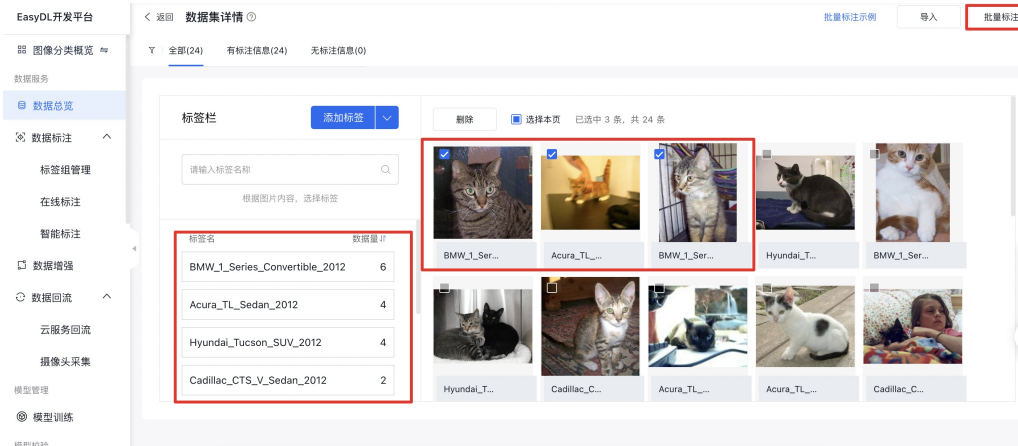
上传未标注数据后，即可进入「标注数据集」页面进行在线标注。标注的方式非常简单，只需在右侧标签栏新建并选定标签即可

标签名支持中英文数字中下划线，不超过256字符



批量标注 当图片数量较多

时，推荐您批量标注：选择好属于同一类的图片后，点击对应标签即可完成标注



### 导入已标注数据

### 本地数据

支持上传压缩包, 或通过API导入

压缩包支持通过两种格式上传, 点击「上传压缩包」, 即可查看详细的格式要求



### 已有数据集

支持选择线上已有的数据集, 仅支持选择图像分类数据集



使用智能标注功能可降低数据的标注成本。启动后，系统会从数据集所有图片中筛选出最关键的图片并提示需要优先标注。通常情况下，只需标注数据集30%左右的数据即可训练模型。与标注所有数据后训练相比，模型效果几乎等同

整体流程以图像分类的智能标注流程为例：

### 创建智能标注任务

启动图像分类数据集的智能标注前，请先检查以下是否已满足以下条件：

- 所有需要识别的分类标签都已创建
- 每个标签的图片数不少于10个
- 所有需要标注的图片都已加入数据集，且所有不相关的图片都已删除

若已满足，即可从导航栏进入「数据服务」-「智能标注」，创建智能标注任务，系统会基于您选择数据类型及数据量级，自动预估任务运行时长。

智能标注分为两种任务类型：**主动学习**、**指定模型**

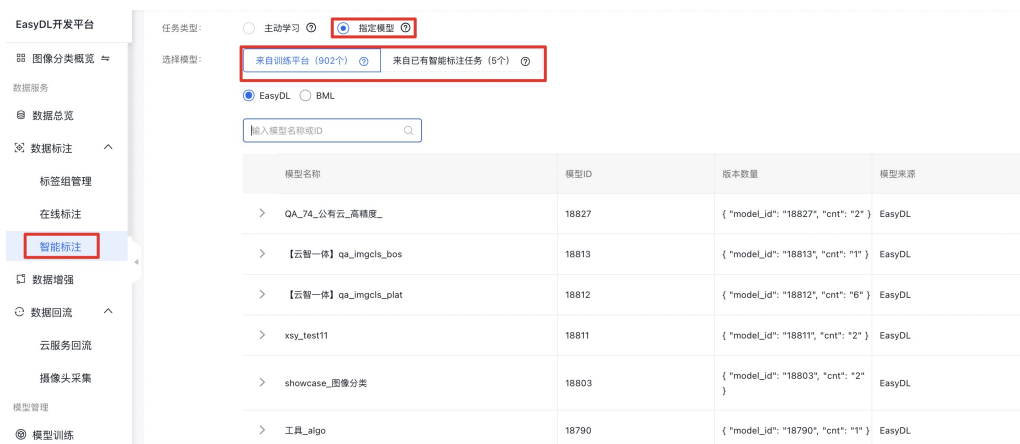
**主动学习**适合您在首次上传数据，没有使用数据训练过模型的情况下进行。智能标注算法会通过您已标注数据的规则，去标注数据

**指定模型**适合您已将需要智能标注的数据进行过模型训练，且模型精度表现良好（评估指标90%以上）时选择。智能标注算法将会以指定模型为基础去智能标注数据

**主动学习** 主动学习仅需选择需要智能标注的数据集即可发起智能标注任务



**指定模型** 当选择指定模型后，除了选择需要进行智能标注的数据集，还需选择智能标注算法的模型。支持从EasyDL/BML已训练的模型中选择，同时也支持从历史智能标注产生的模型中选择，建议选择所选数据集的数据训练产生的精度表现良好（评估指标90%以上）的模型



### 系统筛选难例

系统会分批筛选出最关键需标注的图片，即难例图片。

Tips：难例筛选需要一定时间，在此期间您可以正常进行其他未标注图片的标注



用户确认难例

智能标注任务启动后，系统为您自动筛选难例，您可以通过总览页查看进度按钮查看当前难例筛选进度，同时，进度图中也会全局展示您处于难例筛选的具体哪一环节，以便您的操作后续。筛选难例完成后，绿色进度条会进展到确认难例阶段，您可以点击【确认难例】完成对预标注结果的人工确认。



我们为您的人工确认提供两种模式：

- 单张确认，在该模式下支持您对预标注结果进行修正后点击保存
- 一键保存所有标注，为提升您的确认效率，默认您对难例的预标注结果全部满意，即可进入下一阶段



标注难例的预训练模型，也会对您无标注信息下的图片进行预标注结果的展示，您有余力的情况下，可以完成标注确认，确认后该张图片将升级为已标状态，该环节并非是您进入智能标注下一阶段的必备要求。





### 评估难例效果，完成任务

当您对难例完成确认后，您可以根据本轮次预标注的结果是否满意，判断您是否还需要进入下一轮难例筛选阶段，如果满意本轮难例的预标注效果，系统将自动为您系统其他的未标图片打标签。

#### 第1轮难例标注中（共4轮）

- 1、点击右下角【保存当前标注】该预标注结果将完成确认，支持您对标注修改后再保存
- 2、您只有对【待确认标注】下所有预标注结果完成确认，所有难例均升级为已标状态，才可进入下一阶段



### 中止任务

当您在任务运行中想要中止任务时，可实时点击标注页面右上方【中止任务】按钮，任务将被提前结束。



### 其他操作提示

- 在智能标注任务中，有任务上限吗？

支持五条智能标注任务同时运行，超过该上限您需要中止其他任务

- 智能标注中可以增删标签吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签 如果确实需要增删标签，可以先结束智能标注

- 智能标注中可以增删图片吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片。如果确实需要增删图片，可以先结束智能标注

- 智能标注中可以修改已标注图片的标注框吗？

可以。但为了保证智能标注的效果，建议不要大量改动。如果确实需要修改大量标注，建议先结束智能标注

- 为什么我已经人工标注了很多图片，但系统预标注依然不准？

系统预标注的结果会受以下因素影响：智能标注期间，对“已标注”图片的标签进行大量改动；曾结束智能标注，并对标签、图片进行增删

- 多个数据集是否可以同时启动智能标注？

目前每个账号同一时间仅支持对一个数据集启动智能标注

- 共享中的数据集是否可以启动智能标注？

暂不支持。智能标注中的数据集也暂不支持共享

- 智能标注失败了怎么办？

可以先尝试稍后重新启动，如多次失败请[提交工单](#)联系我们

## 问题反馈

您在使用EasyData过程中可以通过以下任何方式联系我们：

- 在社区咨询

在论坛发帖提交问题，也可以在论坛与其他用户一起交流。[前往论坛](#)

- 提交工单

如果使用EasyData遇到其他任何问题或任何bug，您可以点此[提交工单](#)

- 添加微信小助手留言

请在微信搜索“BaiduEasyDL”，并备注暗号“EasyData”，添加小助手后留言。

## 🔗 数据集管理API

本文档主要说明当您线下已有大量的已经完成分类的图片数据，如何通过调用API完成图片的便捷上传和管理。EasyDL图像数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

### 数据集创建API

#### 接口描述

该接口可用于创建数据集。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key





### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/create

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“Access Token获取”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

### 数据集列表API

#### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

#### 接口鉴权

同模型上线后获取的API：

- 1、在EasyDL控制台-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/list

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

### 分类（标签）列表API

### 接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

添加数据API

接口描述

该接口可用于在指定数据集添加数据。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

请求说明

请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
append Label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为 IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类10000个汉字</b>
entity_name	是	string	文件名
labels	否	array(object)	标签/分类数据。若为空，则只上传图片，不上传标签/分类。若不为空，则应在数组中包含以下前面带+的参数
+label_name	是	string	标签/分类名称（由中文、数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

#### 接口描述

该接口可用于删除数据集。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

分类（标签）删除API

接口描述

该接口可用于删除分类（标签）。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

请求说明

请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 🔗 数据质检

**功能概述** 该功能旨在对您数据集中的图像数据进行质量检测，通过提供客观指标，为您对数据集的下一步操作（标注、清洗等）进行参照引导。

整体质检报告将包括对原图、标注信息两个层面的指标进行统计，本期先上线原图维度的质检指标，标注层面的质检指标敬请期待。

### 使用流程 Step 1 功能入口

您可从数据总览页操作列点击【质检报告】或查看页面点击【质检报告】进入该功能页面

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	142909	5	● 已完成	图像分类	0% (0/5)	-	<a href="#">查看与标注</a> <a href="#">导出</a> <a href="#">删除</a> <a href="#">质检报告</a>

我的数据总览 > 【图片】的V1查看

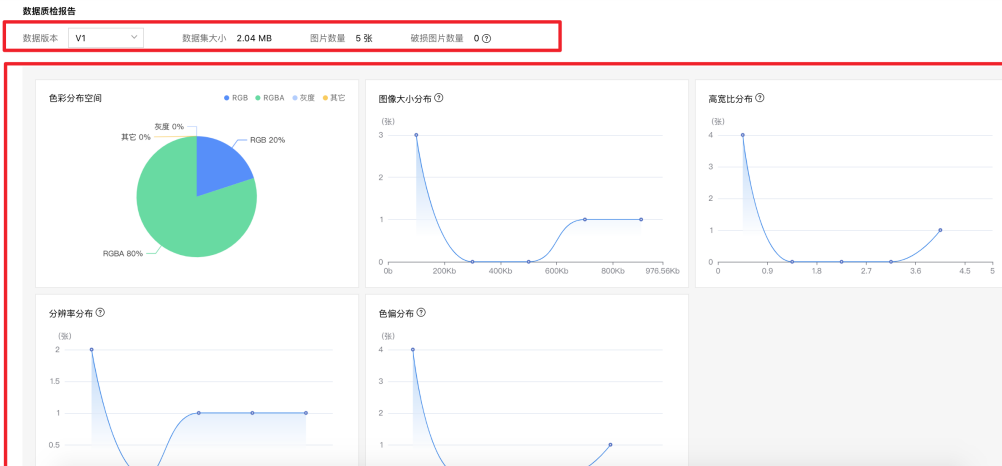
全部 (5) 有标注信息 (0) 无标注信息 (5) + 导入图片 质检报告 批量标注示例

的V1版本的图片列表 筛选 本页全选 删除



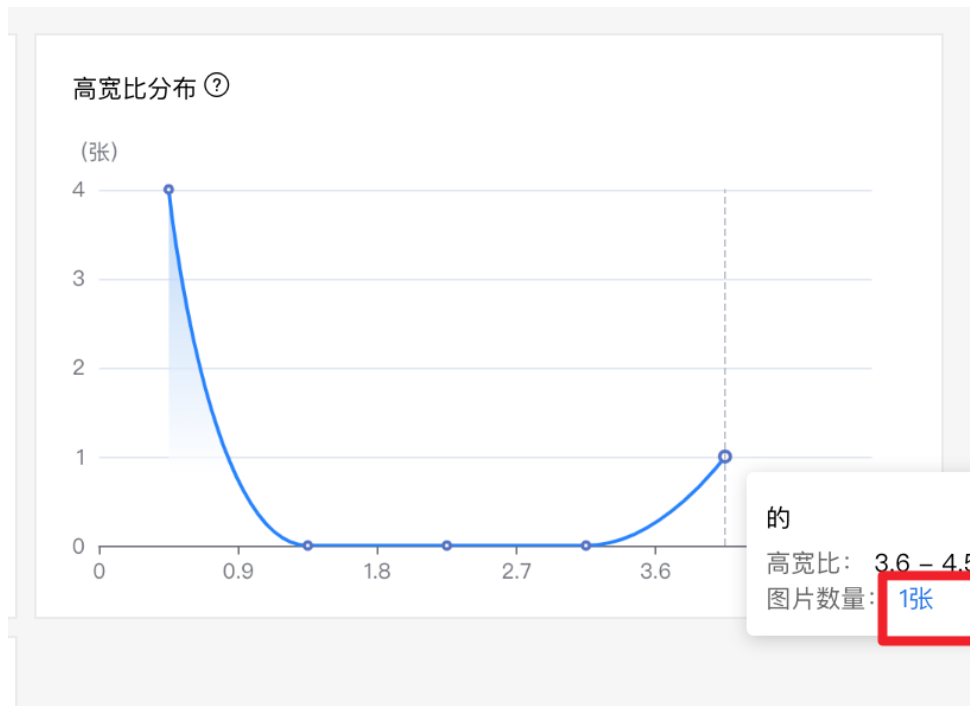
Step 2 指标查看 本期报告分为整体指标和分布指标两类。整体指标包括数据集存储大小、图片数量、破损图像数三类；分布指标包括色彩分布空间、图像存储大小分布、高宽比分布、分辨率分布、色偏分布五类。

可以通过切换数据集版本查看不同版本下质检报告。



Step 3 对应处理 可通过hover具体指标数值进行相关操作，以高宽比分布为例：

第一步，高宽比大于3.6的超长图hover显示有1张图片比，支持点击



第二步，点击后进入符合该指标的图片操作页，可针对筛选后图片进行删除、标注等操作



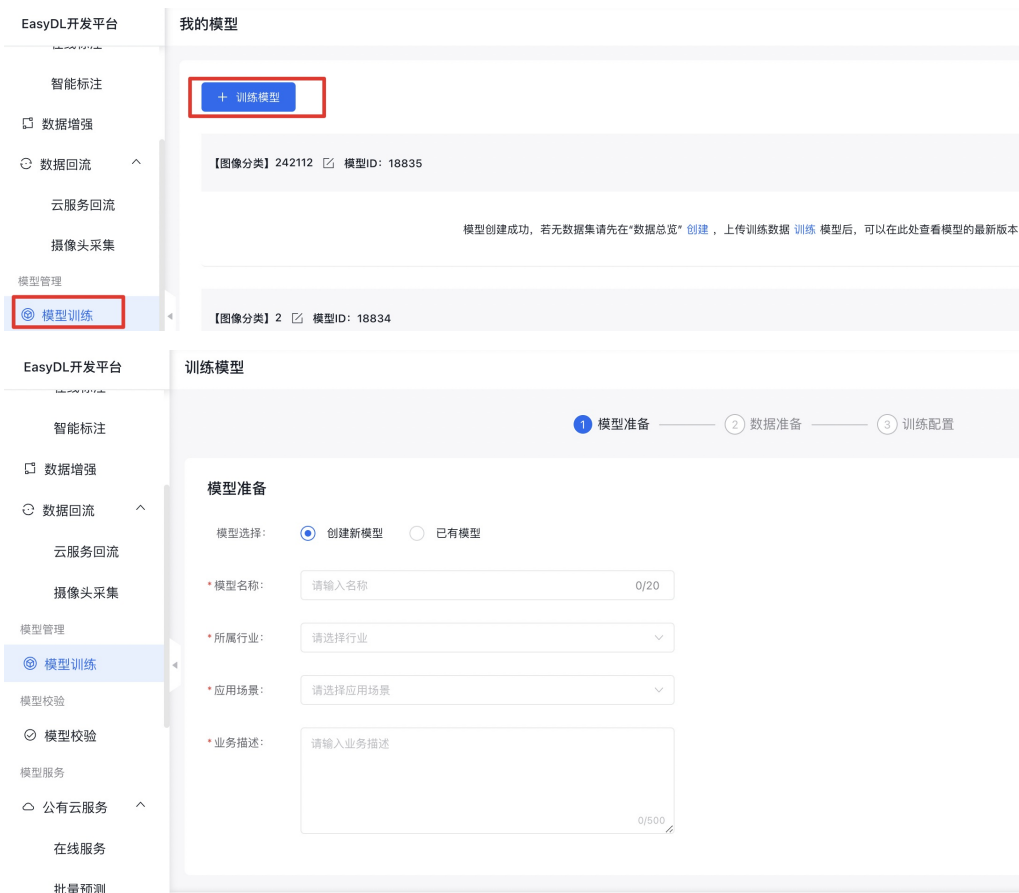


## 模型训练

### 🔗 图像分类创建模型

在导航【模型训练】中，点击训练模型，填写相关信息，即可创建训练模型。

操作示例：



- 注：1. 创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型  
 2. 目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练。  
 3. 如果您是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

### 🔗 图像分类训练操作说明

数据提交后，可以在导航中找到【训练模型】，按以下步骤操作，启动模型训练：

- 注：1. 启动训练前请确保数据已经标注完成，否则无法启动训练  
 2. 下述训练功能点中，标注为星号 (\*) 的功能为非必要选择项，可根据实际需求考虑是否使用



## ① 选择模型

选择此次训练的模型

## ② 添加数据

### 添加训练数据

- 先选择数据集，再按分类选择数据集里的图片，可从多个数据集选择图片
- 训练时间与数据量大小有关，以实际训练时长为准

Tips :

- 如只有1个分类需要识别，或者实际业务场景所要识别的图片内容不可控，可以在训练前勾选“增加识别结果为[其他]的默认分类”。勾选后，模型会将与训练集无关的图片识别为“其他”
- 如果同一个分类的数据分散在不同的数据集里，可以在训练时同时从这些数据集里选择分类，模型训练时会合并分类名称相同的图片

**添加自定义验证集\*** AI模型在训练时，每训练一批数据会进行模型效果检验，以某一张验证图片作为验证数据，通过验证结果反馈去调节训练。可以简单地把AI模型训练理解为学生学习，训练集则为每天的上课内容，验证集即为每周的课后作业，质量更高的每周课后作业能够更好的指导学生学习和找寻自己的不足，从而提高成绩。同理AI模型训练的验证集也是这个功效。

注：学生的课后作业应该与上课内容对应，这样才能巩固知识。因此，验证集的标签也应与训练集完全一致。

**添加自定义测试集\*** 如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果。

注：期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可。

### 配置数据增强策略

深度学习模型的成功很大程度上要归功于大量的标注数据集。通常来说，通过增加数据的数量和多样性往往能提升模型的效果。当在实践中无法收集到数目庞大的高质量数据时，可以通过配置数据增强策略，对数据本身进行一定程度的扰动从而产生“新”数据。模型会通过学习大量的“新”数据，提高泛化能力。

你可以在「默认配置」、「手动配置」、「自动数据增强」3种方式中进行选择，完成数据增强策略的配置。

#### 默认配置

如果你不需要特别配置数据增强策略，就可以选择默认配置。后台会根据你选择的算法，自动配置必要的数据增强策略。

#### 手动配置

EasyDL提供了大量的数据增强算子供开发者手动配置。你可以通过下方的算子功能说明或训练页面的效果展示，来了解不同算子的功能：

算子名	功能
ShearX	剪切图像的水平边
ShearY	剪切图像的垂直边
TranslateX	按指定距离（像素点个数）水平移动图像
TranslateY	按指定距离（像素点个数）垂直移动图像
Rotate	按指定角度旋转图像
AutoContrast	自动优化图像对比度
Contrast	调整图像对比度
Invert	将图像转换为反色图像
Equalize	将图像转换为灰色值均匀分布的图像
Solarize	为图像中指定阈值之上的所有像素值取反
Posterize	减少每个颜色通道的bits至指定位数
Color	调整图像颜色平衡
Brightness	调整图像亮度
Sharpness	调整图像清晰度
Cutout	通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例



**自动数据增强** 在训练方式选择「精度提升配置包」选项后，此处数据增强策略提供「自动数据增强」选项。自动数据增强算法会根据您数据的特性，自动选择数据增强算子。使用付费机型训练的用户请注意，自动数据增强算法可能会增加模型训练时间。

模型训练完成后，可在「我的模型-查看版本配置」中，查看配置记录：

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V7	训练完成	未申请	未发布	top1准确率100.00% top5准确率100.00% <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>

### 配置建议

算子的配置建议贴合实际场景。

比如，数字识别的数据集中，因为对数字的旋转很有可能导致错误样本的产生，所以不建议对数字数据集进行旋转操作。再比如，检测数据集中，如果标注量比较少，就可以通过随机平移的算子增强数据集，模型也更容易学习到目标物体的平移不变性。

### ③ 训练配置

#### 部署方式

- 可选择「公有云API」、「EasyEdge本地部署」
- 不知道如何选择？请参考[如何选择部署方式](#)

### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择设备
- 如果您选择了「公有云API」，则可按需选择训练方式

**增量训练\*** 增量训练：在模型迭代训练时，用户在原训练数据上增加了训练数据，可通过加载原训练数据训练的模型参数进行模型训练。这样可以让模型收敛速度变快，训练时间变短，同时在数据集质量较高的情况下，可能获得的模型效果也会更好。

注：1. 仅可选择同一部署方式下的训练的模型作为基准模型版本

2. 增量训练时所选择的标签应完全包含基准模型的标签。例如基准模型训练的标签为“A”、“B”，那么增量训练时的标签至少有“A”、“B”，不能只有“A”，不能为“C”，可以是“A”、“B”、“C”。

**训练方式** EasyDL目前提供完全免费的「常规训练」，以及限时免费的「精度提升配置包」两种选项。

- 「常规训练」包括EasyDL历史提供的「高精度」、「高性能」等模型选择，以及常规的模型训练配置
- 「精度提升配置包」选用百度自有超大规模预训练模型，让模型有更好的精度效果。并提供按云调用时延选择网络模型的形式，根据您的实际应用场景需求，选择更合适的模型。另外，EasyDL会持续在「精度提升配置包」中新增提升模型精度效果的配置策略，敬请期待。

**自动超参搜索\*** 自动超参搜索目前仅在精度提升配置包的选项下提供。选择开启自动超参搜索后，算法会多次实验，自动搜寻出适合模型训练的各种参数，来达到高精度的模型效果。

注：开启自动超参搜索后会增加3倍以上的训练时间，请根据实际需求考虑后选择

**高级训练配置\*** 高级训练配置开关默认关闭，建议对深度学习有一定了解的用户根据实际情况考虑使用。高级训练配置目前提供「输入图片分辨率」、「epoch」、「数据不平衡优化」三个配置项

- 输入图片分辨率：可以根据具体应用场景选择输入图片分辨率，如目标主体在图片中较小，就可适当增加输入图片分辨率，增强目标在数据层面的特性。推荐值为该类算法任务输入图片分辨率普遍最优值。
- epoch：训练集完整参与训练的次数。如有训练数据集较大，模型训练不充分，模型精度较低的情况，可适当设置较大epoch值（大于100），使模型训练更完整。
- 数据不平衡优化：适用于不同分类图片量差异较大的情况。当不同分类之间图片数量差异超过10倍以上时，建议开启。开启后可提升模型准确率及泛化能力

### 选择算法

不同的部署方式下，可以选择不同的算法。每个算法旁边有一个小问号，可以查看详细说明。

例如：选择「公有云API」后，可以在「高精度」、「高性能」、「AutoDL Transfer」3种算法中选择。鼠标移动到「AutoDL Transfer」右侧的问号上，可以看到对AutoDL算法的详细说明。

- 高精度模型在识别准确率上表现较好，但在识别速度上表现较弱。高性能模型反之。
- 如果你已从AI市场购买了模型算法，也可以基于已购模型的算法训练：[前往AI市场购买](#)>
- 本地部署的用户可在[算法推理性能大表](#)中查看具体硬件上评测的性能信息

## ③ 添加数据

### 添加训练数据

- 先选择数据集，再按分类选择数据集里的图片，可从多个数据集选择图片
- 训练时间与数据量大小有关，以实际训练时长为准

Tips：

- 如只有1个分类需要识别，或者实际业务场景所要识别的图片内容不可控，可以在训练前勾选"增加识别结果为[其他]的默认分类"。勾选后，模型会将与训练集无关的图片识别为"其他"
- 如果同一个分类的数据分散在不同的数据集里，可以在训练时同时从这些数据集里选择分类，模型训练时会合并分类名称相同的图片

**添加自定义验证集\*** AI模型在训练时，每训练一批数据会进行模型效果检验，以某一张验证图片作为验证数据，通过验证结果反馈去调节训练。可以简单地把AI模型训练理解为学生学习，训练集则为每天的上课内容，验证集即为每周的课后作业，质量更高的每周课后作业能够更好的指导学生学习和找寻自己的不足，从而提高成绩。同理AI模型训练的验证集也是这个功效。

注：学生的课后作业应该与上课内容对应，这样才能巩固知识。因此，验证集的标签也应与训练集完全一致。

**添加自定义测试集\*** 如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果。

注：期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可。

### 配置数据增强策略

深度学习模型的成功很大程度上要归功于大量的标注数据集。通常来说，通过增加数据的数量和多样性往往能提升模型的效果。当在实践中无法收集到数目庞大的高质量数据时，可以通过配置数据增强策略，对数据本身进行一定程度的扰动从而产生"新"数据。模型会通过学习大量的"新"数据，提高泛化能力。

你可以在「默认配置」、「手动配置」、「自动数据增强」3种方式中进行选择，完成数据增强策略的配置。

#### 默认配置

如果你不需要特别配置数据增强策略，就可以选择默认配置。后台会根据你选择的算法，自动配置必要的数据增强策略。

#### 手动配置

EasyDL提供了大量的数据增强算子供开发者手动配置。你可以通过下方的算子功能说明或训练页面的效果展示，来了解不同算子的功能：

算子名	功能
ShearX	剪切图像的水平边
ShearY	剪切图像的垂直边
TranslateX	按指定距离（像素点个数）水平移动图像
TranslateY	按指定距离（像素点个数）垂直移动图像
Rotate	按指定角度旋转图像
AutoContrast	自动优化图像对比度
Contrast	调整图像对比度
Invert	将图像转换为反色图像
Equalize	将图像转换为灰色值均匀分布的图像
Solarize	为图像中指定阈值之上的所有像素值取反
Posterize	减少每个颜色通道的bits至指定位数
Color	调整图像颜色平衡
Brightness	调整图像亮度
Sharpness	调整图像清晰度
Cutout	通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例



**自动数据增强** 在训练方式选择「精度提升配置包」选项后，此处数据增强策略提供「自动数据增强」选项。自动数据增强算法会根据您数据的特性，自动选择数据增强算子。使用付费机型训练的用户请注意，自动数据增强算法可能会增加模型训练时间。

模型训练完成后，可在「我的模型-查看版本配置」中，查看配置记录：

【图像分类】 zh_handle 模型ID: 10064						训练	历史版本	删除
部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作		
公有云API	V7	训练完成	未申请	未发布	top1准确率100.00% top5准确率100.00% 完整评估结果	<a href="#">查看版本配置</a>	<a href="#">申请发布</a>	<a href="#">校验</a>

## 配置建议

算子的配置建议贴合实际场景。

比如，数字识别的数据集中，因为对数字的旋转很有可能导致错误样本的产生，所以不建议对数字数据集进行旋转操作。再比如，检测数据集中，如果标注量比较少，就可以通过随机平移的算子增强数据集，模型也更容易学习到目标物体的平移不变性。

## ④ 训练模型

点击「开始训练」，训练模型。

- 模型训练过程中，可以设置训练完成的短信提醒并离开页面。
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。
- 训练时间与数据量大小有关，以实际训练时长为准。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

## 🔗 图像分类模型效果评估

可通过模型评估报告或模型校验了解模型效果：

- 模型评估报告：训练完成后，可以在列表中看到模型效果，以及详细的模型评估报告。
- 模型在线校验：可以在左侧导航中找到【模型校验】，在线校验模型效果。校验功能示意图：

## 模型评估报告

### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、准确率、F1-score等。这部分模型效果的指标是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如图片数量低于100个），参与评估的数据可能不超过30个，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

查看模型评估结果时，需要思考在当前业务场景，更关注精确率与召回率哪个指标。是更希望减少误识别，还是更希望减少漏识别。前者更需要关注精确率的指标，后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况，对于希望准确率与召回率兼具的场景，F1-score越接近1效果越好。评估指标具体的说明如下。

**F1-score**：对某类别而言为精确率和召回率的调和平均数，评估报告中指各类别F1-score的平均数

**准确率**：基于随机测试集进行计算，为正确分类的样本数与总样本数之比

注意：若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口批量测试，获取更准确的模型效果。

### 整体评估

test02 V2效果优异，建议针对识别错误的图片示例继续优化模型效果。 [如何优化效果?](#)



### top1-top5准确率

对于每一个评估的图片文件，模型会根据置信度高低，依次给出top1-top5的识别结果，其中top1置信度最高，top5的置信度最低。那么top1的准确率值是指对于评估标准为“top1结果识别为正确时，判定为正确”给出准确率。top2准确率值是指对于评估标准为“top1或者top2只要有一个命中正确的结果，即判定为正确”给出的准确率。……以此类推。

**模型调优建议** 在模型评估中，EasyDL将会通过智能算法对误识别的样本进行归因分析，可推断出误识别的样本对某个模型评估指标的具体影响以及影响程度，并提供对应优化的方案。同时还可针对某个具体表现不好的标签进行归因分析，针对性优化识别效果



**模型调优建议**

归因粒度 **基于整个模型** 基于单个标签

序号	受影响指标	影响程度	根因分析	调优对策
1	F1-Score	中	"高宽比"对"F1-Score"的效果有"一定"影响,不同特征区间的"F1-Score"方差达到"0.0352"	在【添加数据】->【数据增强策略】中配置"ShearX,ShearY"进行增强。
2	F1-Score	中	"分辨率"对"F1-Score"的效果有"一定"影响,不同特征区间的"F1-Score"方差达到"0.0321"	在【添加数据】->【数据增强策略】中配置"ShearX,ShearY"进行增强。
3	F1-Score	中	"色偏"对"F1-Score"的效果有"一定"影响,不同特征区间的"F1-Score"方差达到"0.0205"	在【添加数据】->【数据增强策略】中配置"Color,Posterize"进行增强。
4	F1-Score	中	"亮度"对"F1-Score"的效果有"一定"影响,不同特征区间的"F1-Score"方差达到"0.0188"	在【添加数据】->【数据增强策略】中配置"Brightness"进行增强。
5	F1-Score	中	"饱和度"对"F1-Score"的效果有"一定"影响,不同特征区间的"F1-Score"方差达到"0.018"	在【添加数据】->【数据增强策略】中配置"Color"进行增强。

**详细评估**

这个部分支持查看模型识别错误的图片示例，以及使用混淆矩阵定位易混淆的分类。

**识别错误图片示例**

通过分标签查看模型识别错误的图片，寻找其中的共性，进而有针对性的扩充训练数据。

**详细评估**

按分类查看错误示例

不同分类的F1-score及对应的识别错误的图片（不包含训练时可能勾选的“其他”类识别错误的图片）

例如，你训练了一个将小番茄和樱桃分类的模型。在查看小番茄分类的错误示例时，发现错误示例中有好几张图片都是带着绿色根茎的小番茄（与樱桃比较相似）。这种情况下，就需要在小番茄分类的训练集中，多增加一些带绿色根茎的图片，让模型有足够的学习数据能够学习到带根茎的小番茄和樱桃的区别。

这个例子中，我们找到的是识别错误的图片中，目标特征上的共性。除此之外，还可以观察识别错误的图片在以下维度是否有共性，比如：图片的拍摄设备、拍摄角度，图片的亮度、背景等等。



**定位易混淆分类**

支持按识别错误样本量的绝对数值/相对数值查看混淆矩阵，获得具体到数据级别的精度评价信息。同时支持下载完整的混淆矩阵进行更深入的分析。



定位易混淆分类

下方是imagenet2012\_6c\_V6模型的混淆矩阵，每一个橙色的方格都对应一组易混淆的分类（最多展示10个易混淆的分类），点击方格即可进一步分析模型在识别该组分类时，依据的关键特征。

展示了数据标注与模型预测不符数量前10的标签，点击标签可在下方查看示例图，帮助您有针对性地设计特征，使得类别更具区分性。

[下载完整混淆矩阵](#)

序号	标签名称	误识别标签TOP5及其数量	精确率	测试集数量	召回率	f1-score
1	n02777292	[2] n03942813 [1] n03179701	100.0%	7	57%	73%
2	n03942813	[1] n03188531 [2] n02777292	85.0%	17	100%	92%
3	n03188531	[1] n03942813	100.0%	18	94%	97%
4	n03179701	[1] n02777292	97.0%	32	100%	98%


分析热力图

点击混淆矩阵中带有数字的方格，可以进一步分析对应易混淆分类的示例图，非常直观地对影响模型精度的因素进行判断。

**易混淆分类示例图**  查看图片  查看热力图

请仔细分析以下三组图片，通过有针对性地补充训练数据等方式，提升模型准确度。 [参考文档](#)


3张标注为Adventure分类的图片被模型误识别为Action分类



被模型准确识别为Adventure分类的图片

暂无相关图片

被模型准确识别为Action分类的图片



示例图共分为3组，假设选定的易混淆分类是「A分类被误识别为B分类」：

- 1、实际标注为A分类，但被模型识别为B分类的所有图片
- 2、被模型准确识别为A分类的图片
- 3、被模型准确识别为B分类的图片

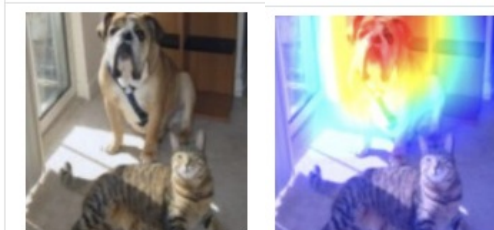
3组图片均支持查看原图与热力图。其中，热力图可以进一步地解释模型的决策依据，在整图范围内给出影响模型识别结果的像素重要程度，热力图的颜色如下图，颜色越靠右代表像素重要性越高。



以猫狗分类为例：

如果分类为狗，但被模型误识别为猫的示例图里出现的狗，与模型准确识别为狗的示例图的图片性状大不一致，那可以判断出数据集对某种性状的狗的识别能力不足，需要继续增加该性状的狗的数据。

如下猫狗的图片，模型给出预测结果为狗，通过热力图的查看，可以看到支持模型给出狗的预测结果的决策依据正是图中狗脸附近像素区域。



按分类挖掘人工易错标图片 在标注数据时可能由于粗心将不属于某类别的图片标注为了该类别，这种情况十分常见。在评估报告中的「按分类挖

掘人工易错标图片」中即可快速检查到这些图片，并同时完成标签修改。



### 🔗 图像分类模型如何提升效果

一个模型很难一次性就训练到最佳的效果，可能需要结合模型评估报告和校验结果不断扩充数据和调优。

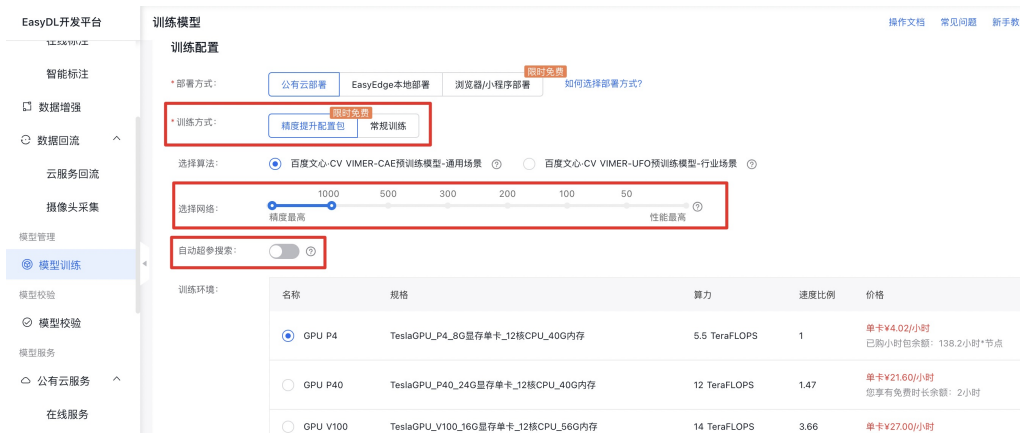
为此我们设计了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，获得更好的模型效果。

**注意：如果模型已经是上线状态（包括已付费的模型服务），依然支持模型迭代。只需要在训练完毕后发布新的版本，就可以获得更新后的模型服务。**

想要提升模型效果，可以尝试以下方法：

**尝试不同的训练配置** 可前往训练配置页面尝试不同的配置组合，因不同数据集在不同的算法上可能表现不一致，所以建议您多尝试不同的算法选型后综合挑选精度最高的模型使用，你可以选择如下的配置项：

- 精度提升配置包
- 平衡精度性能
- 自动超参搜索



### 🔗 免训练迭代模式

### 🔗 免训练迭代模式

**整体介绍** 免训练迭代模式是EasyDL针对于“需要高频迭代模型，但模型训练时间成本太高”的用户使用场景推出的新型模型迭代模式，在常规模型训练完成后开启免训练迭代模式，即可在「免训练模式数据底库」中通过增删数据来迭代模型的预测能力。值得一提的是，该模式下新增一类标签数据也可以短时间内马上获得新标签预测能力的模型

注：1. 免训练迭代模式仅适合短期内快速获得具备一定精度模型的应用场景，如需获得效果更好的模型，请提升数据量和数据丰富度后进行模型训练

2. 免训练迭代模式会根据模式开启时的模型和数据生成预测基础模型，基础模型影响后续的预测能力，所以在初次打开免训练迭代模式时，尽量保证当时的模型和数据质量

### 使用流程

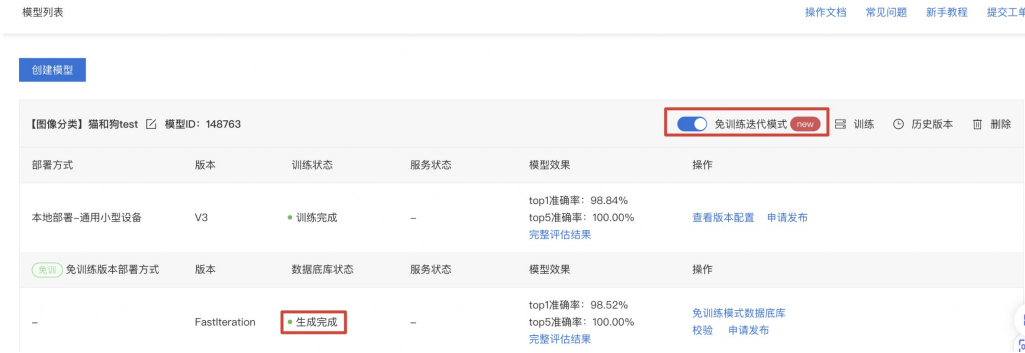
#### ①模型训练 请创建模型并训练训练模型，在模型训练完成后请勿删除原训练数据



②启动免训练迭代模式 模

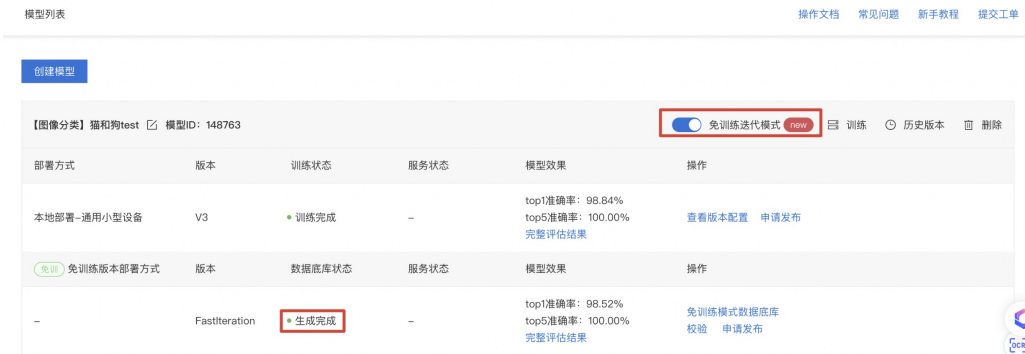
型训练完成之后，打开免训练迭代模式开关，生成「免训练模式数据底库」

注：此过程可能需要一段时间，具体时长与模型训练相差无几



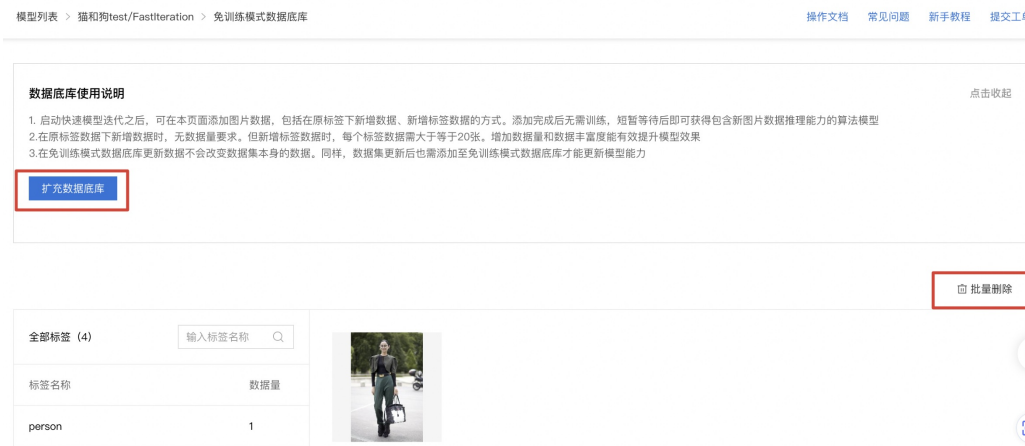
③免训练极速迭代 免训练

模式数据底库生成完成后，点击进入数据底库



在数据底库中，您可以点

击「扩充数据底库」来批量添加新标签数据，来新增模型的预测能力。也可以通过「批量删除」操作，来删除噪声数据，提升模型预测精度。



说明：在数据底库中的增删操作，不论数据量的大小都会引起模型更新，建议您确定需要调整的数据后一并操作，避免重复操作，重复等待

### 模型发布

#### 🔗 图像分类模型发布整体说明

训练完成后，可将模型部署在公有云服务器、通用小型设备、本地服务器，或直接购买软硬一体方案，灵活适配各种使用场景及运行环境

#### 公有云在线服务

训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合

具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

支持查找云端模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，不断优化模型效果

**纯离线服务** 训练完成的模型整体打包为纯离线服务，可下载在本地稳定调用。纯离线服务按部署硬件芯片不同分为本地服务器部署、通用小型设备部署。为了提供更好的算法与硬件推理效果，EasyDL提供软硬一体方案部署。纯离线服务的整体支持与评测信息可详见[算法与性能评测大表](#)

#### 本地服务器部署

可将训练完成的模型部署在私有CPU/GPU服务器上，支持服务器API和服务器SDK两种集成方式

模型服务性能表现更好，适用于对性能要求较高的场景，例如工业质检、流水线产品分拣等

#### 通用小型设备

训练完成的模型被打包成适配智能硬件的SDK，可进行设备端离线计算。满足推理阶段数据敏感性要求、更快的响应速度要求

支持iOS、Android、Linux、Windows四种操作系统，基础接口封装完善，满足灵活的应用侧二次开发

#### 软硬一体方案

高性能硬件与模型深度适配，多种方案可选。可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松。[了解更多](#)

## 端云协同服务

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新

断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）

联网状态下在平台管理设备运行状态、资源利用率

#### ☞ 公有云部署

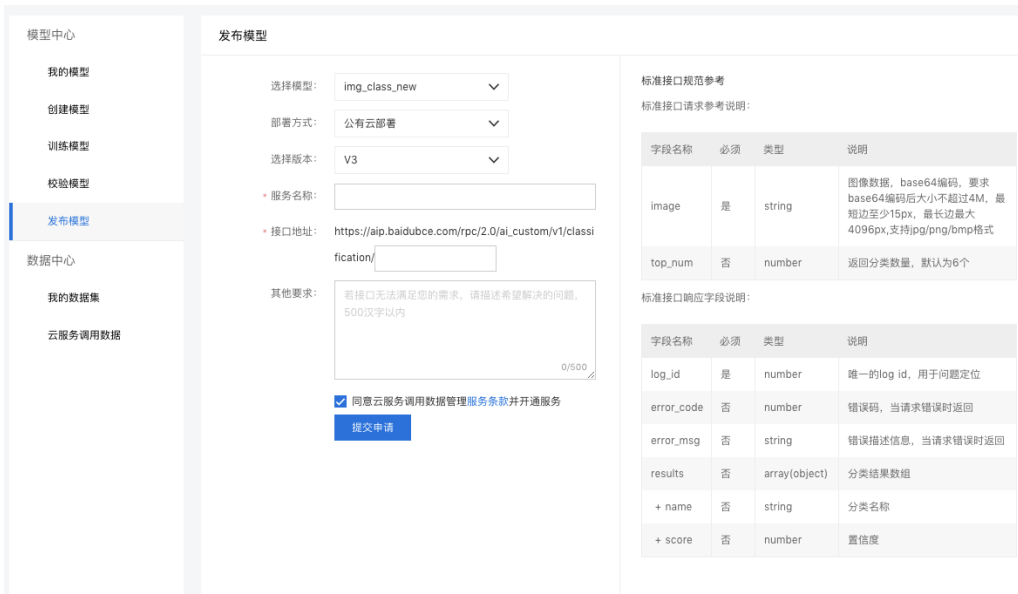
#### ☞ 如何发布图像分类API

训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可发布公有云API：

- 选择模型
- 选择部署方式「公有云部署」
- 选择版本
- 自定义服务名称、接口地址后缀
- 申请发布

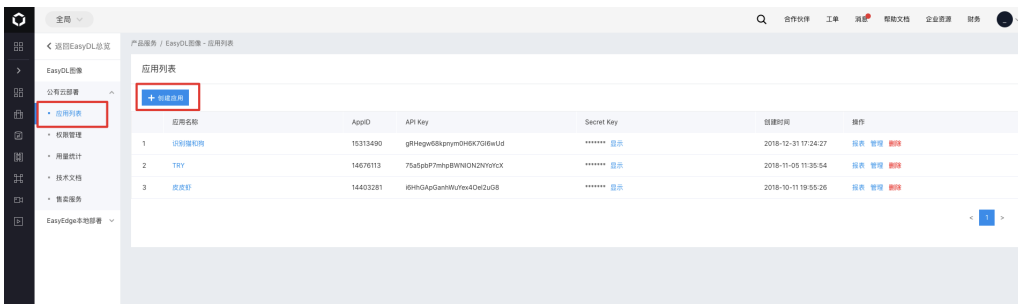
申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。如果需要加急、或者遇到莫名被拒的情况，请在百度智能云控制台内[提交工单反馈](#)。

发布模型界面示意：



### 接口赋权

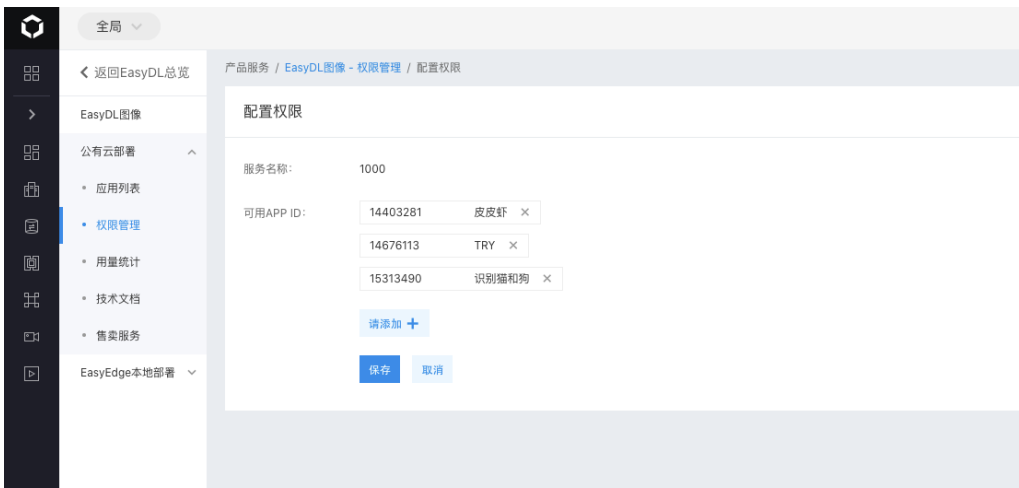
在正式使用之前，还需要做的一项工作为接口赋权，需要登录EasyDL控制台中创建一个应用，获得由一串数字组成的appid，然后就可以参考接口文档正式使用了



同时支持在「公有云服务管理」-「权限管理」中为第三方用户配置权限

示意图如下：





🔗 图像分类API调用文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可通过以下方式联系我们：

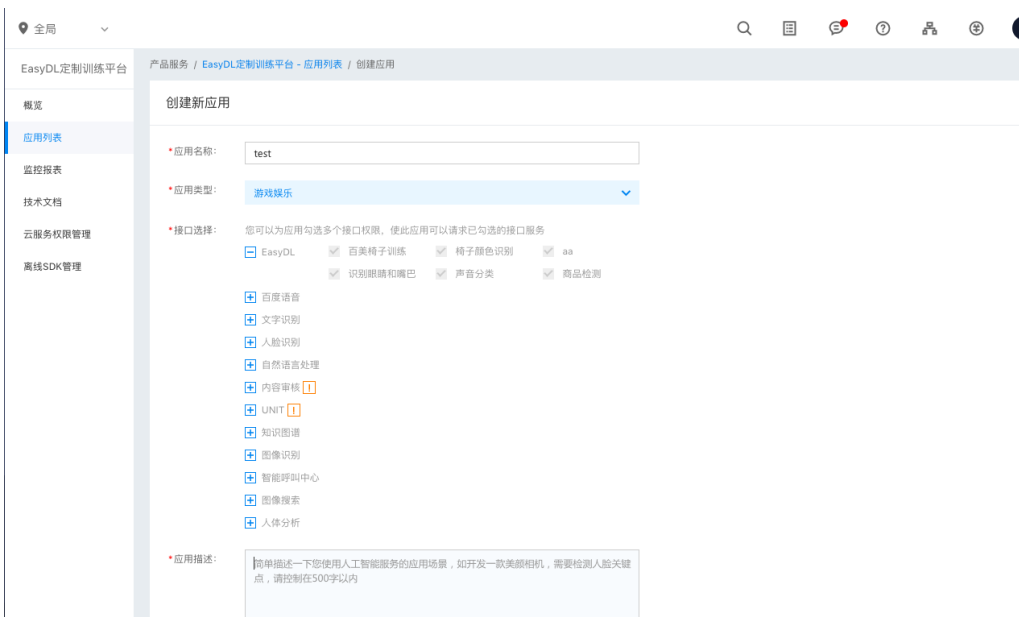
- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

接口描述

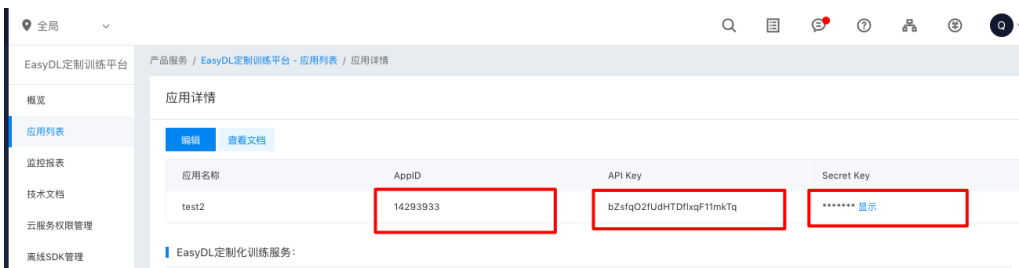
基于自定义训练出的图像分类模型，实现个性化图像识别。模型训练完毕后发布可获得定制化图像分类API

接口鉴权

1、在EasyDL控制台创建应用



2、应用详情页获取AK SK



请求说明

请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
input_type	当取值为 url 时，需在请求参数中传入图片的URL string
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
top_num	否	number	-	返回分类数量，默认为6个
url	否	string	-	如果在请求URL参数中增加“input_type=url”，则该参数必传，否则“image”参数必传。参数内容为URL string，用户需确保该string是有效的图片URL，否则会下载失败

请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

PHP
JAVA
Python3
C++

```

<?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = "", $param = "")
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
}

```

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```

{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}

```

需要重新获取新的Access Token再次请求即可。



错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg.bmp.png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	image字段缺失（未上传图片）

## 🔗 批量预测

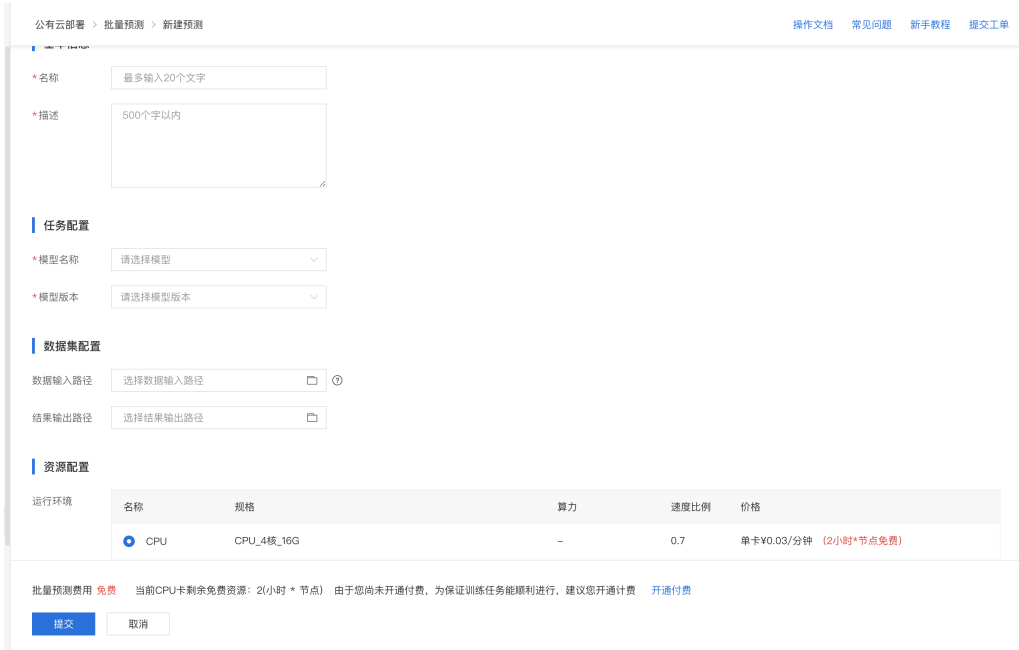
### 批量预测

部署方式为公有云服务的模型训练完成后即可使用批量预测服务。批量预测支持您将已导入百度云BOS数据存储的图片数据进行一次预测任务，任务完成后会将预测结果保存至百度云BOS数据存储服务中。您可按照如下步骤使用批量预测服务：

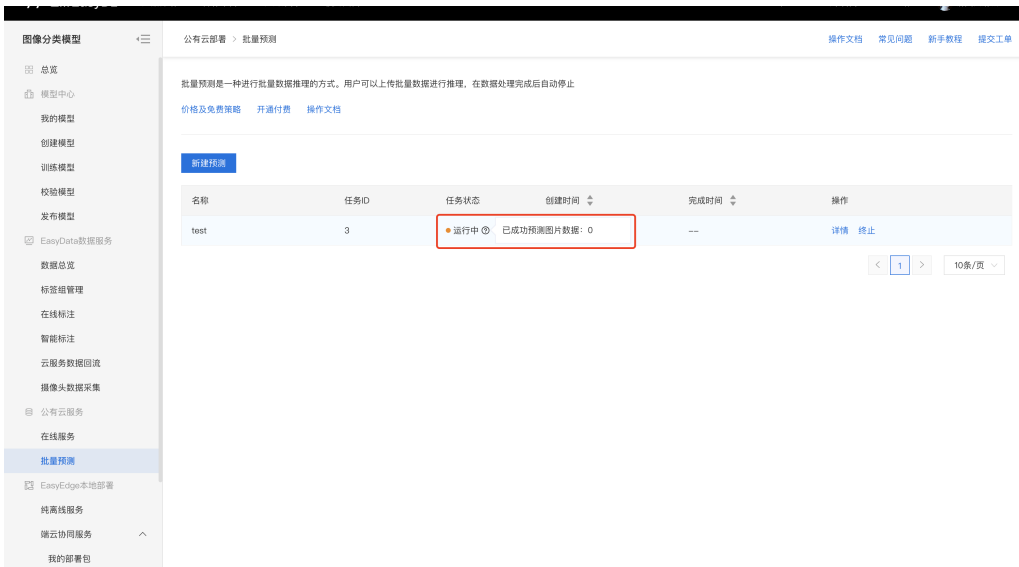
①**新建批量预测任务** 首次使用批量预测功能会提示您开通BOS权限，请按照说明指引开通。



请按照页面说明填写相关信息，选择预测模型以及所需预测的数据地址，并选择对应的资源配置。每个账户享有2个小时CPU算力的免费额度



②等待任务运行完成 可根据任务状态查看当前已成功预测的图片数量



③任务完成，查看数据 任务完成后可在任务配置时所选择的结果保存BOS地址中查看预测结果数据

🔗 本地服务器部署

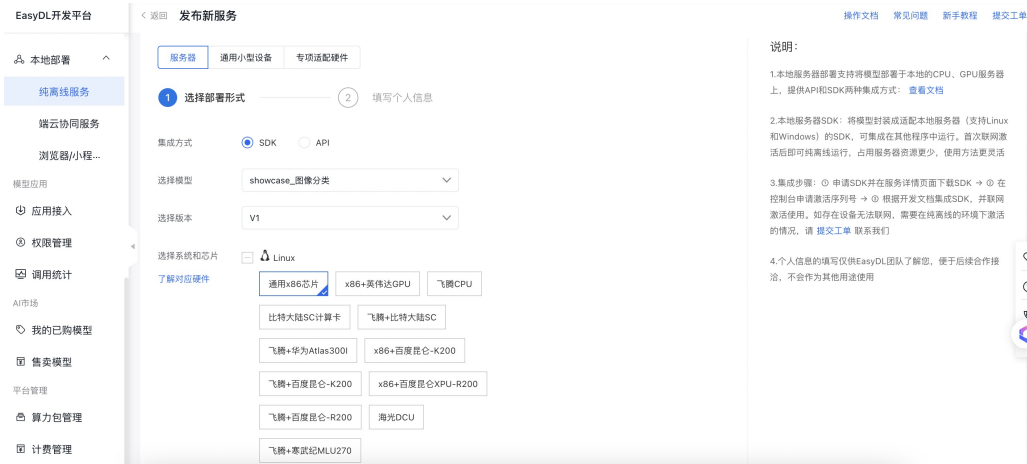
🔗 如何在本地服务器部署

训练完毕后，可以选择将模型通过「纯离线服务」或「端云协同服务」部署，具体介绍如下：

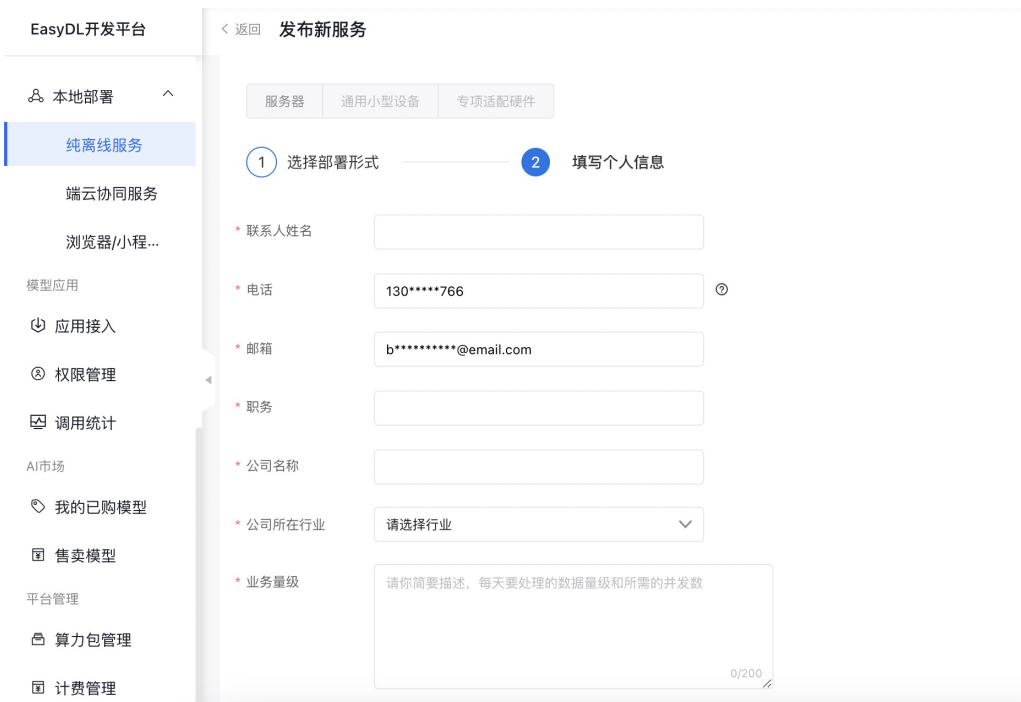
**纯离线服务部署**

可以在左侧导航栏中找到「纯离线服务」，依次进行以下操作即可将模型部署到本地服务器：

- 选择部署方式「服务器」
- 选择集成方式
- 选择模型、版本、系统和芯片
- 点击下一步



- 填写部分信息（注：个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用）
- 点击发布



### ① 私有API

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。点击「发布」后，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成。

### ② 服务器端SDK

将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活。

1. 点击「发布」后，前往[控制台](#)申请服务器端SDK的试用序列号。
2. 点击「新增测试序列号」，根据模型类型选择「序列号类型」，填写「新增设备数」（所得序列号数量），点击确定即可。



3、离线SDK的激活和使用，请参考文档完成集成



### 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

### 本地服务器部署价格说明

EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

如需购买永久使用授权，服务器SDK用户请在[控制台](#)点击「购买正式授权」，并按照对应步骤激活。

服务器API用户请微信搜索“BaiduEasyDL”添加小助手咨询，通过线下签订合同购买使用。

### 更多参考

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[纯离线SDK说明](#)

[纯离线SDK简介](#)

本文档主要说明定制化模型发布后获得的服务器端SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### SDK说明

图像分类服务器端SDK支持Linux、Windows两种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux		Intel CPU: x86_64 NVIDIA GPU: x86_64 HUAWEI Atlas 300: x86_64
Windows	64位 Windows7 及以上	NVIDIA GPU: x86_64  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015  GPU依赖： CUDA 9.x + cuDNN 7.x

#### 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)页面查看评测信息表

#### 激活&使用步骤

离线SDK的激活与使用分以下三步：

- ① 下载SDK后，在[控制台](#)获取序列号
- ② 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

- ③ 正式使用

#### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在[百度智能云控制台](#)内[提交工单](#)反馈

#### 1、激活失败怎么办？

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ④模型发布者和序列号所属账号非同一个账号，如果存在这种异常建议更换账号获取有效序列号
- ⑤序列号已过有效期，请更换序列号后重试
- ⑥如有其他异常请在[百度智能云控制台](#)内[提交工单](#)反馈

#### Windows集成文档

##### 简介

本文档介绍图像分类服务器端Windows SDK的使用方法。

- 硬件支持：

- NVIDIA GPU（普通版，加速版）
- 操作系统支持
  - 64位 Windows 7 及以上
  - 64位Windows Server 2012及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- GPU基础版（EasyEdge-win-x86-nvidia-gpu）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib：<http://www.winimage.com/zLibDll/zlib123dllx64.zip>，解压后将dll\_x64/zlibwapi.dll 拷贝到cuda的bin目录下）+ 硬件计算能力(<https://developer.nvidia.com/cuda-gpus#compute>)达6.1及以上
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + 硬件计算能力达7.5及以上
- GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.x.x
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.x.x
- GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.3.1 + 硬件计算能力达6.1及以上
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.1.6 + 硬件计算能力达7.5及以上
- GPU加速版（x86-nvidia-gpu-torch）
  - CUDA 11.0.x + cuDNN 8.0.5.x
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级，下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 优化模型算法；GPU CUDA9.0 CUDA10.0 标记为待废弃状态 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复个别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | GPU基础版推理引擎优化升级；GPU加速版支持自定义模型文件缓存路径；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | 修复已知问题 | | 2021-08-19 | 1.3.2 | 新增支持EasyDL小目标检测，新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | 修复已知问题 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020-12-18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020-10-29 | 1.1.19 | 修复已知问题 | | 2020-09-17 | 1.1.18 | 支持更多模型 | | 2020.08.11 | 1.1.17 | 支持专业版更多模型 | | 2020.06.23 | 1.1.16 | 支持专业版更多模型 | | 2020.05.15 | 1.1.15 | 更新加速版tensorrt版本，支持高精度检测 | | 2020.03.13 | 1.1.14 | 支持声音分类 | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | 支持物体检测高精度算法的CPU加速版，EasyDL 专业版支持 SDK 加速版 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版！ |

## 快速开始

### 1. 安装依赖

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

### 如果使用GPU版SDK，请安装CUDA + cuDNN

<https://developer.nvidia.com/cuda>  
<https://developer.nvidia.com/cudnn>

### 如果使用GPU版加速版SDK（EasyEdge-win-x86-nvidia-gpu-tensorrt），请安装TensorRT

<https://developer.nvidia.com/tensorrt>

根据cuda版本下载，下载后把lib目录下的所有dll，拷贝到SDK的dll目录下

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验”，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num，选择鉴权模式，点击“启动服务”，等待数秒即可启动成功，本地服务默认运行在

<http://127.0.0.1:24401/>

其他任何语言只需通过HTTP调用即可。

如启动失败，可参考如下步骤排查：



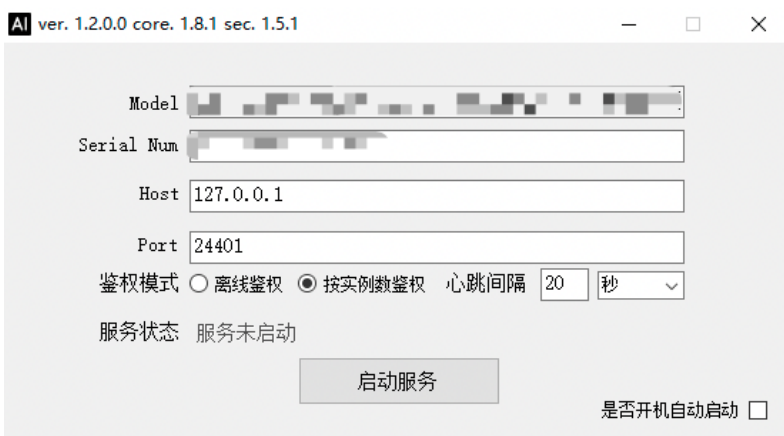
#### 2.1 离线鉴权（默认鉴权模式）

首次联网激活，后续离线使用



## 2.2 按实例数鉴权

周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

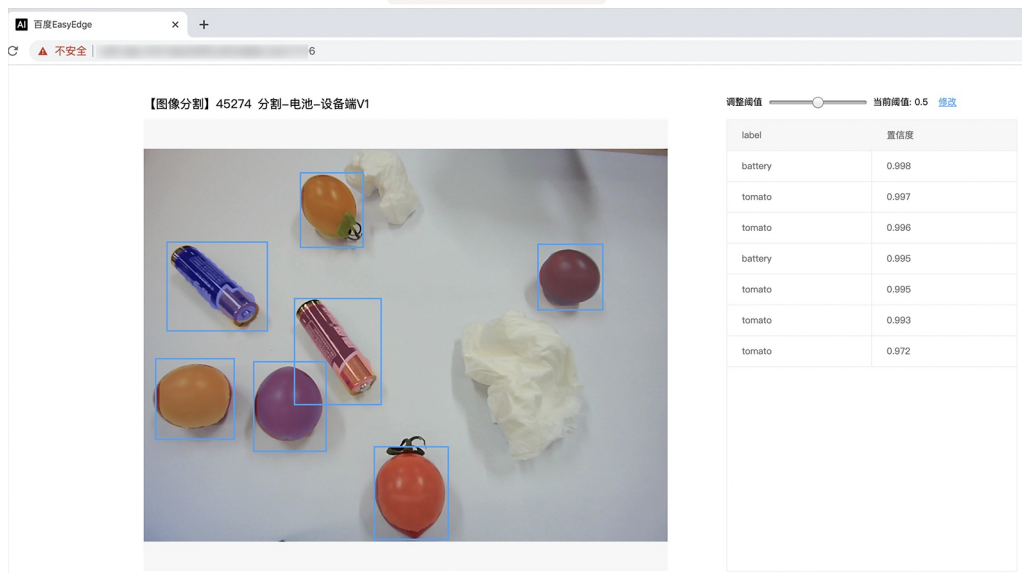
## 2.3 序列号激活错误码



错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。



#### 使用说明

#### 调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                        data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**

**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----|-----| | confidence | float | 0~1 | 分类的置信度 | | label | string | | 分类的类别 | | index | number | | 分类的类别 |

### 集成指南

#### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

#### 基于c++ dll集成

#### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

### 集成方法

参考src目录中的CMakeLists.txt进行集成

### 基于c# dll集成

### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

### FAQ

#### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
 .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

GPU依赖，版本必须如下： \* CUDA 11.0.x + cuDNN 8.4.x 或者 CUDA 11.7.x + cuDNN 8.4.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖，版本必须如下： \* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.x.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖，版本必须如下： \* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.3.1

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## Linux集成文档-C++

### 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：图像分类，物体检测，图像分割，目标追踪
- 硬件支持：
  - CPU 基础版：- intel x86\_64 \* - AMD x86\_64 - 龙芯 loongarch64 - 飞腾 aarch64

- CPU 加速版 - Intel Xeon with Intel®AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE - AMD Core Processors with AVX2
- NVIDIA GPU: x86\_64 PC
- 寒武纪 Cambricon MLU270
- 比特大陆计算卡SC5+
- 百度昆仑XPU K200
  - x86\_64 - 飞腾 aarch64 - 百度昆仑XPU R200
  - x86\_64 - 飞腾 aarch64
- 华为Atlas 300
- 海光DCU: x86\_64 PC
- 寒武纪 MLU370 on x86\_64
- 操作系统支持：Linux

根据开发者的选择，实际下载的版本可能是以下版本之一：

- EasyDL图像
  - x86 CPU 基础版
  - x86 CPU 加速版
  - Nvidia GPU 基础版
  - Nvidia GPU 加速版
  - x86 mlu270基础版
  - x86 SC5+基础版
  - Phytium MLU270基础版
  - Phytium XPU基础版
  - Phytium Atlas300I基础版
  - Hygon DCU基础版

性能数据参考[算法性能及适配硬件](#)

\*intel 官方合作，拥有更好的适配与性能表现。

#### Release Notes

时间	版本	说明
2023.0 8.31	1.8.3	Atlas系列Soc支持语义分割模型，Atlas Cann升级到6.0.1，昆仑XPU后端推理引擎升级
2023.0 6.29	1.8.2	模型压缩能力升级
2023.0 5.17	1.8.1	支持物体检测自定义四边形模型精度无损压缩发布x86 CPU版SDK
2023.0 3.16	1.8.0	支持图像分类精度提升包本地部署
2022.1 2.29	1.7.2	模型性能优化；推理库性能优化
2022.1 0.27	1.7.1	新增语义分割模型http请求示例；升级海光DCU SDK，需配套rocm4.3版本使用；Linux GPU基础版下线适用于CUDA10.0及以下版本的SDK；Linux GPU加速版升级推理引擎版本

2022.0 9.15	1.7.0	Linux GPU加速版升级预测引擎；Linux GPU加速版适用于CUDA9.0、CUDA10.0的SDK为deprecated，未来移除；新增实例分割高性能模型离线部署；性能优化
2022.0 7.28	1.6.0	Linux CPU普通版、Linux GPU普通/加速版、Jetson新增目标追踪模型接入实时流的demo
2022.0 5.27	1.5.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2022.0 5.18	1.5.0	GPU加速版max_batch_size参数含义变更；修复GPU加速版并发预测时部分图片结果预测错误及耗时增加问题；CPU普通版预测引擎升级；新增版本号头文件；新增飞腾Atlas300I支持，并且在EasyDL新增多种加速版本；示例代码移除frame_buffer，新增更安全高效的safe_queue；新增Tensor In/Out接口和Demo
2022.0 4.25	1.4.1	EasyDL, BML升级支持paddle2模型
2022.0 3.25	1.4.0	新增支持海光服务器搭配海光DCU加速卡；
2021.1 2.22	1.3.5	GPU加速版支持自定义模型文件缓存路径；新增支持飞腾MLU270服务器、飞腾XPU服务器
2021.1 0.20	1.3.4	CPU加速版推理引擎优化升级，新增支持飞腾CPU、龙芯CPU服务器、比特大陆计算卡SC5+ BM1684、寒武纪MLU270；大幅提升EasyDL GPU加速版有损压缩加速模型的推理速度
2021.0 8.19	1.3.2	CPU、GPU普通版及无损加速版新增支持EasyDL小目标检测，CPU普通版、GPU普通版支持检测模型的batch预测
2021.0 6.29	1.3.1	CPU普通版、GPU普通版支持分类模型的batch预测，CPU加速版支持分类、检测模型的batch预测；GPU加速版支持CUDA11.1；视频流解析支持调整分辨率；预测引擎升级
2021.0 5.13	1.3.0	新增视频流接入支持；模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告
2021.0 3.09	1.2.1	GPU新增目标追踪支持，http server服务支持图片通过base64格式调用，EasyDL高性能检测模型和均衡检测模型CPU加速版新增量化压缩模型
2021.0 1.27	1.1.0	EasyDL经典版分类高性能模型升级；部分SDK不再需要单独安装OpenCV
2020.1 2.18	1.0.0	1.0版本发布！安全加固升级、性能优化、引擎升级、接口优化等多项更新
2020.1 1.26	0.5.8	EasyDL经典版分类模型CPU加速版里新增量化压缩模型
2020.1 0.29	0.5.7	新增CPU加速版支持：EasyDL经典版高精度、超高精度物体检测模型和EasyDL经典版图像分割模型
2020.0 9.17	0.5.6	性能优化，支持更多模型
2020.0 8.11	0.5.5	提升预测速度；支持百度昆仑芯片
2020.0 5.15	0.5.3	优化性能，支持专业版更多模型
2020.0 4.16	0.5.2	支持CPU加速版；CPU基础版引擎升级；GPU加速版支持多卡多线程
2020.0 3.12	0.5.0	x86引擎升级；更新本地http服务接口；GPU加速版提速，支持批量图片推理
2020.0 1.16	0.4.7	ARM引擎升级；增加推荐阈值支持
2019.1 2.26	0.4.6	支持海思NNIE
2019.1 1.02	0.4.5	移除curl依赖；支持自动编译OpenCV；支持EasyDL 专业版 Yolov3；支持EasyDL经典版高精度物体检测模型升级
2019.1		

2019.1 0.25	0.4.4	ARM引擎升级,性能提升30%;支持EasyDL专业版模型
2019.0 9.23	0.4.3	增加海思NNIE加速芯片支持
2019.0 8.30	0.4.2	ARM引擎升级;支持分类高性能与高精度模型
2019.0 7.25	0.4.1	引擎升级,性能提升
2019.0 7.25	0.4.0	支持Xeye,细节完善
2019.0 6.11	0.3.3	paddle引擎升级;性能提升
2019.0 5.16	0.3.2	新增NVIDIA GPU支持;新增armv7l支持
2019.0 4.25	0.3.1	优化硬件支持
2019.0 3.29	0.3.0	ARM64 支持;效果提升
2019.0 2.20	0.2.1	paddle引擎支持;效果提升
2018.1 1.30	0.1.0	第一版!

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE含义变更。变更前:预测输入图片数不大于该值均可。变更后:预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理,在输入图片数小于该值时依然可正常运行,但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值,尽可能保持最小。

2020-12-18: 【接口升级】参数配置接口从1.0.0版本开始已升级为新接口,以前的方式被置为deprecated,并将在未来的版本中移除。请尽快考虑升级为新的接口方式,具体使用方式可以参考下文介绍以及demo工程示例,谢谢。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES。更新模型时,如果SDK版本号有更新,请务必同时更新SDK,旧版本的SDK可能无法正确适配新发布出来部署包中的RES模型。

## 快速开始

SDK在以下环境中测试通过

- x86\_64, Ubuntu 16.04, gcc 5.4
- x86\_64, Ubuntu 18.04, gcc 7.4
- Tesla P4, Ubuntu 16.04, cuda 9.0, cudnn 7.5
- x86\_64, Ubuntu 16.04, gcc 5.4, XTCL r1.0
- aarch64, Kylin V10, gcc 7.3
- loongarch64, Kylin V10, gcc 8.3
- Bitmain SC5+ BM1684, Ubuntu 18.04, gcc 5.4
- x86\_64 MLU270, Ubuntu 18.04, gcc 7.5
- phytiun MLU270, Kylin V10, gcc 7.3.0
- phytiun XPU, Kylin V10, gcc 7.3.0
- hygon DCU, CentOS 7.8 gcc 7.3.0

- XPU K200, x86\_64, Ubuntu 18.04
- XPU K200 aarch64, Ubuntu 18.04
- XPU R200, x86\_64, Ubuntu 18.04
- XPU R200 aarch64, Ubuntu 18.04
- MLU370, x86\_64, Centos7.6.1810

#### 依赖包括

- cmake 3+
- gcc 5.4 (需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.11 (可选)
- cuda && cudnn (使用NVIDIA-GPU时必须, SDK内提供多个Cuda版本推理套件, 根据需要安装依赖的Cuda和Cudnn版本)
- XTCL 1.0.0.187 (使用昆仑服务器时必须)
- Rocm4.3, Miopen 2.14(使用海光DCU服务器时必须)

### 1. 安装依赖

以下步骤均可选, 请开发者根据实际运行环境选择安装。

#### (可选) 安装cuda&cudnn

##### 在NVIDIA GPU上运行必须(包括GPU基础版, GPU加速版)

对于GPU基础版, 若开发者需求不同的依赖版本, 请在[PaddlePaddle官网](#) 下载对应版本的libpaddle\_fluid.so或参考其文档进行编译, 覆盖lib文件夹下的相关库文件。

#### (可选) 安装TensorRT

##### 在NVIDIA GPU上运行GPU加速版必须

下载包中提供了对应 cuda9.0、cuda10.0、cuda10.2、cuda11.0+四个版本的 SDK, cuda9.0 和 cuda10.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.0.0.11, cuda10.2 及以上的 SDK 默认依赖的 TensorRT 版本为 TensorRT8.4, 请在[这里](#)下载对应 cuda 版本的 TensorRT, 并把其中的lib文件拷贝到系统lib目录, 或其他目录并设置环境变量。

(可选) 安装XTCL 使用昆仑服务器及对应SDK时必须 请安装与1.0.0.187版本兼容的XTCL。必要时, 请将运行库路径添加到环境变量。

#### (可选) 安装Rocm、Miopen

##### 使用海光DCU服务器对应SDK时必须

海光DCU SDK依赖Rocm 4.3和Miopen 2.14版本, 推荐使用easyedge镜像

(registry.baidubce.com/easyedge/hygon\_dcu\_infer:1.0.2.rocm4.3), SDK镜像内运行, 镜像拉取方式(wget https://aipe-easyedge-public.bj.bcebos.com/dcu\_docker\_images/hygon\_dcu\_rocm4.3.tar.gz && docker load -i hygon\_dcu\_rocm4.3.tar.gz), 关于海光DCU使用更多细节可参考[paddle文档](#)

### 2. 使用序列号激活 请在官网获取序列号

**纯离线服务说明**

发布纯离线服务, 将训练完成的模型部署在本地, 离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体方案专项适配硬件上。通过API, SDK进一步集成, 灵活适应不同业务场景。

[发布前准备](#) [控制台](#)

---

**服务器** 通用小型设备 专项适配硬件

[SDK](#) [API](#)

此处发布、下载的SDK为未授权SDK, 需要前往控制台[获取序列号](#)激活后才能正常使用。SDK内附有对应版本的Demo及开发文档, 开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
su_x小目标检测	134319-V1 <a href="#">查看详情报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

SDK内bin目录下提供预编译二进制文件, 可直接运行(二进制运行详细说明参考下一小节), 用于图片推理和模型http服务, 在二进制参数的



serial\_num(或者serial\_key)处填入序列号可自动完成联网激活（请确保硬件首次激活时能够连接公网，如果确实不具备联网条件，需要使用纯离线模式激活，请下载使用百度智能边缘控制台纳管SDK）

```
**SDK内提供的一些二进制文件，填入序列号运行可自动完成激活，以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}
```

如果是基于源码集成，设置序列号方法如下

```
global_controller()->set_licence_key("")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量或者源码设置）实例数鉴权环境变量设置方法

```
export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30
```

实例数鉴权源码设置方法

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)
```

3. 基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；在Intel CPU上运行CPU加速版，如果thirdparty里包含openvino文件夹的，必须在编译或运行demo程序前执行以下命令：source \${cpp\_kit位置路径}/thirdparty/openvino/bin/setupvars.sh 或者执行 source \${cpp\_kit位置路径}/thirdparty/openvino/setupvars.sh(openvino-2022.1+) 如果SDK内不包含setupvars.sh脚本，请忽略该提示

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号首次激活需要使用，激活后可不用填序列号也能运行二进制)

```
**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=./lib ./easyedge_image_inference ../.././RES /xxx/cat.jpeg "1111-1111-1111-1111"
```

demo运行效果：



图片加载失败

```
> ./easyedge_image_inference ../.././RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

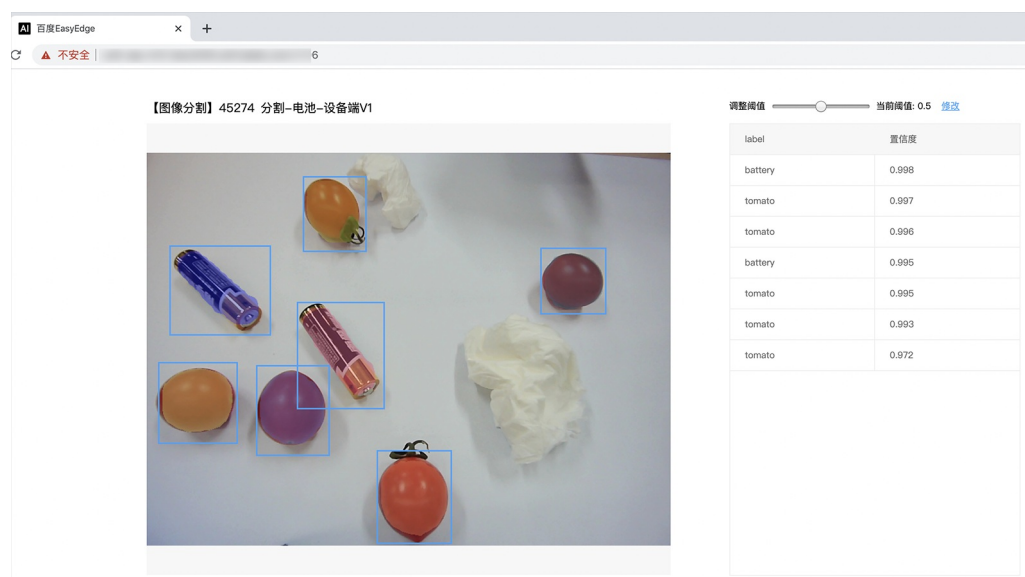
```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=./lib ./edgekit_serving --cfg=./edgekit_serving.yml

**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=./lib ./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

HTTP(or Webservice) is now serving at 0.0.0.0:24401

字样，此时，开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试，网页右侧会展示模型推理结果



【图像分割】45274 分割-电池-设备端V1

调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

对于目标追踪的模型，请选择一段视频，并耐心等待结果



同时，可以调用HTTP接口来访问服务。

**请求http服务** 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

python示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }

        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。编译demo项目 SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```

cd src
mkdir build && cd build
cmake .. && make
./easymage_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easymage_image_inference {模型RES文件夹} {测试图片路径}

```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEDGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 1: 配置模型资源目录
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 在这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}
}
```

## 输入图片不限制大小

**SDK参数配置** SDK的参数通过EdgePredictorConfig::set\_config和global\_controller()->set\_config配置。set\_config的所有key在easyedge\_xxxx\_config.h中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过EdgePredictorConfig::set\_config设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过global\_controller()->set\_config设置

以序列号为例，KEY的说明如下：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";
```

使用方法如下：

```
EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");
```

具体支持的运行参数配置列表可以参考开发工具包中的头文件的详细说明。

相关配置均可以通过环境变量的方法来设置，对应的key名称加上前缀EDGE\_即为环境变量的key。如序列号配置的环境变量key为EDGE\_PREDICTOR\_KEY\_SERIAL\_NUM，如指定CPU线程数的环境变量key为EDGE\_PREDICTOR\_KEY\_CPU\_THREADS\_NUM。注意：通过代码设置的配置会覆盖通过环境变量设置的值。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image, std::vector<std::vector<EdgeResultData>>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测、图像分割时才有意义
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割的模型, 该字段才有意义
    // 请注意: 图像分割时, 以下两个字段会比较大, 使用完成之后请及时释放EdgeResultData
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask

    // 目标追踪模型, 该字段才有意义
    int trackid; // 轨迹id
    int frame; // 处于视频中的第几帧
    EdgeTrackStat track_stat; // 跟踪状态
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

cv::Mat mask为图像掩码的二维数组

```

{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}

```

其中1代表为目标区域, 0代表非目标区域

### 关于图像分割mask\_rle

该字段返回了mask的游程编码, 解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding, 此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};          // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;            // frame存储为视频文件的路径
    bool save_all{false};             // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被



抽取返回,以作为显示或存储用。 `input_fps` : 用于抽帧前设置fps。 `resolution` : 设置摄像头采样的分辨率,其值请参考`easyedge_video.h`中的定义,注意该分辨率调整仅对输入源为摄像头时有效。 `conf` : 高级选项。部分配置会通过该map来设置。

### 注意:

1. 如果使用VideoConfig的display功能,需要自行编译带有GTK选项的opencv,默认打包的opencv不包含此项。
2. 使用摄像头抽帧时,如果通过`resolution`设置了分辨率调整,但是不起作用,请添加如下选项:

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容,如遇到问题,可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程,可以参考SDK中的`demo_video_inference`。

### 设置序列号

请在网页控制台中申请序列号,并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

1. 开启http服务 http服务的启动可以参考`demo_serving.cpp`文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里,图片的解码运行在cpu之上,可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量,根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

### 2. http接口详细说明

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片或视频来进行测试。

http 请求方式一: 无额外编码 URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供,则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img.json())
```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
```

Python请求示例 (视频测试, 注意: 区别于图片预测, 需指定Content-Type; 否则会调用图片推理接口)

```
import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data.json())
```

http 请求方法二: json格式, 图片传base64格式字符串 HTTP方法: POST Header如下:

参数	值
Content-Type	application/json

Body请求填写:

- 图像分类网络: body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据, base64编码, 要求base64图片编码后大小不超过4M,最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量, 不填该参数, 则默认返回全部分类结果

- 物体检测和实例分割网络: Body请求示例:

```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)

```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型，同其他CV模型不同，语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
        with open("gray_result.png", "wb") as fb:
            fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    }
  ]
}
```

其他配置

### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



### 2. CPU线程数设置

CPU线程数可通过 EdgePredictorConfig::set\_config配置

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_CPU_THREADS_NUM, 4);
```

### 3. 批量预测设置

```
int batch_size = 2; // 使用前修改batch_size再编译、执行
while (get_next_batch(imgs, img_files, batch_size, start_index)) {
  ...
}
```

**GPU 加速版 预测接口** GPU 加速版 SDK 除了支持上面介绍的通用接口外，还支持图片的批量预测，预测接口如下：

```

/**
 * @brief
 * GPU加速版批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& result
) = 0;

/**
 * @brief
 * GPU加速版批量图片推理接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;

```

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE`，其含义见下方参数配置接口的介绍。

**运行参数选项** 在上面的内容中我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对GPU加速版开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值：4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值：false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1：如果当前max_batch_size与历史编译产生的max_batch_size不相等时，则重新编译模型（推荐）
 * 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
 * 值类型: int
 * 默认值：1
 */

```

```

static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名, 默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置; 序列号不设置留空时, SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**: 首次加载模型会先对模型进行编译优化, 通过此值可以设置优化后的产出文件名, 这在多进程加载同一个模型的时候是有用的。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**: 首次加载模型经过编译优化后, 产生的优化文件会存储在这个位置, 可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**: 设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**: 此值用来控制批量图片预测可以支持的最大图片数, 实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**: 设置需要使用的 GPU 卡号。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**: 模型编译等级。通常模型的编译会比较慢, 但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制。当此值为 0 时, 表示忽略当前设置的 `max_batch_size` 而仅使用历史产出 (无历史产出时则编译模型); 当此值为 1 时, 会比较历史产出和当前设置的 `max_batch_size` 是否相等, 如不等, 则重新编译; 当此值为 2 时, 无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**: 通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量, 其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源, 建议结合实际使用控制此值, 使用多少则设置多少。注意: 此值的增加会降低单次 infer 的速度, 建议优先考虑 batch inference 和 multi predictor。

**PREDICTOR\_KEY\_GTURBO\_FP16**: 默认是 fp32 模式, 置 true 可以开启 fp16 模式预测, 预测速度会有所提升, 但精度也会略微下降, 权衡使用。注意: 不是所有模型都支持 fp16 模式。目前已知不支持fp16的模型包括: 图像分类高精度模型。

**多线程预测** GPU 加速版 SDK 的多线程分为单卡多线程和多卡多线程两种。单卡多线程: 创建一个 predictor, 并通过

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY** 控制单卡所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。多卡多线程: 多卡的

支持是通过创建多个 predictor，每个 predictor 对应一张 GPU 卡，predictor 的创建和 init 的调用放在主线程，通过多线程的方式调用 infer 接口。

**已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时，部分结果错误** A：EasyDL图像分类高精度模型在有些显卡上可能存在此问题，可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

**2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object** A：部分显卡存在此问题，如果遇到此问题，请确认没有频繁调用 init 接口，通常调用 infer 接口即可满足需求。

**3. 开启 fp16 后，预测结果错误** A：不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括：图像分类高精度模型。目前不支持的将在后面的版本陆续支持。

**昆仑服务器** 昆仑服务器SDK支持将EasyDL的模型部署到昆仑服务器上。SDK提供的接口风格一致，简单易用，轻松实现快速部署。Demo的测试可参考上文中的测试Demo部分。

**参数配置接口** 在上面的内容中我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对昆仑服务器开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * 使用哪张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 设置需要同时预测的图片数量
 * 值类型：int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_KUNLUN_BATCH_SIZE = "PREDICTOR_KEY_KUNLUN_BATCH_SIZE";
```

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的加速卡的卡号。

**PREDICTOR\_KEY\_KUNLUN\_BATCH\_SIZE**：设置单次预测可以支持的图片数量。

使用方法：

```
int batch_size = 1;
config.set_config(easyedge::params::PREDICTOR_KEY_KUNLUN_BATCH_SIZE, batch_size);
```

**模型调优** 通过设置如下环境变量，可以在初始化阶段对模型调优，从而让预测的速度更快。

```
export XPU_CONV_AUTOTUNE=5
```

## FAQ

### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

## 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

## 3. NVIDIA GPU预测时，报错显存不足 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请根据显存大小和模型配置。调整合适的初始 fraction\_of\_gpu\_memory。参数的含义参考[这里](#)。

## 4. 如何将我的模型运行为一个http服务？目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

## 5. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

## 6. 运行SDK报错 Authorization failed

情况一：日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

## 7. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

## 8. 运行二进制时，提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 运行二进制时提示 libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 同上面8的问题类似，没有正确设置动态库的查找路径，可通过设置LD\_LIBRARY\_PATH为sdk的thirdparty/opencv/lib文件夹解决



```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/thirdparty/opencv/lib
(tips: 上面冒号后面接的thirdparty/opencv/lib路径以实际项目中路径为准, 比如也可能是../thirdparty/opencv/lib)
```

10. 编译时报错: **file format not recognized** 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中, 再解压缩、编译

11. 进行视频解码时, 报错符号未找到、格式不支持、解析出的图片为空、无法设置抽帧 请确保安装OpenCV时, 添加了-DWITH\_FFMPEG=ON选项 (或者GStream选项), 并且检查OpenCV的安装日志中, 关于Video I/O段落的说明是否为YES。

```
-- Video I/O:
-- DC1394:          YES (ver 2.2.4)
-- FFMPEG:          YES
-- avcodec:         YES (ver 56.60.100)
-- avformat:        YES (ver 56.40.101)
-- avutil:          YES (ver 54.31.100)
-- swscale:         YES (ver 3.1.101)
-- avresample:      NO
-- libv4l/libv4l2:  NO
-- v4l/v4l2:        linux/videodev2.h
```

如果为NO, 请搜索相关解决方案, 一般为依赖没有安装, 以apt为例:

```
apt-get install yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
```

12. GPU加速版运行有损压缩加速的模型, 运算精度较标准模型偏低 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除, 并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true, 使用FP16的运算精度重新评估模型效果。若依然不理想, 可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false, 从而使用更高精度的FP32的运算精度。

## Linux集成文档-Python

### 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法, 适用于 EasyDL 和 BML。

EasyDL 通用版:

- 网络类型支持: 图像分类, 物体检测, 图像分割, 声音分类, 表格预测
- 硬件支持:
  - Linux x86\_64 CPU (基础版, 加速版)
  - Linux x86\_64 Nvidia GPU (基础版, 加速版)
- 语言支持: Python 3.5, 3.6, 3.7, 3.8, 3.9

BML:

- 网络类型支持: 图像分类, 物体检测, 声音分类, 表格预测
- 硬件支持:
  - Linux x86\_64 CPU (基础版)
  - Linux x86\_64 Nvidia GPU (基础版)
- 语言支持: Python 3.5, 3.6, 3.7, 3.8, 3.9

### Release Notes

时间	版本	说明
2023-03-16	1.3.7	迭代升级，新增支持文本类模型；新增GPU 多卡多进程推理demo
2022.10.27	1.3.5	新增华为Atlas300、飞腾Atlas300 Python SDK，支持图像分类、物体检测、人脸检测、实例分割
2022.09.15	1.3.3	EasyDL CPU普通版新增支持表格预测
2022.05.27	1.3.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2021.12.22	1.2.7	声音分类模型升级
2021.10.20	1.2.6	CPU基础版、CPU加速版、GPU基础版推理引擎优化升级
2021.08.19	1.2.5	CPU基础版、CPU无损加速版、GPU基础版新增支持EasyDL小目标检测
2021.06.29	1.2.4	CPU、GPU新增EasyDL目标跟踪支持；新增http server服务启动demo
2021.03.09	1.2.2	EasyDL CPU加速版新增支持分类、高性能检测和均衡检测的量化压缩模型
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.03.12	1.1.14	新增声音识别python sdk
2020.02.12	1.1.13	新增口罩模型支持
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持 SDK 加速版
2019.12.04	1.1.10	支持图像分割
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.08.29	1.1.8	CPU 加速版支持
2019.07.19	1.1.7	提供模型更新工具
2019.05.16	1.1.3	NVIDIA GPU 支持
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】 序列号的配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。
- 使用声音分类SDK需要安装额外依赖
  - \* pip 安装 `resampy pydub six librosa` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg (windows系统的ffmpeg已基于sdk中无需额外安装，linux系统需要手动安装)
- 使用表格预测SDK需要安装额外依赖
 

```
pip安装brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3 catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2
psutil==5.7.2 pymml==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0
```

 另外ml算法安装（目前只支持python3.7）

```
pip install BaiduAI_TabularInfer-0.0.0-cp37-cp37m-linux_x86_64.whl
```

### 安装 paddlepaddle

- 使用x86\_64 CPU 基础版 预测时必须安装（目标跟踪、表格预测除外）：

```
python -m pip install paddlepaddle==2.2.2 -i https://mirror.baidu.com/pypi/simple
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU 基础版 预测时必须安装（目标跟踪、表格预测除外）：

```
python -m pip install paddlepaddle-gpu==2.2.2.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA10.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2 -i https://mirror.baidu.com/pypi/simple #CUDA10.2的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post110 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.0的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post111 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post112 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.2的PaddlePaddle
```

不同cuda版本的环境，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持

安装 OpenVINO 使用x86\_64 CPU 加速版 SDK 预测时必须安装。

1) 请参考 [OpenVINO toolkit 文档](#)安装 2021.4版本, 安装时可忽略Configure the Model Optimizer及后续部分

2) 运行之前，务必设置环境变量

```
source /opt/intel/opencvino_2021/bin/setupvars.sh
```

### 安装 cuda、cudnn

- 使用Nvidia GPU 加速版 预测时必须安装。依赖的版本为 cuda9.0、cudnn7。版本号必须正确。

### 安装 pytorch (torch >= 1.7.0)

- 目标跟踪模型的预测必须安装pytorch版本1.7.0及以上（包含：Nvidia GPU 基础版、x86\_64 CPU 基础版）。
- 目标跟踪模型Nvidia GPU 基础版 还需安装依赖cuda、cudnn。

关于不同版本的pytorch和CUDA版本的对应关系：[pytorch官网](#) 目标跟踪模型还有一些列举在requirements.txt里的依赖（包括torch >= 1.7.0），均可使用pip下载安装。

```
pip3 install -r requirements.txt
```

## 2. 安装 easyedge python wheel 包 安装说明

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。安装说明：华为 Atlas300 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Atlas300-{版本号}-cp36-cp36m-linux_x86_64.whl
```

安装说明：飞腾 Atlas300 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Phytium.Atlas-[版本号]-cp36-cp36m-linux_aarch64.whl
```

### 3. 使用序列号激活

#### 获取序列号

**将离线服务说明**

发布将离线服务，将训练完成的模型部署在本地，离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体方案专项适配硬件上。通过API、SDK进一步集成，灵活适应不同业务场景。

[发布新设备](#) [控制台](#)

**服务器** 通用小型设备 专项适配硬件

**SDK** **API**

此页发布、下载的SDK为本授权SDK，需要前往控制台[获取序列号](#)激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标检测	134319-V1 <a href="#">查看任务报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		高伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

#### 修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

### 4. GPU 加速版 使用 GPU 加速版，在安装完 whl 之后，必须：

1. 从[这里](#)下载 TensorRT7.0.0.11 for cuda9.0，并把解压后的 lib 放到 C++ SDK 的 lib 目录或系统 lib 目录
2. 运行时，必须在系统库路径中包含 C++ SDK 下的 lib 目录。如设置 LD\_LIBRARY\_PATH

```
cd ${SDK_ROOT}

**1. 安装 python wheel 包**
tar -xzf python/*.tar.gz
pip install -U {对应 Python 版本的 wheel 包}

**2. 设置 LD_LIBRARY_PATH**
tar -xzf cpp/*.tar.gz
export EDGE_ROOT=$(readlink -f $(ls -h | grep "baidu_easyedge_linux_cpp"))
export LD_LIBRARY_PATH=${EDGE_ROOT}/lib

**3. 运行 demo**
python3 demo.py {RES文件夹路径} {测试图片路径}
```

如果是使用 C++ SDK 自带的编译安装的 OpenCV，LD\_LIBRARY\_PATH 还需要包括 C++ SDK 的 build 目录下的 thirdparty/lib 目录

如果没有正确设置 LD\_LIBRARY\_PATH，运行时可能报错：

```
ImportError: libeasyedge.so.0.4.3: cannot open shared object file: No such file or directory
ImportError: libopencv_core.so.3.4: cannot open shared object file: No such file or directory
```

### 5. 测试 Demo

#### 5.1 图片预测

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



## 5.2 视频预测 (适用于目标跟踪)

输入对应的模型文件夹 (默认为RES) 和测试视频文件路径 / 摄像头id / 网络视频流地址, 运行:

```

**video_type: 输入源类型 type:int**
**1 本地视频文件**
**2 摄像头的index**
**3 网络视频流**
**video_src: 输入源地址, 如视频文件路径、摄像头index、网络流地址 type: string**
python3 demo.py {model_dir} {video_type} {video_src}

```

## 5.3 表格预测

输入对应模型文件夹 (默认为RES) 和测试数据地址 (csv文件地址), 运行:

```
python3 demo.py {model_dir} {/xxx/xxx.csv}
```

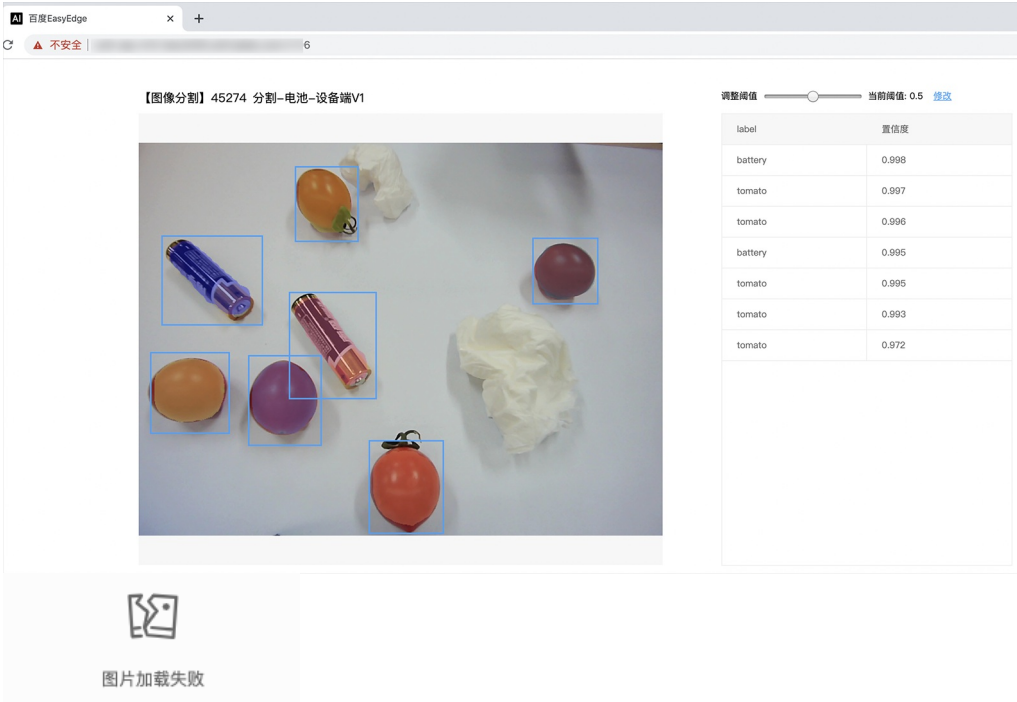
6. 测试Demo HTTP 服务 输入对应的模型文件夹 (默认为RES)、序列号、设备ip和指定端口号, 运行:

```
python3 demo_serving.py {model_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

后, 会显示:

```
Running on http://0.0.0.0:24401/
```

字样, 此时, 开发者可以打开浏览器, <http://{设备ip}:24401>, 选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里 `http_client_test()`函数请求http服务进行推理。



【图像分割】 45274 分割-电池-设备端V1

调整阈值: 0.5 修改

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

图片加载失败

## 使用说明

使用流程 `demo.py`

```

import BaiduAI EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image({numpy.ndarray的图片})
pred.close()

```

`demo_serving.py`

```
import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir=(RES文件夹路径), license=serial_key)
**请参考同级目录下demo.py里:**
**pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)**
**对以下参数device\device_id和engine进行修改**
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
```

## 初始化

- 接口

```
def init(
    self,
    model_dir,
    device=Device.CPU,
    engine=Engine.PADDLE_FLUID,
    config_file="conf.json",
    preprocess_file="preprocess_args.json",
    model_file="model",
    params_file="params",
    label_file="label_list.txt",
    infer_cfg_file="infer_cfg.json",
    device_id=0,
    thread_num=1,
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device, 比如 : Device.CPU
        engine: BaiduAI.EasyEdge.Engine, 比如 : Engine.PADDLE_FLUID
        config_file: str
        preprocess_file: str
        model_file: str
        params_file: str
        label_file: str 标签文件
        infer_cfg_file: 包含预处理、后处理信息的文件
            device_id: int 设备ID
            thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success
    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

## 预测图像

- 接口

```
def infer_image(self, img, threshold=0.3, channel_order="HWC", color_format="BGR", data_type="numpy"):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list
    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

**预测视频**（目前仅限目标跟踪模型调用）

- 接口

```
def infer_frame(self, frame, threshold=None):
    """
    视频推理(抽帧之后)
    :param frame:
    :param threshold:
    :return:
    """
```

- 返回格式dict

字段	类型	说明
pos	dict1	当前帧每一个类别的追踪目标的像素坐标(tlwh)
id	dict2	当前帧每一个类别的追踪目标的id
score	dict3	当前帧每一个类别的追踪目标的识别置信度
label	dict4	class_idx(int)与label(string)的对应关系
class_num	int	追踪类别数

**预测声音**

- 使用声音分类SDK需要安装额外依赖 pip 安装 `resampy pydub` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已集成在sdk中无需额外安装，linux系统需要手动安装）

- 接口

```
def infer_sound(self, sound_binary, threshold=0.3):
    """
    Args:
        sound_binary: sound_binary
        threshold: confidence

    Returns:
        list
    """
```



- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类的置信度
label	string		分类的类别
index	number		分类的类别

### 表格预测

- 考虑到表格字段内容和长度的不固定性，我们建议您参考“校验服务”页面提供的详细信息。您可以访问该页面：<https://ai.baidu.com/easydl/app/validate/ml/models/verify>，并从中复制数据请求的 Body 部分作为参考模板。这将帮助您理解如何灵活处理各种不同的字段。
- 接口

```
def infer_csv(self, data):
    """
    结构化数据推理
    Args:
        data: pd.DataFrame or list or dict
    Returns:
    """
```

- 返回格式: list 接口直接反馈预测结果数组

**升级模型** 适用于经典版升级模型，执行bash update\_model.sh，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

### FAQ

**Q: EasyDL 离线 SDK 与云服务效果不一致，如何处理？** A: 后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**Q: 运行时报错 "非法指令" 或 "illegal instruction"** A: 可能是 CPU 缺少 avx 指令集支持，请在[paddle官网](#) 下载 noavx 版本覆盖安装

**Q: NVIDIA GPU预测时，报错显存不足：** A: 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过export FLAGS\_fraction\_of\_gpu\_memory\_to\_use=0.3来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 allocate 更多的显存。

**Q: 我想使用多线程预测，怎么做？** 如果需要多线程预测，可以每个线程启动一个Program实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的thread\_num参数。

**Q: 运行SDK报错 Authorization failed**

**情况一：**日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：**日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更

- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

情况三：Atlas Python SDK日志提示 `ImportError: libavformat.so.58: cannot open shared object file: No such file or directory` 或者其他类似so找不到 可以在 `LD_LIBRARY_PATH`环境变量加上 `libs`和 `thirdpartylibs`路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内 `libs`和 `thirdpartylibs`路径的方法如下(以华为Atlas300 SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas300 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## Linux集成文档-Atlas

### 简介

本文档介绍EasyEdge/EasyDL的Linux Atlas SDK的使用方法。

注意Atlas有两种产品形态，Atlas 200和Atlas 300，请参见此处的[文档说明](#)

- 网络类型支持：图像分类
- 硬件支持：
  - CPU: aarch64
  - Atlas 300 卡
- 操作系统支持：Atlas指定的Linux版本，Ubuntu 16.04 x86\_64 或 centos 7 x86\_64，请从Atlas文档中下载。

### Release Notes

时间	版本	说明
2020.3.23	0.1	初始版本，支持图像分类

### 性能数据

数据仅供参考，实际数值根据使用线程数、利用率等情况可能有所波动

模型类型	模型算法	芯片类型	SDK类型	实测硬件	单次预测耗时
EasyDL 图像分类	高性能	Atlas 300	Atlas 300	Atlas 800服务器	9ms
EasyDL 图像分类	高精度	Atlas 300	Atlas 300	Atlas 800服务器	12ms
EasyDL 物体检测	高性能	Atlas 300	Atlas 300	Atlas 800服务器	11ms
EasyDL 物体检测	高精度	Atlas 300	Atlas 300	Atlas 800服务器	31ms

### atlas 300 加速卡注意事项

一般服务器（HOST侧）安装多个300加速卡，每个300加速卡有4个芯片。一个芯片（DEVICE侧）可以认为是一个单独的系统，并且不共享储存系统。

每个芯片都有独立的device-id，可以通过命令查看：`sudo npu-smi info`

由于模型需要在芯片上运行。因此运行SDK前，需要手动将模型复制到每个单独芯片的储存系统上。

### 测试atlas 300的官方demo

### 环境准备

请参见此处的[文档说明](#)，搭建环境，测试HelloDavinci demo通过后，再测试本demo

## 修改300加速卡SSH密码 (可选)

请在咨询华为技术人员后, 修改Device登录密码

```
ssh HwHiAiUser@192.168.1.199
**登录后会强制修改密码**
ssh HwHiAiUser@192.168.1.198
```

## 快速开始

SDK在以下环境中测试通过

- ubuntu 16.04, Atlas 800 服务器指定版本;

Atlas DDK 的ddk\_info信息 :

```
{
  "VERSION": "1.3.8.B902",
  "NAME": "DDK",
  "TARGET": "ASIC"
}
```

## 1. 安装软件

```
sudo apt-get install sshpass build-essential
```

## 2. 测试Demo

编译运行 :

下载后, 模型资源文件默认已经打包在开发者下载的SDK包中,

Step 0 : 使用HwHiAiUser登录

Step 1 : 运行一次install-demo.sh脚本, 会得到测试demo。

Step 2 : 请在官网获取序列号, 填写在demo\_async.cpp及demo\_sync.cpp的开始处license\_key字段。



图片加载失败

step3 : 准备测试图片

覆盖image目录下的 1.jpg, 更多图片可以用于demo中的批量测试模式

step4(可选) : 修改test\_300.sh下的以下开发板登录信息

```
export DDK_PATH=$HOME/tools/che/ddk/ddk # ddk的安装路径

declare -a DEVICE_IPS=("192.168.1.199") # 300加速卡芯片的ip地址, device=0 对应192.168.1.199
DEVICE_PASSWORD="Huawei@SYS3" # 之前 修改300加速卡SSH密码
MAIN_CPP="demo_async.cpp" # demo_async.cpp" 异步接口, "demo_async.cpp" 同步接口

OpenCV_install_dir=/home/HwHiAiUser/opencv_x64/ # OpenCV 3.4版本, 需要存在
${OpenCV_install_dir}/share/OpenCV/OpenCVConfig.cmake文件
```

step5: 运行demo, 会自动编译OpenCV 3.4库, 如果报错请自行编译, 目录设置在 OpenCV\_install\_dir

```
cd demo
sh test_300.sh
```

图像分类demo运行效果 :

```
[stat] [100001]image/1.jpg(4 images) time used: 41ms (at 1583765958531) total:705ms
[result][100001]image/1.jpg[281470472005664] is: n07747607 orange 0.973633 950;
```

n07747607 orange 分类名  
0.973633 分类概率  
950 分类名的序号

物体检测的demo运行效果：

```
[stat] time used : 101ms; all time used:478
images[3] result:
label:no2_ynen;prob:0.985352 loc:[(0.459961,0.839844), (0.5625,0.988281)]

no2_ynen 分类名 ， 也可以获取分类名的序号
0.985352 分类概率
loc:[(0.459961,0.839844), (0.5625,0.988281)]， 检测框的位置。(0.459961,0.839844表示左上角的点，(0.5625,0.988281)右下角的点；
如原始图片608， 左上角(0.459961*608,0.839844*608)， 右下角(0.5625*608,0.988281*608)
```

### SDK接口使用

使用该方式，将运行库嵌入到开发者的程序当中。

### 同步接口使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ;
auto predictor = global_controller()->CreateEdgePredictor(config);
int ret = predictor->init();
# 若返回非0，请查看输出日志排查错误原因。
auto img = cv::imread({图片路径});
// step 3: 预测图像
std::vector<EdgeResultData> result2;
predictor->infer(img, result2);
# 解析result2即可获取结果
```

### 异步接口使用流程

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 3: 创建Predictor ; 这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 4: 设置异步回调
predictor->set_result_handler(YOUR_HANDLER);

// step 5: 初始化
int ret = predictor->init();
**若返回非0, 请查看输出日志排查错误原因。**

// step 6: 预测图像
auto img = cv::imread({图片路径});
color_format = kBGR;
float threshold = 0.1;

uint64_t seq_id;
predictor->infer_async(img, color_format, 0.1, nullptr, seq_id);
**YOUR_HANDLER里面有seq_id的回调结果**

```

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

- 接口

```
virtual int set_licence_key(const std::string& license) = 0;
```

### 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

### 日志及报错

#### 日志

日志需要开启Atlas 的 INFO级别，/etc/slog.conf中配置关闭zip格式。清空/var/dlog 目录，运行atlas 300 [官方示例代码](#)，可以在/var/dlog目录下看见host和device开头的2个日志文件，中间是明文的info级别的日志

日志共有3处：

- host 测的easyedge.log。当前运行目录下。
- device侧的easyedge.device.xxx.log。 device侧的日志，在芯片的同名目录下。
- /var/dlog host 与device开头的log文件， ddk运行日志，其中device侧有略微延时

#### 通用错误码

错误码	常量	解释
1000004	RESOURCE_LOAD_FAILED	缺少data/model/conf.json文件或者该文件以及被改动。下载包中的data/model下的所有文件都不要改动，尝试使用默认配置。或者按照报错复制到对应目录。
7000001	AUTH_FAILED	服务端校验序列号失败
7000002	AUTH_LICENSE_INVALID	校验序列号
7000003	AUTH_LICENSE_EXPIRED	序列号过期
5000001	NET_CURL_PERFORM_FAILED	服务端校验序列号的请求因为网络原因失败
6000001	GET_MACHINE_ID_FAILED	没有相关权限，请反馈

**Atlas SDK 错误码**

错误码值	常量	含义	报错示例信息	示例解释及解决方式
12000011	FILE_NOT_READABLE	资源文件不可读	data/model/params IS NOT READABLE	data/model/params，这个文件不可读。SDK下载包中的data/model下的所有文件都不要改动，尝试使用默认配置。或者按照报错复制到对应目录。
12000012	HIAI_ERRORLIST_FILE	status.h.list不是原始文件	data/model/status.h.list IS TOO SMALL	下载包中的data/model下的所有文件都不要改动，包括status.h.list
12000102	PREDICTOR_NOT_INITED	create后没有调用init()函数	please call init() first	调用infer函数前没有调用init()
12000103	PREDICTOR_NO_HANDLER	create后没有调用set_result_handler()函数	please call set_result_handler() first	调用infer_async函数前没有调用set_result_handler(),建议init前调用
12000104	PREDICTOR_ALREADY_INITED	init()不管是否成功，不能连续调用。	don't call init() more than once	如果失败，请再次新建一个Predictor
12000105	BATCH_SIZE	AtlasConfig里的batch_size设置与model_name不符合	model batch size is 1; your config batch size is 4	batch_size设置里4，model_name设置里params，不对应导致报错。model_name应该设置为params-batch4
12000106	INPUT_WIDTH	preprocess_args.json被改动	model input tensor width is 224; your config resize is 226	请勿修改preprocess_args.json
12000107	INPUT_HEIGHT	同上	同上	同上
12000201	BATCH_TOO_MANY_IMAGES	一次输入的图片大于batch_size	too_many_images input:2; batch_size is 1	调用infer函数，输入了2张图片，大于batch数。如果batch=1的话，每次infer只能传一张图。
12000202	IMAGE_FORMAT_CHANNELS	infer函数输入的color_format与cv::Mat里的channel数不匹配	EdgeColorFormat is not according to cv channels; format is 101; channels is 3; seq_id1	101表示kRGBA，cv::Mat里channel应该期望是4。如果是直接读的图片，填kBGR。
12200001	ENGINE_MATRIX_COMMON	Atlas DDK Matrix部分（非CreateGraph函数）接口报错。即返回值HIAI_StatusT不是HIAI_OK。具体解释见Atlas官方文档。	hiai::Graph::ParseConfigFile(graph.prototxt); status Code is 16855066; HIAI ERROR CODE is 101 HIAI_GRAPH_PROTO_FILE_PARSE_FAILED_CODE,	调用hiai::Graph::ParseConfigFile()返回16855066，对应的status.h.list中的错误码是101。保留日志，具体见Atlas官方文档。
12200002	ENGINE_AI_COMMON	Atlas DDK Device引擎部分 hiai::AIStatus 不为hiai::SUCCESS	_ai_model_manager->Process()	保留日志，具体见Atlas官方文档。
12200003	ENGINE_MATRIX_INIT	Atlas DDK CreateGraph() 初始化DDK报错。具体解释见Atlas官方文档。	hiai::Graph::CreateGraph(); data/model/graph.prototxt; status Code is 16855190; HIAI ERROR CODE is 225 HIAI_FILE_NOT_EXIST_CODE,	示例为缺少libatlas_device.so导致
12200004	EDGEATLAS_ENGINE_MATRIX_INIT_DEVICE	Atlas DDK CreateGraph() 初始化DDK报错，这个报错很可能是device侧出现问题	hiai::Graph::CreateGraph(); data/model/graph.prototxt; status Code is 16855057; HIAI ERROR CODE is 92 HIAI_GRAPH_ENGINE_INIT_FAILED_CODE	需要具体排查DEVICE侧日志再次找具体报错，发现原因
12200005	ENGINE_ARGS_NULL	内部错误		请反馈
12300001	SYNC_INFER_TIMEOUT	调用infer同步接口时，内部会调用infer_async函数，这个函数超时	infer sync wait timeout more than 10ms	内部会调用infer_async函数超过10ms。1. 不要并发过高 2. 超时参数略微大些。

[🔗 图像分类服务器端SDK集成文档-EdgeKitProxy](#)

## 简介

本文档介绍EdgeKitProxy的使用方法。

**Release Notes** | 时间 | 版本 | 说明 | | ----- | ----- | ----- | | 2023-05-17 | 1.0.0 | 第一版 ! |

## 快速开始

### 二进制位置

位于SDK内bin目录中，文件名为edgekit\_serving，配套edgekit\_serving.yml为默认配置文件

### 注意事项

请参考各SDK文档中的注意事项

### 使用说明

### 服务启动

```
usage: edgekit_serving [<flags>]
```

#### Flags:

```
--help          显示帮助
-c, --cfg=./edgekit_serving.yml
                配置文件
-m, --model_dir=./RES    模型目录
-s, --serial_num=ABCD-EFGH-IJKL-MNOP
                序列号
--pool_min_size=1    预测池最小预测器个数
--pool_max_size=1    预测池最大预测器个数
--pool_full_interval_seconds=1
                    预测池满载多少秒进行扩容
--pool_idle_interval_seconds=1
                    预测池未满载多少秒进行缩容
--pool_available_device=1 ...
                    预测池可用设备列表
-d, --debug          开启debug模式
--log_to_std          日志输出至终端
--log_to_file          日志输出至文件
--log_file=easyedge.log 日志文件名
--log_max_size=10     日志最大大小 (MB)
--log_max_age=10      日志旧文件保留天数
--log_max_backups=100 日志旧文件保留个数
-h, --host=127.0.0.1  服务监听地址
-p, --port=24401      服务监听端口
--ws_max_handle_num=1 websocket接口最大处理请求个数
--ws_max_handle_timeout=30
                    websocket接口超时时间
```

### 配置文件说明



```
controller:
serialNum: AAAA AAAA AAAA AAAA # 序列号
modelDir: ../.././RES # 模型目录

predictorPool:
minSize: 1 # 预测池最小预测器个数
maxSize: 3 # 预测池最大预测器个数
fullIntervalSeconds: 1 # 预测池满载多少秒进行扩容
idleIntervalSeconds: 1 # 预测池未满载多少秒进行缩容
availableDevice: [-1] # 预测池可用设备列表

serving:
host: 0.0.0.0 # 服务监听地址
port: 24401 # 服务监听端口
enableHTTP: true # 对外开启HTTP服务
enableWS: false # 对外开启websocket服务
ws:
maxHandleNum: 1 # websocket接口最大处理请求个数
maxHandleTimeout: 30 # websocket接口超时时间

logging:
debug: true # 开启debug模式
logToStd: true # 日志输出至终端
logToFile: false # 日志输出至文件
logFile: easyedge.log # 日志文件名
maxSize: 10 # 日志最大大小 (MB)
maxAge: 10 # 日志旧文件保留天数
maxBackups: 100 # 日志旧文件保留个数
```

命令行参数会覆盖配置文件中同义配置

### 服务调用

HTTP服务接口url: \${监听地址}/ HTTP服务接口url: \${监听地址}/ws

### 请求参数

```

syntax = "proto3";

package easyedge.kit.proxy;

enum ImageType {
  Bin = 0; // 图片原始二进制内容，json格式下为base64编码后结果
  Mat = 1; // 图片Mat格式内容，json格式下为base64编码后结果
}

message HTTPRequest {
  bytes image = 1;
  ImageType image_type = 2;
  int32 height = 3;
  int32 width = 4;
  int32 channel = 5;
  float threshold = 6;
  int32 top_num = 7;
}

enum CommandType {
  GetInfo = 0;
  InferImage = 1;
}

enum InfoType {
  Hardware = 0;
}

message WebSocketRequest {
  string request_id = 1;
  CommandType command_type = 2;
  InfoType info_type = 3;
  bytes image = 4;
  ImageType image_type = 5;
  int32 height = 6;
  int32 width = 7;
  int32 channel = 8;
  int64 frame_id = 9;
  float threshold = 10;
  int32 top_num = 11;
}

```

## 返回参数

```

syntax = "proto3";

package easyedge.kit.proxy;

message BasicGPUInfo {
  string productName = 1;
  string memUsed = 2;
  string memTotal = 3;
  string gpuUtil = 4;
  string powerLimit = 5;
  string powerDraw = 6;
  string temperature = 7;
}

message DevStat {
  string name = 1;
  uint64 rx = 2;
  uint64 tx = 3;
}

message Chip {
  string name = 1;
  double powerUsed = 2;
  double powerLimit = 3;
  double temperature = 4;
}

```

```

double temperature = 4;
double chipUtil    = 5;
int64 memoryUsed  = 6;
int64 memoryTotal = 7;
}

message SMI {
  string name      = 1;
  string sdkVersion = 2;
  string driverVersion = 3;
  repeated Chip chips = 4;
}

message HInfo {
  string osName          = 1;
  string hostname        = 2;
  repeated string ipAddr = 3;
  repeated string macAddr = 4;
  uint64 bootTime       = 5;
  int32 cpuCores         = 6;
  double cpuMhz          = 7;
  string cpuModelName    = 8;
  double cpuUsage        = 9;
  map<string, double> cpuUsageDetail = 10;
  uint64 memTotal        = 11;
  uint64 memTotalUsed    = 12;
  double memUsage        = 13;
  map<string, double> memUsageDetail = 14;
  uint64 diskTotal       = 15;
  uint64 diskTotalUsed   = 16;
  double diskUsage       = 17;
  map<string, double> diskUsageDetail = 18;
  string userName        = 19;
  bool isInternetConnected = 20;
  string deviceId        = 21;
  int64 deviceTimestamp  = 22;
  map<string, DevStat> netUsageDetails = 23;
  repeated BasicGPUInfo gpuInfo = 24;
  double gpuUtil         = 25;
  uint64 gpuMemTotal     = 26;
  uint64 gpuMemTotalUsed = 27;
  double gpuMemUsage     = 28;
  map<string, SMI> aiChiplInfo = 29;
}

message LocationPoint {
  optional int32 x = 1;
  optional int32 y = 2;
}

message Location {
  optional int32 left = 1;
  optional int32 top = 2;
  optional int32 width = 3;
  optional int32 height = 4;
  repeated LocationPoint points = 5;
}

message Point {
  optional double x = 1;
  optional double y = 2;
}

message InferResultItem {
  optional int64 index = 1;
  optional double confidence = 2;
  optional double score = 3;
  optional string label = 4;
  optional string name = 5;
  optional int32 modelId = 6;
}

```

```

optional int32 modelKind = 6;

// 矩形检测
optional double x1 = 7;
optional double x2 = 8;
optional double y1 = 9;
optional double y2 = 10;
optional Location location = 11;

// 四边形检测
repeated Point points = 12;

// 追踪
optional int64 trackId = 13;
optional int64 frame = 14;
optional double fps = 15;

optional string mask = 16;
}

message HTTPResponse {
  int64 cost_ms = 1;
  int32 error_code = 2;
  int64 frame_id = 3;
  repeated InferResultItem results = 4;
}

message WebSocketInferResponse {
  string request_id = 1;
  int64 cost_ms = 2;
  int32 error_code = 3;
  int64 frame_id = 4;
  repeated InferResultItem results = 5;
  bytes annotated = 6; // 渲染后的图片原始二进制内容，json格式下为base64编码后结果，目前语义分割返回这个类型
}

message WebSocketHInfoResponse {
  string request_id = 1;
  int32 status = 2;
  string msg = 3;
  HInfo data = 4;
}

```

## 其他说明

### 单机负载均衡

通过配置文件或命令行参数配置了预测池相关配置后，若预测池最小与最大预测器个数不同，且扩缩容配置不为-1则开启单机负载均衡，服务启动时会创建最小数量的预测器，后续根据实际请求情况，若所有预测器均有负载的持续时间大于配置中的满载扩容时间，且预测器数量未到达最大个数时，会自动扩容，后续若请求并发数下降，预测器池中预测器不能跑满负载时，则会自动缩容，尽可能最大化利用单机资源

### 纯离线API集成说明

本文档主要说明定制化图像分类模型发布为本地服务器API（通过API部署包实现）后如何使用。如还未训练模型，请先前往[EasyDL](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### 部署包使用说明 部署方法

EasyDL定制化图像分类模型的服务器API通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#)使用python2命令来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。[运维检查](#)

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络联通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

### 授权说明

部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

### 性能指标

图像分类模型可部署在CPU或GPU服务器上，单实例具体性能指标参见[算法性能及适配硬件 API参考](#)

### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请部署包，部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/ImageClassification](http://{IP}:{PORT}/{DEPLOY_NAME}/ImageClassification) IP：服务部署所在机器的ip地址 PORT：服务部署后获取的端口

DEPLOY\_NAME：申请时填写的服务名称

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```

{
  "image": "<base64数据>",
  "top_num": 5
}

```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
top_num	否	number	-	返回分类数量，默认为6个

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如缺少必要出入参时返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
337000	Auth check failed	离线鉴权调用失败

### 模型更新/回滚操作说明

#### 模型更新

1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。

两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如`easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}` :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxvf xx.tar.gz
**解压后,进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh

**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}

**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务: ${DEPLOY_NAME},前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务: ${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}

** (可选操作) 更新证书**
python2 install.py lu

```

### 模型回滚

以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}

**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}

**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh

** (可选操作) 进入V1版本部署包所在目录执行license更新操作,假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录,参考上述【模型更新】步骤,执行模型升级操作(即先卸载v2,后升级为v1)

### 通用小型设备部署

#### 如何在通用小型设备部署

训练完毕后,可以选择将模型通过「SDK-纯离线服务」或「API-端云协同服务」部署,具体介绍如下:

#### 纯离线服务部署

纯离线服务目前仅支持通过SDK集成,可以在左侧导航栏中找到「发布模型」,依次进行以下操作即可发布设备端SDK:

- 选择模型
- 选择部署方式「EasyEdge本地部署」-「通用小型设备」
- 选择版本
- 选择集成方式
- 点击发布

发布模型

选择模型

部署方式

选择版本

集成方式  SDK-纯离线服务

说明:

1. 设备端SDK支持Android、iOS、Windows、Linux操作系统，具体的系统、硬件环境支持请参考[技术文档](#)。提供可直接体验的移动端app安装包，以及相应代码包、说明文档，供企业用户/开发者二次开发
2. 如SDK生成失败，或有任何其他问题，欢迎[提交工单](#)或加入QQ群(679517246) 咨询了解

发布

- 再根据实际使用设备选择系统与芯片
- 点击发布

纯离线服务 > 发布新服务

部署方式  服务器  通用小型设备  专项适配硬件

选择模型

选择版本

选择系统和芯片

Linux

Windows

通用X86 CPU

Android

iOS

模型加速:  同时获取加速版 [?](#)

发布

也可以直接在「EasyEdge本地部署」-「纯离线服务」页面点击发布新服务，按上图所述进行申请发布

### 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供，基于[百度智能边缘](#)构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

[纯离线SDK说明](#)

[纯离线SDK简介](#)

本文档主要说明定制化模型发布后获得的SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)



- 前往[官方论坛](#)交流，与其他开发者进行互动

## SDK说明

SDK支持iOS、Android、Linux、Windows四种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
iOS	iOS 8.0 以上 (A仿生芯片版要求11.0以上)	ARMv7 ARM64 (Standard architectures) (暂不支持模拟器)
Android	通用ARM: Android 19以上 SNPE : Android 21以上 DDK : Android 21以上	通用ARM: 绝大部分的手机和平板、比较耗时 SNPE : 高通Soc, 仅支持Qualcomm Snapdragon 450 之后发布的soc。其中 660 之后的型号可能含有 Hexagon DSP模块, 具体列表见snpe 高通骁龙引擎 DDK : CPU支持华为麒麟970N、980的arm-v8a的soc, 支持的机型 mate10, mate10pro, P20, mate20等  支持armeabi-v7a arm-v8a CPU 架构, DDK仅支持 arm-v8a
Linux C++		CPU: AArch64 ARMv7l ASIC: Hisilicon NNIE1.1 on AArch64 (Hi3559AV100/Hi3559CV100等) ASIC: Hisilicon NNIE1.2 on ARMv7l (Hi3519AV100/Hi3559V200等)
Linux Python		Intel Movidius Myriad2/Myriad X
Linux Ubuntu 16.04		AArch64 HUAWEI Atlas 200
Windows	64位 Windows7 及以上	Intel CPU x86_64 Intel Movidius Myriad2/Myriad X (仅支持Win10)  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015

## 说明

针对iOS操作系统：虽然SDK库文件很大（ipa文件很大），但最终应用在用户设备中所占用的大小会缩小很多，如图像分类下载的ipa文件可能会100M以上，但实际集成在设备中只有20M左右。这与multi architectures、bitcode和AppStore的优化有关。

## 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)页面查看评测信息表

## 自适应芯片版SDK

发布SDK时可根据实际应用时的硬件/芯片配置选择最合适的SDK。如“华为NPU版”就是针对华为NPU芯片做了适配与加速的SDK。如实际应用时需要适配多种芯片，就可以发布“自适应芯片版”SDK，SDK被集成后会自动判断设备的芯片并运行相应的模型。

## 加速版SDK

发布SDK时，勾选「同时获取加速版」，就可以同时获得适配部分芯片（需选中且右侧带有加速标记）的基础版SDK和加速版SDK。



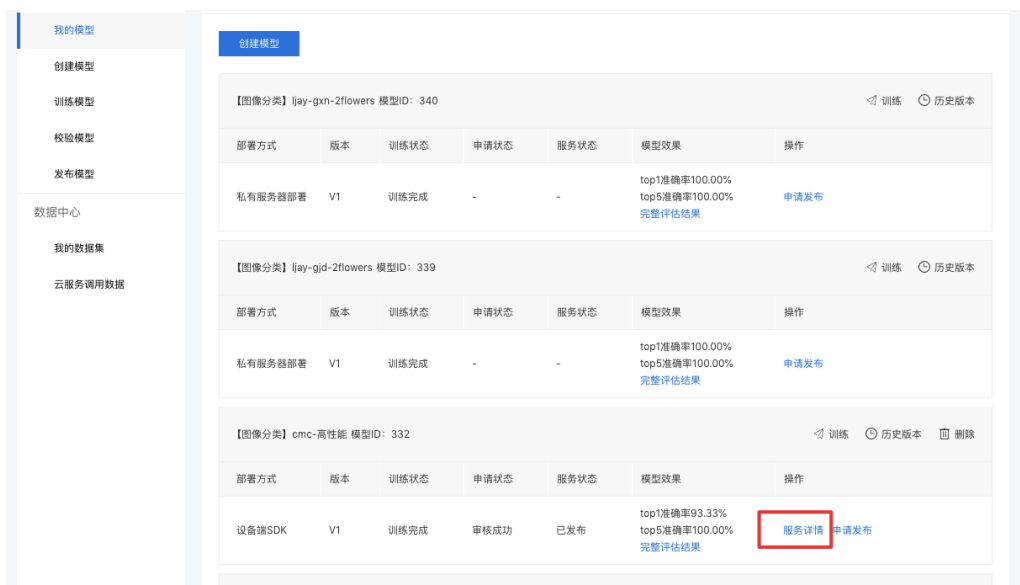
目前加速版SDK已支持Windows X86、Linux ARM、iOS ARM、Android ARM环境，加速后的SDK普遍在包大小、内存占用、识别速度等方面表现更优，详细对比请见[算法性能及适配硬件](#)。

加速版SDK和基础版的测试方式类似，只需在EasyDL控制台新增「加速版」测试序列号，即可获得3个月的测试期。

### 激活&使用SDK

SDK的激活与使用分以下四步：

#### ① 在【我的模型】-【服务详情】内下载SDK



### 设备端SDK下载

此处下载的SDK为未授权SDK，需要获取序列号激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发

操作系统	操作
iOS	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Android	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Linux	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Windows	<a href="#">下载SDK</a> <a href="#">获取序列号</a>

#### ② 在控制台获取序列号

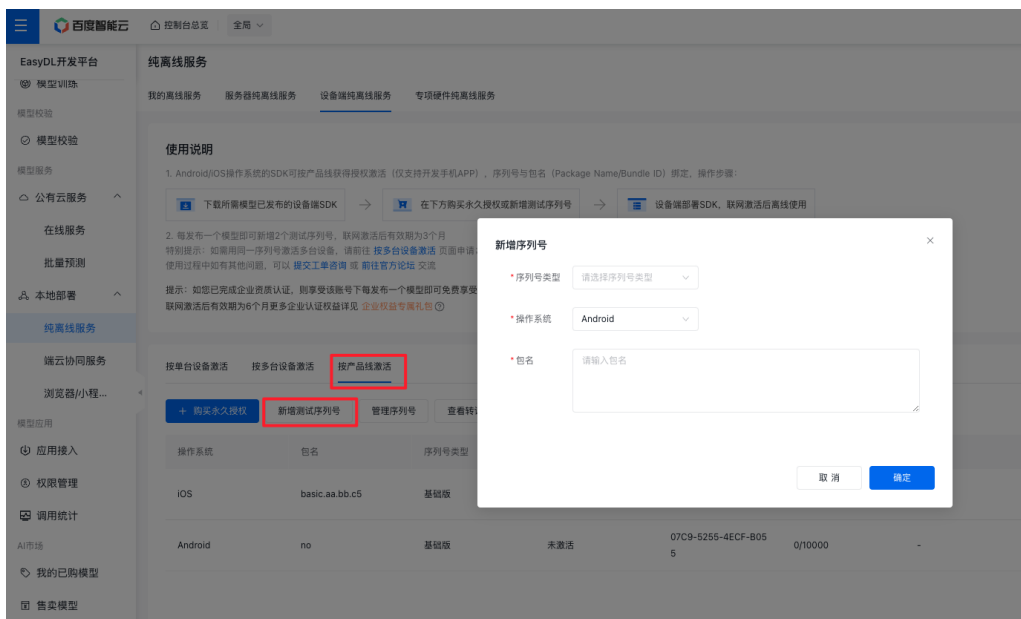
按单台或多台设备获得授权并使用SDK时：



包括两种激活方式：

1. 按设备激活：是按单台设备数授权（依据硬件指纹信息），每个序列号可支持多台设备，激活后授权永久有效，但如果硬件信息发生变更（如拆卸网卡或后续增加网卡），或者硬件损坏，则授权不可恢复
2. 按实例数激活：是按单台实例数授权，每隔一次“定期确认时间”，后台会确认当前设备是否在线，如果离线，当前设备解绑，新设备可以接入，该方式仅支持联网激活，设备需要保证联网，定期确认状态。

Android或iOS操作系统的SDK可以选择按产品线激活（仅支持开发手机APP），序列号与包名（Package Name/Bundle ID）绑定：



### ③ 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

### ④ 正式使用

#### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在百度智能云控制台内[提交工单](#)反馈

#### 1、激活失败怎么办？

按设备激活时，激活失败可能由于以下几个原因造成：

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活

- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在百度智能云控制台内[提交工单](#)反馈

按产品线激活时，激活失败可能由于以下几个原因造成：

- ①可能是包名填写错误，请核对与序列号绑定的包名是否与实际包名一致
- ②序列号填写错误，请核实序列号后重新激活
- ③首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ④模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑤序列号已过有效期，请申请延期后重试
- ⑥如有其他异常请在百度智能云控制台内[提交工单](#)反馈

## 2、怎样申请序列号使用延期

序列号激活后有效期为三个月，可以在[控制台](#)进行申请，申请流程：

### 1) 填写申请信息

The screenshot shows the 'EasyDL定制训练平台' (EasyDL Custom Training Platform) interface. The breadcrumb path is '产品服务 / EasyDL定制训练平台 - 离线SDK管理 / 申请延期'. The main content area is titled '申请延期' (Apply for Extension). It contains the following fields:

- \* 选择模型: 请选择 (dropdown menu)
- \* 选择版本: 请选择 (dropdown menu)
- \* 申请到期时间: 2019-03-31 (calendar icon)
- \* 申请延期理由: 请填写少于500字的申请延期理由 (text area)

A blue button labeled '提交审核' (Submit for Review) is located at the bottom of the form.

2) 等待审核：审核周期通常需要1-3个工作日左右，期间会有工作人员电话回访，请填写有效的联系方式并保证手机畅通

## Android集成文档

### 简介

**1.1 Android SDK 硬件要求** Android 版本：支持 Android 5.0 (API 21) 及以上

硬件：支持 arm64-v8a 和 armeabi-v7a，暂不支持模拟器

通常您下载的SDK只支持固定的某一类芯片。

- **通用ARM**：支持大部分ARM 架构的手机、平板及开发板。**通常选择这个引擎进行推理。**
- **通用ARM GPU**：支持骁龙、麒麟、联发科等带GPU的手机、平板及开发板。

- 高端芯片AI加速模块：
  - **高通骁龙引擎SNPE**：高通骁龙高端SOC，利用自带的DSP加速。其中 660 之后的型号可能含有 Hexagon DSP模块，具体列表见snpe高通骁龙引擎官网。
  - **华为NPU引擎DDK**：华为麒麟980的arm-v8a的soc。具体手机机型为mate10，mate10pro，P20，mate20，荣耀v20等。
  - **华为达芬奇NPU引擎DAVINCI**：华为NPU的后续版本，华为麒麟810，820，990，985的arm-v8a的soc。具体手机机型为华为mate30，p40，nova6，荣耀v30等。

通用ARM有额外的加速版，但是有一定的精度损失。

因GPU硬件限制，通用ARM GPU物体检测模型输入尺寸较大时会运行失败，可以在训练的时候将输入尺寸设为300\*300。

高端芯片AI加速模块，一般情况下推理速度较快。

运行内存不能过小，一般大于demo的assets目录大小的3倍。

### 1.2 功能支持 | 引擎 | 图像分类 | 物体检测 | 图像分割 | 文字识别

只支持EasyEdge | 姿态估计 | | :: | :: | :: | :: | :: | :: | 通用ARM | √ | √ | √ | √ | √ | 通用ARM GPU | √ | √ | √ | √ | 高通骁龙引擎SNPE | √ | √ | || | 华为NPU引擎DDK | √ | √ | || | 华为达芬奇NPU引擎DAVINCI | √ | √ | √ | ||

### 1.3 Release Notes

时间	版本	说明
2024.12.13	0.10.14	优化不再读取硬件信息，去除硬件信息获取开关
2024.12.10	0.10.13	新增硬件信息获取开关；初始化方式优化
2023.08.31	0.10.12	新增支持实例数鉴权；SNPE引擎升级；迭代优化
2023.06.29	0.10.11	迭代优化
2023.05.17	0.10.10	横屏兼容；迭代优化
2023.03.16	0.10.9	达芬奇NPU支持更多模型及语义分割模型；各芯片支持更多语义分割模型；精简版代码补充；迭代优化
2022.12.29	0.10.8	ARM / ARM-GPU 引擎升级；迭代优化
2022.10.27	0.10.7	达芬奇NPU新增适配麒麟985；迭代优化
2022.09.15	0.10.6	SNPE引擎升级；迭代优化
2022.07.28	0.10.5	迭代优化
2022.06.30	0.10.4	支持Android11；支持EasyEdge语义分割模型；迭代优化
2022.05.18	0.10.3	ARM / ARM-GPU 引擎升级；支持更多加速版模型发布；迭代优化
2022.03.25	0.10.2	ARM / ARM-GPU 引擎升级；支持更多检测模型；迭代优化
2021.12.22	0.10.1	DDK不再支持Kirin 970；迭代优化
2021.10.20	0.10.0	更新鉴权；更新达芬奇NPU、SNPE、通用ARM及ARM-GPU引擎；新增达芬奇NPU对检测模型的支持；支持更多姿态估计模型
2021.07.29	0.9.17	迭代优化
2021.06.29	0.9.16	迭代优化
2021.05.13	0.9.15	更新鉴权，更新通用arm及通用arm gpu引擎
2021.04.02	0.9.14	修正bug
2021.03.09	0.9.13	更新android arm的预处理加速
2020.12.18	0.9.12	通用ARM引擎升级；新增ARM GPU引擎
2020.10.29	0.9.10	迭代优化
2020.9.01	0.9.9	迭代优化
2020.8.11	0.9.8	更新ddk 达芬奇引擎
2020.7.14	0.9.7	支持arm版ocr模型，模型加载优化
2020.6.23	0.9.6	支持arm版fasterrcnn模型
2020.5.14	0.9.5	新增华为新的达芬奇架构npu的部分图像分类模型
2020.4.17	0.9.4	新增arm通用引擎量化模型支持
2020.1.17	0.9.3	新增arm通用引擎图像分割模型支持
2019.12.26	0.9.2	新增华为kirin麒麟芯片的物体检测支持
2019.12.04	0.9.1	使用paddleLite作为arm预测引擎
2019.08.30	0.9.0	支持EasyDL专业版
2019.08.30	0.8.2	支持华为麒麟980的物体检测模型
2019.08.29	0.8.1	修复相机在开发版调用奔溃的问题
2019.06.20	0.8.0	高通手机引擎优化
2019.05.24	0.7.0	升级引擎
2019.05.14	0.6.0	优化demo程序
2019.04.12	0.5.0	新增华为麒麟980支持
2019.03.29	0.4.0	引擎优化，支持sd卡模型读取
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

快速开始

## 2.1 安装软件及硬件准备

扫描模型下载SDK处的网页上的二维码，无需任何依赖，直接体验

如果需要源码方式测试：

打开AndroidStudio，点击 "Import Project..."。在一台较新的手机上测试。

详细步骤如下：

1. 准备一台较新的手机，如果不是通用arm版本，请参见本文的“硬件要求”，确认是否符合SDK的要求
2. 安装较新版本的AndroidStudio，[下载地址](#)
3. 新建一个HelloWorld项目，Android Studio会自动下载依赖，在这台较新的手机上测试通过这个helloworld项目。注意不支持模拟器。
4. 解压下载的SDK。
5. 打开AndroidStudio，点击 "Import Project..."。即：File->New-> "Import Project..."，选择解压后的目录。
6. 此时点击运行按钮（同第3步），手机上会有新app安装完毕，运行效果和二维码扫描的一样。
7. 手机上UI界面显示后，如果点击UI界面上的“开始使用”按钮，可能会报序列号错误。请参见下文修改

## 2.2 使用序列号激活

如果使用的是EasyEdge的开源模型，无需序列号，可以跳过本段直接测试。

建议申请包名为"com.baidu.ai.easyaimobile.demo"的序列号用于测试。

本文假设已经获取到序列号，并且这个序列号已经绑定包名。

SDK默认使用离线激活方式，即首次联网激活，后续离线使用。SDK同时支持按实例数鉴权方式，即周期性联网激活，离线后会释放所占设备实例。按实例数鉴权的启用参考本节2.2.3说明

**2.2.1 填写序列号** 打开Android Studio的项目，修改MainActivity类的开头SERIAL\_NUM字段。 MainActivity 位于 app\src\main\java\com\baidu\ai\edge\demo\MainActivity.java文件内。

```
// 请替换为您的序列号
private static final String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX"; //这里填您的序列号
```

### 2.2.2 修改包名

如果申请的包名为"com.baidu.ai.easyaimobile.demo"，这个是demo的包名，可以不用修改

打开app/build.gradle文件，修改"com.baidu.ai.easyaimobile.demo"为申请的包名

```
defaultConfig {
    applicationId "com.baidu.ai.easyaimobile.demo" // 修改为比如"com.xxx.xxx"
}
```

修改序列号和包名后，可以运行测试，效果同扫描二维码的一致

**2.2.3 按实例数鉴权** 设置好序列号和包名后，调用配置类的以下方法启用并配置心跳间隔时间：

```
XXXConfig config = new XXXConfig();
// 启用按实例数鉴权，配置心跳间隔，单位：秒
config.setInstanceAuthMode(10000);
```

配置类的详细说明参考后续章节【调用流程】

## 2.3 测试精简版

对于通用ARM、高通骁龙引擎SNPE、华为NPU引擎DDK和达芬奇NPU引擎Davinci的常见功能，项目内自带精简版，可以忽略开发板不兼容的摄像头。

此外，由于实时摄像开启，会导致接口的耗时变大，此时也可以使用精简版测试。

目前以下硬件环境有精简版测试：

- 通用ARM：图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation)，姿态估计 (Pose)
- 通用ARM GPU：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)，姿态估计 (Pose)
- 高通骁龙引擎SNPE：图像分类 (Classify)，物体检测 (Detection)
- 华为NPU引擎DDK：图像分类 (Classify)，物体检测 (Detection)
- 华为达芬奇NPU引擎Davinci：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)

具体代码分别在infertest、snpetest、ddktest和davincitest目录下。

修改方法为（以通用ARM为例）：更改app/main/AndroidManifest.xml中的启动Activity。

```
<activity android:name=".infertest.MainActivity"> <!-- 原始的是".MainActivity" -->
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

开启后会自动选择图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation) 或姿态估计 (Pose) 测试。



Demo APP 检测模型运行示例

精简版检测模型运行示例



```

Hello World!
ARM Detection
Start running: 0
Predict 0: (size:100, firstRe
confidence:0.6314938, bo
181)}}
Finish running
Task finished

```

### 识别结果

置信度  0.30

序号	名称	置信度
1	person	0.63
2	person	0.47
3	car	0.42
4	horse	0.40
5	dog	0.34
6	truck	0.34

BU

#### 使用说明

##### 3.1 代码目录结构

集成时需要“复制到自己的项目里”的目录或者文件：

1. app/libs

## 2. app/src/main/assets/xxxx-xxxx 如app/src/main/assets/infer

```

+app 简单的设置，模拟用户的项目
|---+libs 实际使用时需要复制到自己的项目里
|   |---arm64-v8a v8a的so
|   |---armeabi-v7a v7a的so
|   |---easyedge-sdk.jar jar库文件
|---+src/main
|   |---+assets
|       |---demo demo项目的配置，实际集成不需要
|       |---infer 也可能是其它命名，infer表示通用arm。实际使用时可以复制到自己的项目里
|---+java/com.baidu.ai.edge/demo
|   |---+infertest 通用Arm精简版测试，里面有SDK的集成逻辑
|       |--- MainActivity 通用Arm精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里
|       |   面的序列号
|           |--- TestInferArmClassifyTask 通用Arm精简版分类
|           |--- TestInferArmDetectionTask 通用Arm精简版检测
|           |--- TestInferArmOcrTask 通用Arm精简版OCR
|           |--- TestInferArmPoseTask 通用Arm精简版姿态
|           |--- TestInferArmSegmentTask 通用Arm精简版分割
|           |--- TestInferArmGpuClassifyTask 通用ArmGpu精简版分类
|           |--- TestInferArmGpuDetectionTask 通用ArmGpu精简版检测
|           |--- TestInferArmGpuPoseTask 通用ArmGpu精简版姿态
|           |--- TestInferArmGpuSegmentTask 通用ArmGpu精简版分割
|   |---+snpetest SNPE精简版测试
|       |--- MainActivity SNPE精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
|       |   的序列号
|           |--- TestSnpeDspClassifyTask SNPE DSP精简版分类
|           |--- TestSnpeDspDetectionTask SNPE DSP精简版检测
|           |--- TestSnpeGpuClassifyTask SNPE Gpu精简版分类
|           |--- TestSnpeGpuDetectionTask SNPE Gpu精简版检测
|   |---+ddktest DDK精简版测试
|       |--- MainActivity DDK精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面的
|       |   序列号
|           |--- TestDDKClassifyTask DDK精简版分类
|           |--- TestDDKDetectionTask DDK精简版检测
|   |---+davincitest Davinci精简版测试
|       |--- MainActivity Davinci精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
|       |   的序列号
|           |--- TestDavinciClassifyTask Davinci精简版分类
|           |--- TestDavinciDetectionTask Davinci精简版检测
|           |--- TestDavinciSegmentTask Davinci精简版分割
|       |--- CameraActivity 摄像头扫描示例，里面有SDK的集成逻辑
|       |--- MainActivity 启动Activity，使用时需要修改里面的序列号
|--- build.gradle 这里修改包名
+camera_ui UI模块，集成时可以忽略

```

## 3.2 调用流程 以通用ARM的检测模型功能为例，

代码可以参考TestInferDetectionTask

1. 准备配置类，如InferConfig，输入：通常为一个assets目录下的文件夹，如infer。
2. 初始化Manager，比如InferManager。输入：第1步的配置类和序列号
3. 推理图片，可以多次调用 5.1 准备图片，作为Bitmap输入 6.2 调用对应的推理方法，比如detect 7.3 解析结果，结果通常是一个List，调用结果类的Get方法，通常能获取想要的结果
4. 直到长时间不再使用我们的SDK，调用Manger的destroy方法释放资源。

## 3.3 具体接口说明 下文的示例部分以通用ARM的检测模型功能为例

即接口为InferConfig， InferManager， InferManager.detect。

其它引擎和模型调用方法类似。

下文假设已有序列号及对应的包名

### 3.3.1. 准备配置类

- INFER : 通用ARM , InferConfig
- ARM GPU : ArmGpuConfig
- SNPE : 高通骁龙DSP , SnpeConfig
- SNPE GPU : 高通骁龙GPU , SnpeGpuConfig
- DDK : 华为NPU , DDKConfig
- DDKDAVINCI : 华为达芬奇NPU , DDKDaVinciConfig

```
InferConfig mInferConfig = new InferConfig(getAssets(),
    "infer");
// assets 目录下的infer, infer表示通用arm
```

输入 : assets下的配置  
输出 : 具体的配置类

### 3.3.2. 初始化Manager类

- INFER : 通用ARM , InferManager
- ARM GPU : 通用ARM GPU , InferManager
- SNPE : 高通骁龙DSP , SnpeManager
- SNPE GPU : 高通骁龙GPU , SnpeManager
- DDK : 华为NPU , DDKManager
- DDKDAVINCI : 华为达芬奇NPU , DavinciManager

```
String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX";

// InferManager 为例:
InferManager manager = new InferManager(this, config, SERIAL_NUM); // config为上一步的InferConfig
```

#### 注意要点

1. 同一个时刻只能有唯一有效的InferManager。旧的InferManager必须调用destory后, 才能新建一个new InferManager()。
2. InferManager的任何方法, 都不能在UI线程中调用。
3. new InferManager() 及InferManager成员方法由于线程同步数据可见性问题, 都必须在一个线程中执行。如使用android自带的ThreadHandler类。

输入 : 1.配置类; 2.序列号  
输出 : Manager类

### 3.3.3. 推理图片

- 接口可以多次调用, 但是必须在一个线程里, 不能并发
- confidence, 置信度[0-1], 小于confidence的结果不返回。填confidence=0, 返回所有结果
- confidence可以不填, 默认用模型推荐的。

准备图片, 作为Bitmap输入,

- 输入为Bitmap, 其中Bitmap的options为默认。如果强制指定的话, 必须使用*Bitmap.Config.ARGB\_8888*

调用对应的推理方法及结果解析 见下文的各个模型方法

### 3.3.4 分类Classify

```
public interface ClassifyInterface {
    List<ClassificationResultModel> classify(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 ClassifyInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 ClassificationResultModel  
异常：一般首次出现。可以打印出异常错误码。

ClassificationResultModel

- label：分类标签，定义在label\_list.txt中
- confidence：置信度，0-1
- labelIndex：标签对应的序号

### 3.3.5 检测Detect

对于EasyDL口罩检测模型请注意输入图片中人脸大小建议保持在88到9696像素，可根据场景远近程度缩放图片后传入

```
public interface DetectInterface {
    List<DetectionResultModel> detect(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 DetectInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 DetectionResultModel List  
异常：一般首次出现。可以打印出异常错误码。

DetectionResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- bounds：Rect，左上角和右下角坐标

### 3.3.6 图像分割Segmentation

```
public interface SegmentInterface {
    List<SegmentationResultModel> segment(Bitmap bitmap, float confidence) throws BaseException;
    // 如InferManger 继承 SegmentInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 SegmentationResultModel  
异常：一般首次出现。可以打印出异常错误码。

SegmentationResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- labelIndex：标签对应的序号
- box: Rect对象表示的对象框
- mask：byte[]表示的原图大小的0，1掩码，绘制1的像素即可得到当前对象区域

mask 字段说明，如何绘制掩码也可参考demo工程

```
1 0 1
image 1 1 0  => mask(byte[]) 101 110 011
0 1 1
```

### 3.3.7 文字识别OCR

暂时只支持通用ARM引擎，不支持其它引擎，暂时只支持EasyEdge的开源OCR模型。

```
public interface OcrInterface {
    List<OcrResultModel> ocr(Bitmap bitmap, float confidence) throws BaseException;
    // 如InferManger 继承 OcrInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 OcrResultModel List，每个OcrResultModel对应结果里的一个四边形。  
 异常：一般首次出现。可以打印出异常错误码。

OcrResultModel

- label：识别出的文字
- confidence：置信度
- List<Point>：4个点构成四边形

### 3.3.8 姿态估计Pose

暂时只支持通用ARM引擎，不支持其它引擎

```
public interface PoseInterface {
    List<PoseResultModel> pose(Bitmap bitmap) throws BaseException;
    // 如InferManger 继承 PoseInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 PoseResultModel List  
 异常：一般首次出现。可以打印出异常错误码。

PoseResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- Pair<Point, Point>：2个点构成一条线

### 3.3.9 释放

释放后这个对象不能再使用，如果需要使用可以重新new一个出来。

```
public void destory() throws BaseException
```

### 3.3.10 整体示例

以通用ARM的图像分类预测流程为例：

```
try {
    // step 1: 准备配置类
    InferConfig config = new InferConfig(context.getAssets(), "infer");

    // step 2: 初始化预测 Manager
    InferManager manager = new InferManager(context, config, "");

    // step 3: 准备待预测的图像，必须为 Bitmap.Config.ARGB_8888 格式，一般为默认格式
    Bitmap image = getFromSomeWhere();

    // step 4: 预测图像
    List<ClassificationResultModel> results = manager.classify(image, 0.3f);

    // step 5: 解析结果
    for (ClassificationResultModel resultModel : results) {
        Log.i(TAG, "labelIndex=" + resultModel.getLabelIndex()
            + ", labelName=" + resultModel.getLabel()
            + ", confidence=" + resultModel.getConfidence());
    }

    // step 6: 释放资源。预测完毕请及时释放资源
    manager.destory();
} catch (Exception e) {
    Log.e(TAG, e.getMessage());
}
```

### 3.3.11 高通骁龙引擎的额外配置

```
"autocheck_qcom": true, // 如果改成false, sdk跳过检查手机是否是高通的Soc, 非高通的Soc会奔溃直接导致app闪退
```

```
"snpe_runtimes_order": [],
// 不填写为自动, 按照 {DSP, GPU, GPU_FLOAT16, CPU}次序尝试初始化, 也可以手动指定如[2,1,3,0], 具体数字的定义见下段
```

```
public interface SnpeRuntimeInterface {
    int CPU = 0;
    int GPU = 1;
    int DSP = 2;
    int GPU_FLOAT16 = 3;
}
```

```
// SnpeManager 中, 使用public static ArrayList<Integer> getAvailableRuntimes(Context context) 方法可以获取高通SOC支持的运行方式
```

## 集成指南

1. 复制库文件libs
2. 添加Manifest权限
3. 复制模型文件
4. 添加调用代码(见上一步具体接口说明)

### 4.1 复制库文件libs A. 如果项目里没有自己的jar文件和so文件:

复制app/libs 至自己项目的app/libs目录。  
参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

### B. 如果项目里有自己的jar文件, 但没有so文件

easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a放到app/src/main/jniLibs目录下

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
}
```

### C. 如果项目里有自己的jar文件和so文件

easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a取交集和自己的so放在一起，交集的意思是比如自己的项目里有x86目录，必须删除x86。

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a' // abiFilter取交集，即只能少不能多
        }
    }
}
```

jar文件库如果没有设置成功的，编译的时候可以发现报错。

so库如果没有编译进去的话，也可以通过解压apk文件确认。运行的时候会有类似jni方法找不到的报错。

#### 4.2 Manifest配置

参考app/src/main/AndroidManifest.xml文件，添加：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- Android 11 支持 -->
<uses-permission
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />

<!-- 高版本 Android 支持 -->
<application
    android:requestLegacyExternalStorage="true"
    android:usesCleartextTraffic="true">
</application>
```

#### 4.3 混淆规则（可选） 请不要混淆SDK里的jar文件。

```
-keep class com.baidu.ai.edge.core.*.*{ *; }
```

#### 4.4 Android 11支持 除Manifest中必要配置外，请参考BaseActivity获取所有文件访问权限，否则可能影响SDK正常使用。

SDK 默认使用 easyedge-sdk.jar，未启用 AndroidX，若您的项目使用 AndroidX，并在集成中提示 android.support 相关错误，请参考 app/build.gradle 使用 etc/easyedge-sdk-androidx.jar 以支持 AndroidX：

```
// app/build.gradle

dependencies {
    implementation project(':camera_ui')
    implementation files('libs/easyedge-sdk-androidx.jar') // 修改 jar 包依赖
}
```

**错误码** | 错误码 | 错误描述 | 详细描述及解决方法 | |---|---|---| | 1001 | assets 目录下用户指定的配置文件不存在或不正确 | SDK使用assets目录下一系列文件作为配置文件。如果文件缺失或内容不正确，则有此报错 | | 1002 | json格式的配置文件解析出错 | 如缺少某些字段。正常情况下，配置文件请不要修改 | | 1003 | 应用缺少权限 | 请根据提示动态申请缺少的权限 | | 19xx | Sdk内部错误 | 请与百度人员联系 | | 2001 | XxxMANAGER 只允许一个实例 | 如已有XxxMANAGER对象，请调用destory方法 | | 2002 | XxxMANAGER 已经调用过destory方法 | 在一个已经调用destory方法的DETECT\_MANAGER对象上，不允许再调用任何方法 | | 2003 | 传入的assets下模型文件路径非法 | 比如缺少模型文件，XxxConfig.getModelFileAssetPath() 返回为null | | 2012 | JNI内存错误 | heap的内存不够 | | 2103 | license过期 | license失效或者系统时间有异常 | | 2601/2602 | assets 目录下模型文件打开/读取失败 | 请根据报错信息检查模型文件是否存在 | | 27xx | Sdk内部错误 | 请与百度人员联系 | | 28xx | 引擎内部错误 | 请与百度人员联系 | | 29xx | Sdk内部错误 | 请与百度人员联系 | | 3000 | so加载错误 | 请确认所有so文件存在于apk中 | | 3001 | 模型加载错误 | 请确认模型放置于能被加载到的合法路径中，并确保配置文件正确 | | 3002 | 模型卸载错误 | 请与百度人员联系 | | 3003 | 调用模型错误 | 在模型未加载正确或者so库未加载正确的情况下调用了分类接口 | | 31xx | SDK激活失败 | 请与百度人员联系 | | 4011 | SDK类型与设备硬件不匹配 | 比如适配DSP的SDK运行在麒麟芯片上会出现此报错，请在部署包支持的硬件上使用SDK | | 50xx | SDK调用异常 | 请与百度



人员联系 |

**报错日志收集** 通常 Logcat 可以看见日志及崩溃信息，若设备无法获取日志信息，可使用 Demo 中的 xCrash 工具：

```
// 1. 引入 app/build.gradle 的 xCrash 依赖
android {
    ...
    dependencies {
        implementation 'com.iqiyi.xcrash:xcrash-android-lib:2.4.5' // 可以保存崩溃信息，默认未引入
        ...
    }
}
// 2. 启用日志收集。日志将保存在 /sdcard/<包名>/xCrash
// app/src/main/java/com.baidu.ai.edge/demo/MyApplication.java
protected void attachBaseContext(Context context) {
    // 日志保存位置
    String basePath = Environment.getExternalStorageDirectory().toString() + "/" + context.getPackageName();
    // 启用
    XCrash.InitParameters params = new XCrash.InitParameters();
    params.setAppVersion(BaseManager.VERSION);
    params.setLogDir(basePath + "/xCrash");
    XCrash.init(this, params);
}
```

## 🔗 iOS集成文档

### 简介

本文档描述 EasyEdge/EasyDL iOS 离线预测SDK相关功能；

目前支持EasyEdge的功能包括：

- 图像分类
- 物体检测
- 人脸检测
- 姿态估计
- 百度OCR模型

目前支持EasyDL的功能包括：

- 图像分类
- 物体检测
- 图像分割

### 系统支持

系统：

- 通用arm版本：iOS 9.0 以上
- A仿生芯片版：iOS 15.0 及以上

硬件：arm64 (Standard architectures)（暂不支持模拟器）

内存：图像分割模型需要手机内存3GB以上，并尽量减少其他程序内存占用

### 离线SDK包说明

根据用户的选择，下载的离线SDK，可能包括以下类型：

- EasyEdge
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。



- EasyDL 通用版/全功能AI开发平台BML（原EasyDL专业版）
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。
  - 自适应芯片版：同时整合了以上两种版本，自动在iOS 15以下中使用通用ARM版，在iOS 15以上系统中使用A仿生芯片版，自适应系统，但SDK体积相对较大。
- AI市场试用版SDK

### SDK大小说明

SDK库的二进制与\_TEXT增量约3M。

资源文件大小根据模型不同可能有所差异。

物体检测(高性能)的DemoApp在iPhone 6, iOS 11.4下占用空间实测小于40M。

虽然SDK库文件很大（体现为SDK包文件很大，ipa文件很大），但最终应用在用户设备中所占用的大小会缩小很多。这与multi architectures、bitcode和AppStore的优化有关。

**获取序列号** 生成SDK后，点击获取序列号进入控制台获取。EasyEdge[控制台](#)、EasyDL[控制台](#)、BML[控制台](#)。

试用版SDK在SDK的RES文件夹中的SN.txt中包含试用序列号。

更换序列号、更换设备时，首次使用需要联网激活。激活成功之后，有效期内可离线使用。

### Release Notes

时间	版本	说明
2023.08.31	0.7.13	新增按实例数鉴权；迭代优化
2023.06.29	0.7.12	迭代优化
2023.05.17	0.7.11	CoreML引擎升级，支持更多语义分割模型；兼容横屏；迭代优化
2023.03.16	0.7.10	支持更多语义分割模型；迭代优化
2022.12.29	0.7.9	ARM引擎升级；迭代优化
2022.10.27	0.7.8	支持更多检测模型；迭代优化
2022.09.15	0.7.7	支持更多检测模型；迭代优化
2022.07.28	0.7.6	迭代优化
2022.06.29	0.7.5	支持EasyEdge语义分割模型；CoreML引擎升级，新增EasyEdge检测模型支持；迭代优化
2022.05.18	0.7.4	ARM引擎升级；支持EasyDL物体检测超高精度模型；支持更多加速版模型发布；迭代优化
2022.03.25	0.7.3	ARM引擎升级；支持更多检测模型
2021.12.22	0.7.2	支持EasyEdge更多姿态估计模型；迭代优化
2021.10.20	0.7.1	ARM引擎升级
2021.07.29	0.7.0	迭代优化
2021.04.06	0.6.1	ARM引擎升级
2021.03.09	0.6.0	支持EasyEdge人脸检测及姿态估计模型
2020.12.18	0.5.7	ARM引擎升级
2020.09.17	0.5.6	CoreML引擎升级，支持AI市场试用版SDK
2020.08.11	0.5.5	CoreML支持EasyDL专业版模型，支持EasyEdge OCR模型
2020.06.23	0.5.4	ARM引擎升级
2020.04.16	0.5.3	ARM引擎升级；支持压缩加速版模型
2020.03.13	0.5.2	ARM引擎升级；支持图像分割模型
2020.01.16	0.5.1	ARM引擎升级；增加推荐阈值支持
2019.12.04	0.5.0	ARM引擎升级；增加coreml3的支持
2019.10.24	0.4.5	支持EasyDL专业版；ARM引擎升级
2019.08.30	0.4.4	支持EasyDL经典版图像分类高性能、高精度
2019.06.20	0.4.3	引擎优化
2019.04.12	0.4.1	支持EasyDL经典版物体检测高精度、高性能模型
2019.03.29	0.4.0	引擎优化，支持CoreML；
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

### 快速开始 文件结构说明

```

.EasyEdge-iOS-SDK
├── EasyDLDemo # Demo工程文件
├── LIB # 依赖库
├── RES
│   ├── easyedge # 模型资源文件夹
│   │   ├── model
│   │   ├── params
│   │   ├── label_list.txt
│   │   ├── infer_cfg.json
│   │   └── conf.json
└── DOC # 文档

```

### 测试Demo

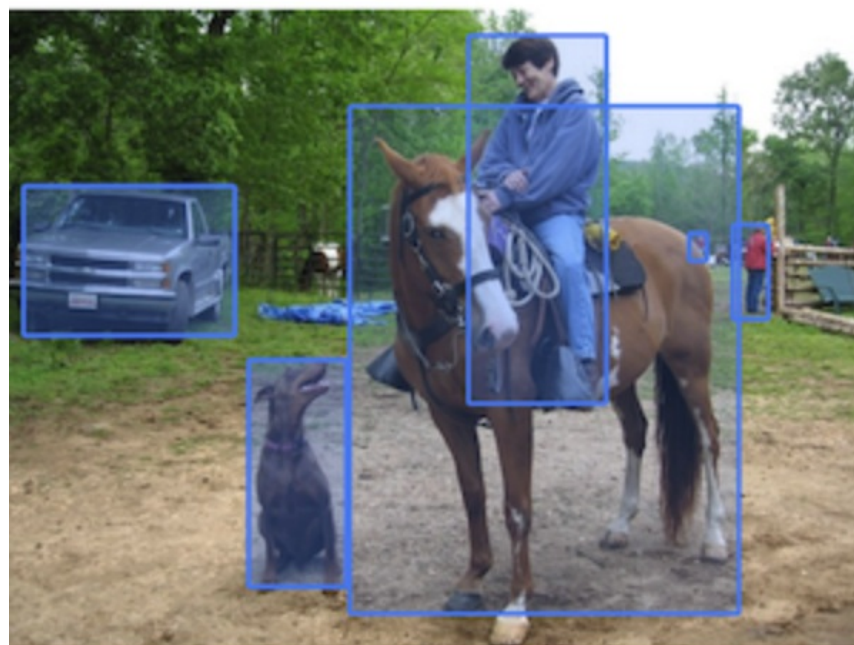
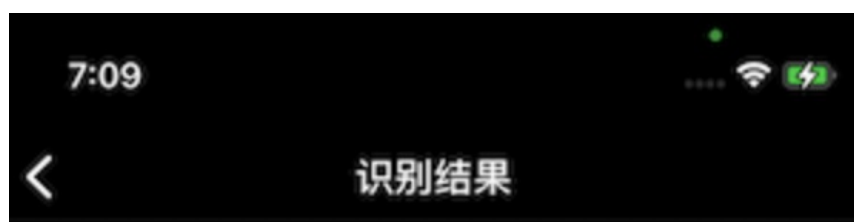
按如下步骤可直接运行 SDK 体验 Demo：

步骤一：用 Xcode 打开 EasyDLDemo/EasyDLDemo.xcodeproj

步骤二：配置开发者自己的签名

步骤三：连接手机运行，不支持模拟器

检测模型运行示例：



阈值：0.30



序号	名称	置信度
1	person	0.632
2	person	0.468
3	car	0.423
4	horse	0.400
5	dog	0.342

重新识别

SDK使用说明 集成指南 步骤一：依赖库集成 步骤二：import <EasyDL/EasyDL.h> , import <Vision/Vision.h>

### 依赖库集成

1. 复制 LIB 目录至项目合适的位置
2. 配置 Build Settings 中 Search paths: 以 SDK 中 LIB 目录路径为例

- Framework Search Paths : \${PROJECT\_DIR}/../LIB/lib
- Header Search Paths : \${PROJECT\_DIR}/../LIB/include
- Library Search Paths : \${PROJECT\_DIR}/../LIB/lib

集成过程如出现错误，请参考 Demo 工程对依赖库的引用

### 使用流程

1. 生成模型，下载SDK 开发者在官网下载的SDK已经自动为开发者配置了模型文件和相关配置，开发者直接运行即可。
2. 使用序列号激活 2.1. 离线激活（默认鉴权方式） 首次联网激活，后续离线使用

将前面申请的序列号填入：

```
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
```

根据序列号类型，序列号与BundleID绑定或与BundleID+设备绑定。  
请确保设备时间正确。

- 2.2. 按实例数激活 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间

填入序列号，配置按实例数鉴权并设置心跳间隔：

```
// 设置序列号
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
// 配置实例数鉴权及心跳间隔，单位：秒
[EasyDL setInstanceAuthMode:10000];
```

### 3. 初始化模型

```
EasyDLModel *_model = [[EasyDLModel alloc] initWithResourceDirectory:@"easymodel" withError:&err];
```

请注意相关资源必须以 folder reference 方式加入Xcode工程。也即默认的easymodel文件夹在Xcode文件列表里显示为蓝色。

### 4. 调用检测接口

```
UIImage *img = .....;
NSArray *result = [model detectUIImage:img withFilterScore:0 andError:&err];

/**
 * 检测图像
 * @param image 带检测图像
 * @param score 只返回得分高于score的结果(0 ~ 1)
 * @return 成功返回识别结果，NSArray的元素为对应模型的结果类型；失败返回nil，并在err中说明错误原因
 */
- (NSArray *)detectUIImage:(UIImage *)image
  withFilterScore:(CGFloat)score
  andError:(NSError **)err;
```

返回的数组类型如下，具体可参考 EasyDLResultData.h 中的定义：

模型类型	类型
图像-图像分类	EasyDLClassfiData
图像-物体检测/人脸检测	EasyDLObjectDetectionData
图像-实例分割/语义分割	EasyDLObjSegmentationData
图像-姿态估计	EasyDLPoseData
图像-文字识别	EasyDLOcrData

### 错误说明

SDK的方法会返回NSError错，直接返回的NSError的错误码定义在EEasyDLErrorCode中。NSError附带message（有时候会附带NSUnderlyingError），开发者可根据code和message进行错误判断和处理。

### FAQ

#### 1. 如何多线程并发预测？

SDK内部已经能充分利用多核的计算能力。不建议使用并发来预测。

如果开发者想并发使用，请务必注意EasyDLModel所有的方法都不是线程安全的。请初始化多个实例进行并发使用，如

```
- (void)testMultiThread {
    UIImage *img = [UIImage imageNamed:@"1.jpeg"];
    NSError *err;
    EasyDLModel * model1 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];
    EasyDLModel * model2 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];

    dispatch_queue_t queue1 = dispatch_queue_create("testQueue", DISPATCH_QUEUE_CONCURRENT);
    dispatch_queue_t queue2 = dispatch_queue_create("testQueue2", DISPATCH_QUEUE_CONCURRENT);

    dispatch_async(queue1, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model1 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"1: %@", res[0];
        }
    });

    dispatch_async(queue2, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model2 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"2: %@", res[0];
        }
    });
}
```

#### 2. 编译时出现 Undefined symbols for architecture arm64: ...

- 出现 `cx11, vtable` 字样：请引入 `libc++.tbd`
- 出现 `cv::Mat` 字样：请引入 `opencv2.framework`
- 出现 `CoreML, VNRequest` 字样：请引入 `CoreML.framework` 并务必 `#import <CoreML/CoreML.h>`

#### 3. 运行时报错 Image not found: xxx ...

请Embed具体报错的库。4.编译时报错：Invalid bitcode version 这个可能是开发者使用的xcodes低于12导致，可以升级至12版本。

### Windows集成文档

#### 简介

本文档介绍图像分类通用小型设备Windows SDK的使用方法。

- 硬件支持：
  - Intel CPU 普通版 \* x86\_64

- CPU 加速版 - Intel Xeon with AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE \* - AMD Core Processors with AVX2
- Intel Movidius Myriad2/Myriad X (仅支持Win10)
- 操作系统支持
  - 普通版：64位 Windows 7 及以上，64位Windows Server2012及以上
  - 加速版：64位 Windows 10，64位Windows Server 2019及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015-2019
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

\*intel 官方合作，拥有更好的适配与性能表现

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | 优化模型算法 | | 2022-09-15 | 1.7.0 | 新增支持表格预测 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | CPU基础版推理引擎优化升级；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | CPU加速版推理引擎优化升级 | | 2021-08-19 | 1.3.2 | 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版 | |

## 快速开始

### 1. 安装依赖

必须安装：

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

Visual C++ Redistributable Packages for Visual Studio 2015-2019

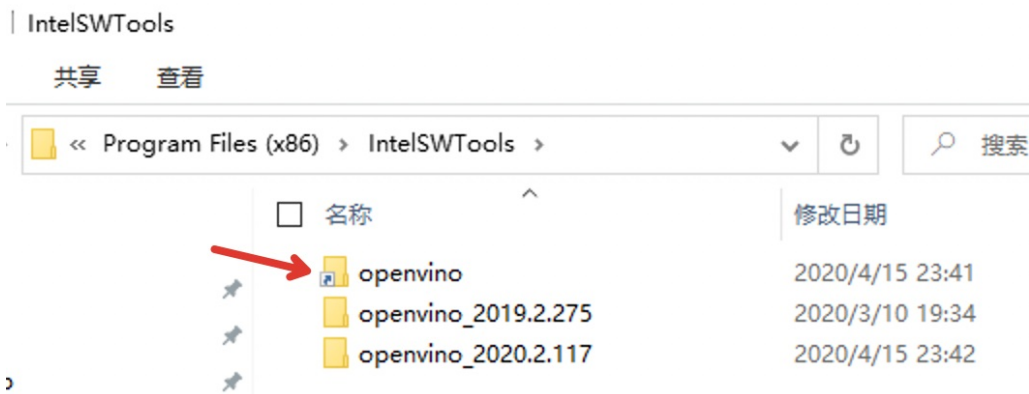
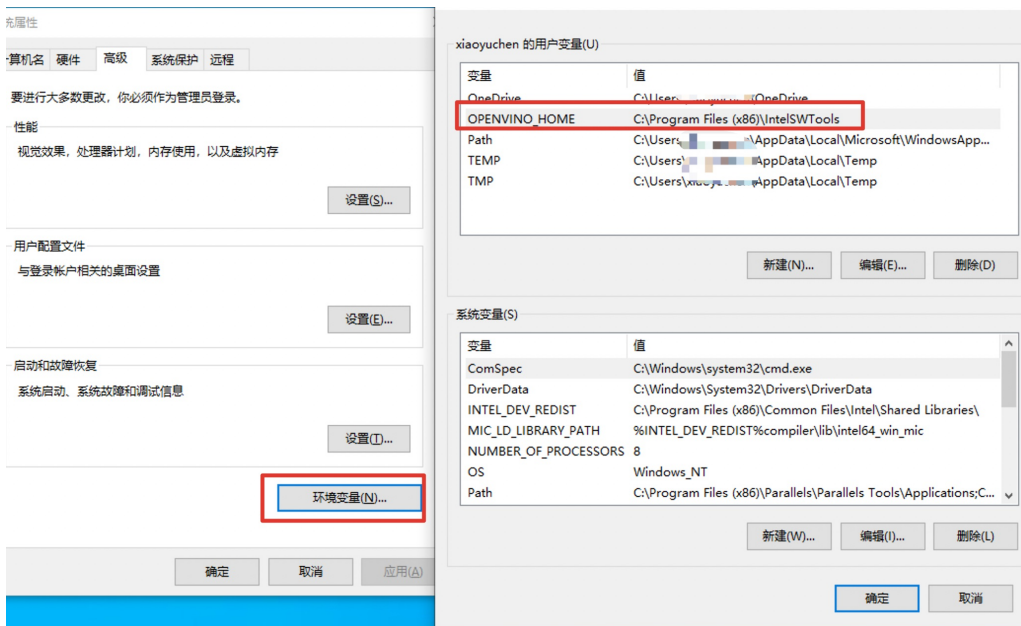
<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-160>

可选安装：

**Openvino (仅使用Python版Intel Movidius必须)**

- 使用 OpenVINO™ toolkit 安装, 请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装, 请参考 [Openvino Inference Engine文档](#) 编译安装 2020.3.1LTS (必须) 版本。

安装完成后, 请设置环境变量OPENVINO\_HOME为您设置的安装地址, 默认是C:\Program Files (x86)\IntelSWTools, 并确保文件夹下的openvino的快捷方式指到了2020.3.1LTS版本。

**注意事项**

1. 安装目录不能包含中文
2. Windows Server 请自行开启, 选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“, 点击安装, 安装之后重启即可。

**2. 运行离线SDK**

解压下载好的SDK, 打开EasyEdge.exe, 输入Serial Num, 选择鉴权模式, 点击“启动服务“, 等待数秒即可启动成功, 本地服务默认运行在

`http://127.0.0.1:24401/`

其他任何语言只需通过HTTP调用即可。

如启动失败, 可参考如下步骤排查:

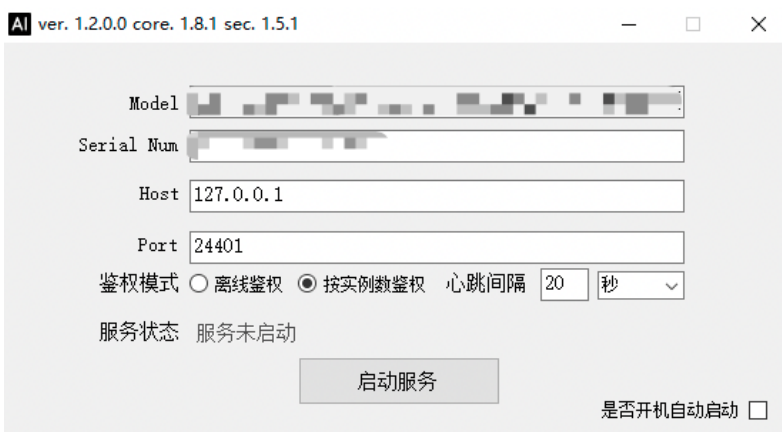




### 2.1 离线鉴权（默认鉴权模式） 首次联网激活，后续离线使用



### 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

### 2.3 序列号激活错误码



错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。

【图像分割】45274 分割-电池-设备端V1

调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

使用说明

调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

结果 获取的结果存储在response字符串中。 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----| | confidence | float | 0~1 | 分类的置信度 | | label | string | | 分类的类别 | | index | number | | 分类的类别 |

## 集成指南

### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

### 基于c++ dll集成

## 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

## 集成方法

参考src目录中的CMakeLists.txt进行集成

## 基于c# dll集成

### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：*.NET Framework 4.5* Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是CPU加速版，需额外确保Openvino安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？SDK设置运行不同的端口，点击运行即可。

### 4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

### 7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

### 8. 勾选“开机自动启动”后，程序闪退

一般是写注册表失败。

可以确认下HKEY\_CURRENT\_USER下Software\Microsoft\Windows\CurrentVersion\Run能否写入（如果不能写入，可能被杀毒软件等工具管制）。也可以尝试基于bin目录下的easyedge\_serving.exe命令行形式的二进制，自行配置开机自启动。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## Linux集成文档-C++

### 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：- 图像分类 - 物体检测 - 图像分割
- 硬件支持：
  - CPU: aarch64 armv7hf
  - GPU: ARM Mali G系列
  - ASIC: Hisilicon NNIE1.1 on aarch64 (Hi3559AV100/Hi3559CV100等)
  - ASIC: Hisilicon NNIE1.2 on armv7l (Hi3519AV100/Hi3559V200等)
  - Intel Movidius Myriad2 / Myriad X on x86\_64
  - Intel Movidius Myriad2 / Myriad X on armv7l
  - Intel Movidius Myriad2 / Myriad X on aarch64
  - Intel iGPU on x86\_64
  - 比特大陆 Bitmain SE5 (BM1684)
  - 瑞芯微 RK3399Pro / RV1109 / RV1126 / RK3568 / RK3588
  - 华为 Atlas200
  - 晶晨 A311D
  - 寒武纪 MLU220 on aarch64
  - 英特尔 iGPU
- 操作系统支持：
  - Linux (Ubuntu, Centos, Debian等)
  - 海思HiLinux
  - 树莓派Raspbian/Debian
  - 瑞芯微Firefly

#### 性能数据参考 [算法性能及适配硬件](#)

**Release Notes** | 时间 | 版本 | 说明 | |---|---|---| | 2023.08.31 | 1.8.3 | Atlas系列Socs支持语义分割模型, Atlas Cann版本升级至6.0.1 | | 2023.06.29 | 1.8.2 | 比特大陆版本升级至V23.03.01 | | 2023.05.17 | 1.8.1 | 新增支持intel iGPU + CPU异构模式 | | 2023.03.16 | 1.8.0 | 新增支持瑞芯微RK3588 | | 2022.10.27 | 1.7.1 | 新增语义分割模型http请求示例 | | 2022.09.15 | 1.7.0 | 新增瑞芯微 RK3568 支持, RK3399Pro、RV1126升级到RKNN1.7.1 | | 2022.07.28 | 1.6.0 | 引擎升级; 新增英特尔 iGPU 支持 | | 2022.04.25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022.03.25 | 1.4.0 | EasyDL新增上线支持晶晨A311D NPU预测引擎; Arm CPU、Arm GPU引擎升级; atlas 200在EasyDL模型增加多个量化加速版本; | | 2021.12.22 | 1.3.5 | RK3399Pro, RV1109/RV1126 SDK扩展模型压缩加速能力, 更新端上推理库版本;边缘控制台IEC功能升级, 适配更多通用小型设备, NNIE 在EasyDL增加量化加速版本; Atlas200升级到Cann5.0.3 | | 2021.06.29 | 1.3.1 | 视频流解析支持调整分辨率; 预测引擎升级; 设备端sdk新增支持瑞芯微RV1109、RV1126 | | 2021.05.13 | 1.3.0 | 新增视频流接入支持; EasyDL模型发布新增多种加速方案选择; 目标追踪支持x86平台的CPU、GPU加速版; 展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.0 | http server服务支持图片通过base64格式调用 | | 2021.01.27 | 1.1.0 | EasyDL经典版分类高性能模型升级; 部分SDK不再需要单独安装OpenCV; 新增RKGPU预测引擎支持; 新增高通骁龙GPU预测引擎支持 | | 2020.12.18 | 1.0.0 | 1.0版本发布! 安全加固升级、性能优化、引擎升级、接口优化等多项更新 | | 2020.10.29 | 0.5.7 | 优化多线程预测细节 | | 2020.09.17 | 0.5.6 | 支持linux aarch64架构的硬件接入intel神经计算棒预测; 支持比特大陆计算盒SE50 BM1684 | | 2020.08.11 | 0.5.5 | 支持linux armv7hf架构硬件(如树莓派)接入intel神经计算棒预测 | | 2020.06.23 | 0.5.4 | arm引擎升级 | | 2020.05.15 | 0.5.3 | 支持EasyDL 专业版新增模型; 支持树莓派(armv7hf, aarch64) | | 2020.04.16 | 0.5.2 | Jetson系列SDK支持多线程infer | | 2020.02.23 | 0.5.0 | 新增支持人脸口罩模型; Jetson SDK支持批量图片推理; ARM支持图像分割 | | 2020.01.16 | 0.4.7 | 上线海思NNIE1.2, 支持EasyEdge以及EasyDL; ARM引擎升级; 增加推荐阈值支持 | | 2019.12.26 | 0.4.6 | 海思NNIE支持EasyDL专业版 | | 2019.11.02 | 0.4.5 | 移除curl依赖; 支持自动编译OpenCV; 支持EasyDL 专业版 Yolov3; 支持EasyDL经典版高精度物体检测模型升级 | | 2019.10.25 | 0.4.4 | ARM引擎升级, 性能提升30%; 支持EasyDL专业版模型 | | 2019.09.23 | 0.4.3 | 增加海思NNIE加速芯片支持 | | 2019.08.30 | 0.4.2 | ARM引擎升级; 支持分类高性能与高精度模型 | | 2019.07.25 | 0.4.1 | 引擎升级, 性能提升 | | 2019.06.11 | 0.3.3 | paddle引擎升级; 性能提升 | | 2019.05.16 | 0.3.2 | 新增armv7l支持 | | 2019.04.25 | 0.3.1 | 优化硬件支持 | | 2019.03.29 | 0.3.0 | ARM64 支持; 效果提升 | | 2019.02.20 | 0.2.1 | paddle引擎支持; 效果提升 | | 2018.11.30 | 0.1.0 | 第一版! |

【1.0 接口升级】参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES一起使用。更新模型时，如果SDK版本号有更新，请务必同时更新SDK，旧版本的SDK可能无法正确适配新发布出来的RES。

## 快速开始

SDK在以下环境中测试通过

- aarch64(arm64), Ubuntu 16.04, gcc 5.3 (RK3399)
- Hi3559AV100, aarch64, Ubuntu 16.04, gcc 5.3
- Hi3519AV100, armv7l, HiLinux 4.9.37, (Hi3519AV100R001C02SPC020)
- armv7hf, Raspbian, (Raspberry 3b)
- aarch64, Raspbian, (Raspberry 4b)
- armv7hf, Raspbian, (Raspberry 3b+)
- armv7hf, Ubuntu 16.04, (RK3288)
- Bitmain se50 BM1684, Debian 9
- Rockchip rk3399pro, Ubuntu 18.04
- Rockchip rv1126, Debain 10
- Rockchip rk3568, Ubuntu 20.04
- Rockchip rk3588, Ubuntu 20.04
- Atlas200(华为官网指定的Ubuntu 18.04版本)
- Amlogic A311D, Ubuntu 20.04
- MLU220, aarch64, Ubuntu 18.04

## 安装依赖

依赖包括

- cmake 3+
- gcc 5.4 以上(需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.5 (可选)

**依赖说明：**树莓派 树莓派Raspberr默认认为armv7hf系统，使用SDK包中名称中包含 armv7hf\_ARM\_的tar包。如果是aarch64系统，使用SDK包中名称中包含 aarch64\_ARM\_的tar包。

在安装前可通过以下命令查看是32位还是64位：

```
getconf LONG_BIT
32
```

**依赖说明：**比特大陆SE计算盒 需要安装SophonSDK V23.05.01及以上版本，SDK的默认安装位置为/opt/sophon/，如SDK安装在自定义地址，需在CMakeList.txt中指定SDK安装地址：

```
**这里修改并填入所使用的SophonSDK路径**
set(EDGE_BMSDK_ROOT "{这里填写sdk路径}")
```

可通过命令 `bm-smi` 查看内部SDK和驱动的版本号（SophonSDK V23.05.01对应的内部SDK和驱动为0.4.6）。对于使用旧版BM1684 SDK或者低版本SophonSDK的用户，可参考[SophonSDK安装包](#)中的《LIBSOPHON 使用手册》先卸载旧版BM1684 SDK，安装、升级SophonSDK。

**依赖说明：**海思开发板 海思开发板需要根据海思SDK文档配置开运行环境和编译环境，SDK和opencv都需要在该编译环境中编译。NNIE1.2用

arm-himix200-linux交叉编译好的opencv, 下载链接:<https://pan.baidu.com/s/13QW0ReeWx4ZwgYg4lretyw> 密码:yq0s。下载后修改SDK CMakeList.txt

依赖说明: RK3399Pro 所有用例基于 Npu driver版本1.7.1的RK3399pro开发板测试通过, SDK采用预编译模式, 请务必确保板上驱动版本为 1.7.1 查看RK3399Pro板上driver版本方法: `dpkg -l | grep 3399pro`

依赖说明: RV1109/RV1126 所有用例基于Rknn\_server版本1.7.3的RV1126开发板测试通过, SDK采用预编译模式, 请务必确保板上驱动版本为 1.7.3 查看RV1109/RV1126板上Rknn\_server版本方法: `strings /usr/bin/rknn_server | grep build`

依赖说明: RK3568 所有用例基于Rknn\_server版本1.2.0的RK3568开发板测试通过, 查看RK3568板上Rknn\_server版本方法: `strings /usr/bin/rknn_server | grep build`

依赖说明: RK3588 RK3588开发板需要确保环境正确安装了RKNPU驱动, 平台用例基于v0.8.0版本的RKNPU驱动测试通过, 查看RK3588NPU驱动版本的方法: `sudo cat /sys/kernel/debug/rknpu/version`

依赖说明: 晶晨A311D 所有用例基于晶晨A311D开发板测试通过, 需要驱动版本为 6.4.4.3 (下载驱动请联系开发版厂商) 查看晶晨A311D开发板驱动版本方法: `dmesg | grep Galcore`

依赖说明: 英特尔iGPU 用户在使用英特尔iGPU SDK前, 需要根据英特尔[官方文档](#)提前安装好英特尔集成显卡驱动以及相关基础软件环境, 安装完成后通过 `clinfo` 指令确认OpenCL能够正常识别到集成显卡信息, 正确识别集显情况下`clinfo`指令输出参考如下:

```
root@baidu-qit(anM38-M000):~# clinfo
Number of Platforms: 1
Platform Name: Intel(R) OpenCL HD Graphics
Platform Vendor: Intel(R) Corporation
Platform Version: OpenCL 3.0
Platform Profile: FULL_PROFILE
Platform Extensions: cl_khr_byte_addressable_store cl_khr_device_uuid cl_khr_fp16 cl_khr_global_int32_base_atomics cl_khr_global_int32_extended_atomics cl_khr_icd cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics cl_khr_command_queue_families cl_khr_subgroups cl_khr_subgroup_size cl_khr_subgroups_short cl_khr_spir cl_khr_accelerator cl_khr_driver_diagnostics cl_khr_priority_hints cl_khr_throttle_hints cl_khr_create_command_queues cl_khr_subgroups_order cl_khr_subgroups_range cl_khr_il_program cl_khr_mem_force_host_memory cl_khr_subgroup_extended_types cl_khr_subgroup_non_uniform_vote cl_khr_subgroup_barrier cl_khr_subgroup_non_uniform_arithmetic cl_khr_subgroup_shuffle cl_khr_subgroup_shuffle_relative cl_khr_subgroup_clustered_reduce cl_khr_device_attribute_query cl_khr_suggested_local_work_size cl_khr_split_work_group_barrier cl_khr_fp64 cl_khr_subgroups cl_khr_spirv_device_side_ave_motion_estimation cl_khr_spirv_media_block_io cl_khr_spirv_subgroups cl_khr_spirv_no_integer_wrap_decoration cl_khr_unified_shared_memory cl_khr_image2d_image_writes cl_khr_image3d_image_writes cl_khr_image2d_from_buffer cl_khr_depth_images cl_khr_3d_image_writes cl_khr_media_block_io cl_khr_v_api_media_sharing cl_khr_sharing_format_query cl_khr_pcl_bus_info
Platform host timer resolution: 1ns
Platform Extensions: INTEL
Platform Name: Intel(R) OpenCL HD Graphics
Number of devices: 1
Device Name: Intel(R) UHD Graphics 630 [0x9bc8]
Device Vendor: Intel(R) Corporation
Device Vendor ID: 0x8086
Device Version: OpenCL 3.0 NEO
Driver Version: 22.53.25242.13
Device OpenCL C Version: OpenCL C 1.2
Device Type: GPU
Device Profile: FULL_PROFILE
Device Available: Yes
Compiler Available: Yes
Linker Available: Yes
Max compute units: 24
Max clock frequency: 1150MHz (Core)
Device Partition:
```

使用序列号激活 请在官网获取序列号

纯离线服务说明
发布纯离线服务, 将训练完成的模型部署在本地, 离线调用模型。可以选择将模型部署在本地服务器、小型设备、软硬一体方案专项适配硬件上。
通过API, SDK进一步集成, 灵活适应不同业务场景。
发布前服务 控制台
服务器 通用小型设备 专项适配硬件
SDK API
此处发布, 下载的SDK为免授权SDK, 需要前往控制台[获取序列号](#)激活后才能正常使用。SDK内附有对应版本的Demo及开发文档, 开发者可参考源代码完成开发。
模型名称 发布版本 应用平台 模型加速 发布状态 发布时间
sun\_小目标检测 134318-V1 查看性能报告 通用X86 CPU-Linux 基础版 已发布 2021-08-19 20:24 下载SDK
通用X86 CPU-Linux 精度无损失加速 已发布 2021-08-19 20:24 下载加速版SDK
英伟达GPU-Linux 基础版 已发布 2021-08-19 20:35 下载SDK
英伟达GPU-Linux 精度无损失加速 已发布 2021-08-19 20:34 下载加速版SDK
基础版 已发布 2021-08-19 18:17 下载SDK

SDK内bin目录下提供预编译二进制文件, 可直接运行(二进制运行详细说明参考下一小节), 用于图片推理和模型http服务, 在二进制参数的 serial\_num(或者serial\_key)处填入序列号可自动完成联网激活 (请确保硬件首次激活时能够连接公网, 如果确实不具备联网条件, 需要使用纯离线模式激活, 请下载使用百度智能边缘控制台纳管SDK)

```
**SDK内提供的一些二进制文件, 填入序列号可完成自动激活, 以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}
```

如果是基于源码集成, 设置序列号方法如下

```
global_controller()->set_licence_key("")
```

默认情况下(联网激活或者离线激活的场景), 按照上述说明正确设置序列号即可, 如果是实例数鉴权模式 (请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式, 仅实例数鉴权需要进行下面的变量或者源码设置) 实例数鉴权环境变量设置方法



```
export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30
```

实例鉴权源码设置方法

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)
```

基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

对于硬件使用为：Intel Movidius Myriad2 / Myriad X / IGPU on Linux x86\_64 / armv7hf / aarch64，在编译或运行demo程序前执行以下命令：

```
source ${cpp_kit位置路径}/thirdparty/opencvino/bin/setupvars.sh
或者执行
source ${cpp_kit位置路径}/thirdparty/opencvino/setupvars.sh (opencvino-2022.1+)
如果SDK内不包含setupvars.sh脚本，请忽略该提示
```

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号尽首次激活需要使用，激活后可不用填序列号也能运行二进制)

```
**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=./lib ./easyedge_image_inference ../../RES /xxx/cat.jpeg "1111-1111-1111-1111"
```

demo运行效果：



图片加载失败

```
> ./easyedge_image_inference ../../RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=./lib ./edgekit_serving --cfg=./edgekit_serving.yml

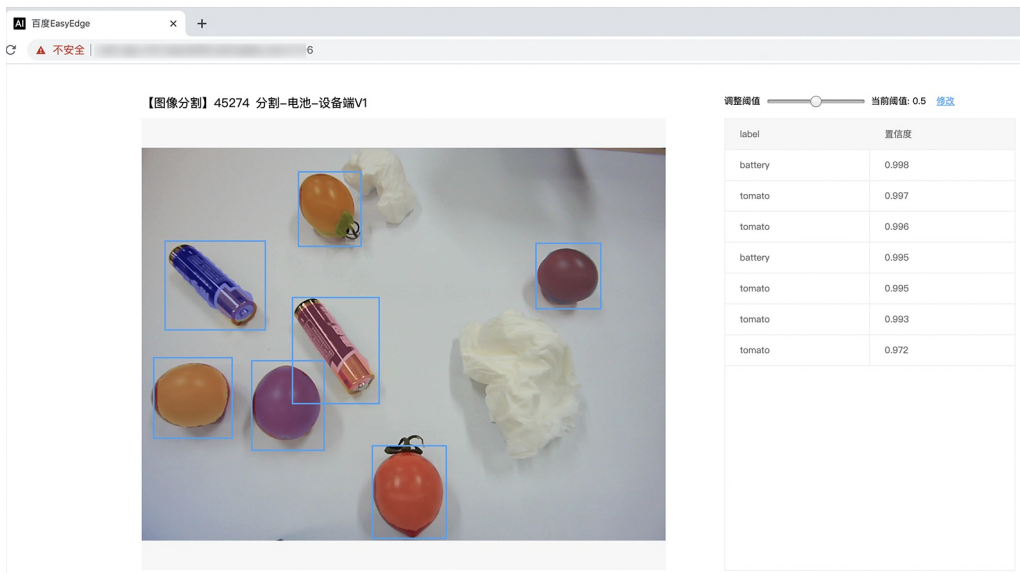
**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=./lib ./easyedge_serving ../../RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

```
HTTP(or Webservice) is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试，网页右侧会展示模型推理结果





同时，可以调用HTTP接口来访问服务。

**请求http服务** 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

python示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }

        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

### 集成文档

使用该方式，将运行库嵌入到开发者的程序当中。 **编译demo项目** SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```

cd src
mkdir build && cd build
cmake .. && make
./easymodel_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easymodel_image_inference {模型RES文件夹} {测试图片路径}

```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DDEGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

对于硬件使用为Intel Movidius Myriad2 / Myriad X 的，如果宿主机找不到神经计算棒Intel® Neural Compute Stick，需要执行以下命令添加USB Rules：

```
cp ${cpp_kit位置路径}/thirdparty/openvino/deployment_tools/inference_engine/external/97-myriad-usbboot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig
```

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 1: 配置运行参数
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor；这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame，需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频，需在video_config中开启配置
}
}
```

对于口罩检测模型，将 EdgePredictorConfig config修改为PaddleMultiStageConfig config即可。

口罩检测模型请注意输入图片中人脸大小建议保持在 88到9696像素之间，可根据场景远近程度缩放图片后再传入SDK。

**SDK参数配置** SDK的参数通过EdgePredictorConfig::set\_config和global\_controller()->set\_config配置。set\_config的所有key在easyedge\_xxxx\_config.h中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过EdgePredictorConfig::set\_config设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过global\_controller()->set\_config设置

以序列号为为例，KEY的说明如下：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";
```

使用方法如下：

```
EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");
```

具体支持的运行参数可以参考开发工具包中的头文件的详细说明。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};
```

## 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

$y_2$  \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域，0代表非目标区域
```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding，此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

classVideoDecoding :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

```
struct VideoConfig
```

```
/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type; // 输入源类型
    std::string source_value; // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0}; // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false}; // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0}; // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false}; // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path; // frame存储为视频文件的路径
    bool save_all{false}; // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};
```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。input\_fps：用于抽帧前设置fps。resolution：设置摄像头采样的分辨率，其值请参考easyedge\_video.h中的定义，注意该分辨率调整仅对输入源为摄像头时有效。conf：高级选项。部分配置会通过该map来设置。

**注意：**1.如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。2.使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的demo\_video\_inference。

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

### 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

## 2. http接口详细说明 http 请求方式一：无额外编码 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```

import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果

```

Python请求示例 (视频测试, 注意：区别于图片预测, 需指定Content-Type；否则会调用图片推理接口)

```

import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data).json()

```

http 请求方法二：json格式，图片传base64格式字符串 HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 图像分类网络：body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 物体检测和实例分割网络：Body请求示例：

```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)



```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    },
  ]
}
```

#### 其他配置

##### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



## FAQ

### 1. 如何处理一些 undefined reference / error while loading shared libraries?

如：./easyedge\_demo: error while loading shared libraries: libeasyedge.so.1: cannot open shared object file: No such file or directory 这是因为二进制运行时ld无法找到依赖的库。如果是正确cmake && make 的程序，会自动处理好链接，一般不会出现此类问题。

遇到该问题时，请找到具体的库的位置，设置LD\_LIBRARY\_PATH。

示例一：libverify.so.1: cannot open shared object file: No such file or directory 链接找不到libveirfy.so文件，一般可通过 export LD\_LIBRARY\_PATH=\${LD\_LIBRARY\_PATH}:/lib 解决(实际冒号后面添加的路径以libverify.so文件所在的路径为准)

示例二：libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 链接找不到libopencv\_videoio.so文件，一般可通过 export LD\_LIBRARY\_PATH=\${LD\_LIBRARY\_PATH}:/thirdparty/opencv/lib 解决(实际冒号后面添加的路径以libopencv\_videoio.so所在路径为准)

### 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

3. 如何将我的模型运行为一个http服务？目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

### 4. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

### 5. 运行SDK报错 Authorization failed

情况一：日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/baidu/easyedge 目录，再重新激活。

### 6. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

7. 运行NNIE引擎报错 `std::bad_alloc` 检查开发板可用内存，一些比较大的网络占用内存较多，推荐内存500M以上

8. 运行二进制时，提示 `libverify.so cannot open shared object file`

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 编译时报错：`file format not recognized` 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中，再解压缩、编译

🔗 Linux集成文档-Python

## 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法，适用于 EasyDL 通用版和BML。

- 网络类型支持：图像分类，物体检测
- 硬件支持：
  - Intel Movidius Myriad2 / Myriad X / IGPU
  - 瑞芯微 RK3399Pro
- 语言支持：*Intel Movidius Myriad2 / Myriad X / IGPU: Python 3.5, 3.6, 3.7* 瑞芯微 RK3399Pro: Python 3.6

## Release Notes

时间	版本	说明
2022.10.27	1.3.5	新增Arm7 CPU、Arm8 CPU、Jetson、华为昇腾Atlas开发板对应Python SDK，支持图像分类、物体检测、人脸检测、实例分割；新增 Intel IGPU 支持
2022.05.18	1.3.0	新增RK3399Pro NPU对应Python SDK，支持图像分类、物体检测
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持加速棒
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.07.19	1.1.7	提供模型更新工具
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】 序列号配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。

**依赖说明：** Intel Movidius 加速棒 使用Intel Movidius加速棒 SDK、 Intel IGPU 预测时，必须安装 OpenVINO 预测引擎，两种方式：

- 使用 OpenVINO™ toolkit 安装，请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装，请参考 [Openvino Inference Engine文档](#) 编译安装 2020.3.1 (必须) 版本。

安装完毕，运行之前，请按照OpenVino的文档 设置环境变量

```
source /opt/intel/openvino/bin/setupvars.sh
```

依赖说明：RK3399Pro 所有用例基于 Npu driver版本1.7.3的RK3399pro开发板测试通过 查看RK3399Pro板上driver版本方法：运行sdk内提供 demo项目，日志里会提供API和Driver版本信息

```
2022-12-20 14:26:07,765 VERBOSE [EasyEdge] [rockchip_edge_predictor.cpp:87] 547887054864 Create predictor , 5029536
D RKNNAPI: =====
D RKNNAPI: RKNN VERSION:
D RKNNAPI: API: 1.7.3 (0cfd4a1 build: 2022-08-15 17:10:10)
D RKNNAPI: DRV: 1.7.3 (c4ea832 build: 2022-08-13 09:13:08)
D RKNNAPI: =====
```

升级399Pro driver版本参考瑞芯微github：[https://github.com/airockchip/RK3399Pro\\_npu](https://github.com/airockchip/RK3399Pro_npu) 2. 安装 easyedge python wheel 包 安装说明：Intel Movidius 加速棒 / Intel IGPU

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。

安装说明：RK3399Pro

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_RK3399Pro-{版本号}-cp36-cp36m-linux_aarch64.whl
```

具体名称以 SDK 包中的 whl 为准，特别注意这里要同时安装两个whl包 安装说明：ArmV7 CPU

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_armv7l.whl
pip3 install -U EasyEdge_Devkit_ARM-{版本号}-cp36-cp36m-linux_armv7l.whl
```

安装说明：ArmV8 CPU (Aarch64 CPU)

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_ARM-{版本号}-cp36-cp36m-linux_aarch64.whl
```

安装说明：Jetson SDK

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_JetPack{版本号}-{版本号}-cp36-cp36m-linux_aarch64.whl
```

安装说明：华为昇腾Atlas开发板

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
EasyEdge_Devkit_Atlas200-{版本号}-cp36-cp36m-linux_aarch64.whl
```

### 3. 使用序列号激活



#### 获取序列号

此发布、下载的SDK为未授权SDK，需要前往控制台获取序列号激活后才能正常使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标test	134318-v1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

#### 修改 demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的设置），需要调用函数指定实例数鉴权模式，并且实例数鉴权模式下，支持指定 license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，参考

```
pred.set_instance_auth_mode()
pred.set_instance_update_interval(200)
```

#### 4. 测试 Demo

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



使用说明

使用流程

```
import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.MOVIDIUS, engine=edge.Engine.OPENVINO)
pred.infer_image((numpy.ndarray的图片))
pred.close()
```

初始化

- 接口

```
def init(self,
          model_dir,
          device=Device.LOCAL,
          engine=Engine.PADDLE_FLUID,
          config_file='conf.json',
          preprocess_file='preprocess_args.json',
          model_file='model',
          params_file='params',
          graph_file='graph.ncsmodel',
          label_file='label_list.txt',
          device_id=0
        ):
    """
    Args:
        device: Device.CPU
        engine: Engine.PADDLE_FLUID
        model_dir: str
            model dir
        preprocess_file: str
        model_file: str
        params_file: str
        graph_file: str
        label_file: str
        device_id: int

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success

    """
```

预测图像

- 接口

```
def infer_image(self, img,
                 threshold=0.3,
                 channel_order='HWC',
                 color_format='BGR',
                 data_type='numpy'):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

**升级模型** 适用于经典版升级模型，执行bash update\_model.sh，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

## FAQ

### Q: 运行SDK报错 Authorization failed

**情况一：日志显示 Http perform failed: null respond** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx)** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

**情况三：ArmV7、ArmV8 CPU、Jetson、Atlas Python SDK日志提示ImportError: libavformat.so.58: cannot open shared object file: No such file or**



`directory` 或者其他类似so找不到 可以在LD\_LIBRARY\_PATH环境变量加上libs和thirdpartylibs路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内libs和thirdpartylibs路径的方法如下(以Atlas SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas200 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## Linux集成文档-Atlas

### 简介

本文档介绍EasyEdge/EasyDL的Linux Atlas SDK的使用方法。

注意Atlas有两种产品形态，Atlas 200和Atlas 300，请参见此处的[文档说明](#)

- 网络类型支持：图像分类
- 硬件支持：
  - CPU: aarch64
  - Atlas 200 卡
- 操作系统支持：Atlas指定的Linux版本，Ubuntu 16.04 aarch64，请从Atlas文档中下载。

### 性能数据

数据仅供参考，实际数值根据使用线程数、利用率等情况可能有所波动

模型类型	模型算法	芯片类型	SDK类型	实测硬件	单次预测耗时
EasyDL 图像分类	高性能	Atlas 200	Atlas 200	Atlas 200DK	9ms
EasyDL 图像分类	高精度	Atlas 200	Atlas 200	Atlas 200DK	12ms
EasyDL 物体检测	高性能	Atlas 200	Atlas 200	Atlas 200DK	11ms
EasyDL 物体检测	高精度	Atlas 200	Atlas 200	Atlas 200DK	31ms

### Release Notes

时间	版本	说明
2020.6.15	0.2	支持物体检测
2020.3.10	0.1	初始版本，支持图像分类

### 测试atlas 200的官方demo

请参见此处的[文档说明](#)，搭建开发环境，测试atlas 200的mindstudio demo通过后，再测试

### 快速开始

SDK在以下环境中测试通过

- ubuntu 16.04, aarch64-linux-gnu-g++ 5.4，编译器
- ubuntu 16.04，开发板

Atlas DDK 的ddk\_info信息：

```
{
  "VERSION": "1.3.T34.B891",
  "NAME": "DDK",
  "TARGET": "Atlas DK"
}
```

## 2. 测试Demo

**编译运行：**下载后，模型资源文件默认已经打包在开发者下载的SDK包中，

Step 1：运行一次unpack.sh脚本，会得到测试demo。

Step 2：请在官网获取序列号，填写在demo\_async.cpp及demo\_sync.cpp的开始处license\_key字段。



step3：准备测试图片

覆盖image目录下的 1.jpg，更多图片可以用于demo中的批量测试模式

step4：修改test\_200.sh下的以下开发板登录信息

```
export DDK_PATH=$HOME/tools/che/ddk/ddk # ddk的安装路径
SSH_USER=HwHiAiUser@192.168.3.25 # 200 开发板的ssh登录信息
PORT=8822 # 200 开发板的ssh登录端口
```

step: 运行demo，会自动编译OpenCV 3.4库

```
cd demo
sh test_200.sh
```

图像分类的demo运行效果：

```
[stat] [100001]image/1.jpg(4 images) time used: 41ms (at 1583765958531) total:705ms
[result][100001]image/1.jpg[281470472005664] is: n07747607 orange 0.973633 950;

n07747607 orange 分类名
0.973633 分类概率
950 分类名的序号
```

物体检测的demo运行效果：

```
[stat] time used: 101ms; all time used:478
images[3] result:
label:no2_ynen:prob:0.985352 loc:[(0.459961,0.839844), (0.5625,0.988281)]

no2_ynen 分类名，也可以获取分类名的序号
0.985352 分类概率
loc:[(0.459961,0.839844), (0.5625,0.988281)]，检测框的位置。(0.459961,0.839844)表示左上角的点，(0.5625,0.988281)右下角的点；
如原始图片608，左上角(0.459961*608,0.839844*608)，右下角(0.5625*608,0.988281*608)
```

### SDK接口使用

使用该方式，将运行库嵌入到开发者的程序当中。

### 同步接口使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ;
auto predictor = global_controller()->CreateEdgePredictor(config);
int ret = predictor->init();
# 若返回非0, 请查看输出日志排查错误原因。
auto img = cv::imread({图片路径});
// step 3: 预测图像
std::vector<EdgeResultData> result2;
predictor->infer(img, result2);
# 解析result2即可获取结果

```

## 异步接口使用流程

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 3: 创建Predictor ; 这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 4: 设置异步回调
predictor->set_result_handler(YOUR_HANDLER);

// step 5: 初始化
int ret = predictor->init();
**若返回非0, 请查看输出日志排查错误原因。 **

// step 6: 预测图像
auto img = cv::imread({图片路径});
color_format = kBGR;
float threshold = 0.1;

uint64_t seq_id;
predictor->infer_async(img, color_format, 0.1, nullptr, seq_id);
**YOUR_HANDLER里面有seq_id的回调结果**

```

## 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

- 接口

```
virtual int set_licence_key(const std::string& license) = 0;
```

## 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

## FAQ

1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3`

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

## 2. EasyDL 离线SDK与云服务效果不一致，如何处理？

目前离线SDK与云服务的处理有些许差异，具体如下：

- 图像分类模型：离线SDK与云服务使用通用(非快速训练、非AutoDL Transfer)的效果类似
- 物体检测模型：离线SDK的高精度模型与云服务的精度较低，服务性能更佳的效果类似

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

## 端云协同服务说明

### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 联网状态下在平台管理设备运行状态、资源利用率

目前通用小型设备的应用平台支持Linux-ARM，具体使用流程请参考下方文档。

### 使用流程

#### Step 1 发布端云协同部署包

在[我的部署包](#)页面点击「发布端云协同部署包」

端云协同服务 &gt; 我的部署包

## 端云协同服务说明

[点击收起](#)

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「我的本地设备」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「下发部署包到设备」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

发布端云协同部署包

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
----------	------	------	------	--------	---------	-------	----



暂无可用数据

请稍后再试



填写服务名称，选择模型版本并提交发布

端云协同服务 &gt; 发布端云协同部署包

设备类型  服务器  通用小型设备

模型名称 test2021

端云协同服务名称

选择版本 V1

选择系统和芯片  Linux

通用ARM

发布部署包

在列表查看部署包发布状态

端云协同服务 > 我的部署包

**端云协同服务说明** 点击收起

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「我的本地设备」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「下发部署包到设备」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

**发布端云协同部署包**

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
ecc	264	通用小型设备	Linux-通用ARM	V1	V1-已发布	0	下发到设备 发布新版本
从发布页过来的	246	服务器	Linux-通用X86 CPU	V2, V1	V2-已发布	1	下发到设备 发布新版本 服务详情
图像分类高精度_猫狗-265	265	通用小型设备	Linux-通用ARM	-	V1-发布中	0	

每页显示 10 < 1 >

### Step 2 新增设备并激活

在[我的本地设备](#)页面新增设备

端云协同服务 > 我的本地设备

**我的本地设备** 点击收起

在本页面新增设备、联网激活本地设备后，即可将「[我的部署包](#)」页面发布成功的部署包一键下发到设备上。设备联网时，可以查看设备上部署的服务、设备的运行状态、资源利用率等信息。

**新增设备**

设备名称	创建时间	设备类型	应用平台	设备连接状态	最新同步时间	操作
------	------	------	------	--------	--------	----

**新增设备** ×

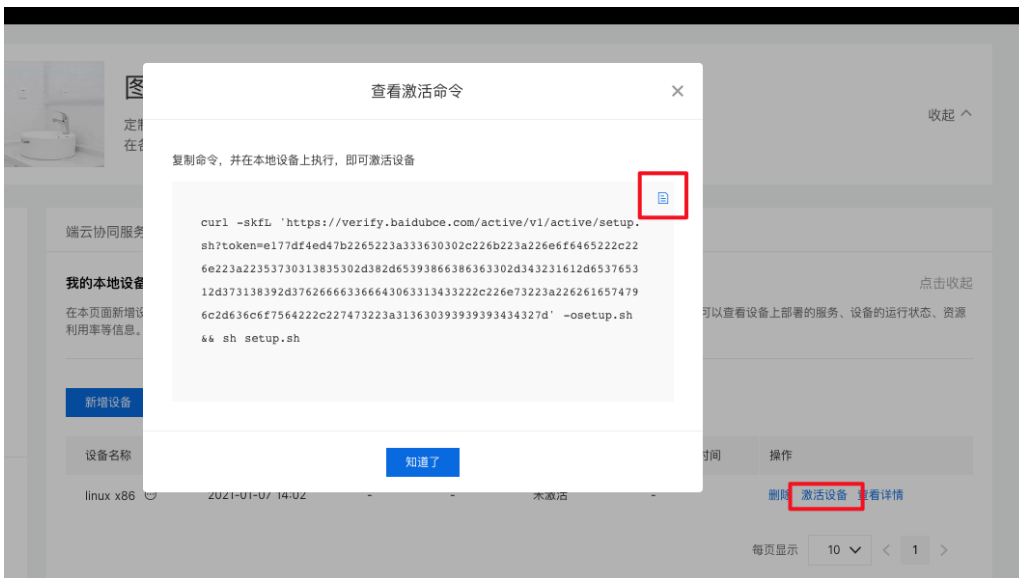
设备名称

备注信息

0/50

确认
取消

在列表中，点击设备对应的「激活设备」操作，复制激活命令并在本地设备上执行即可

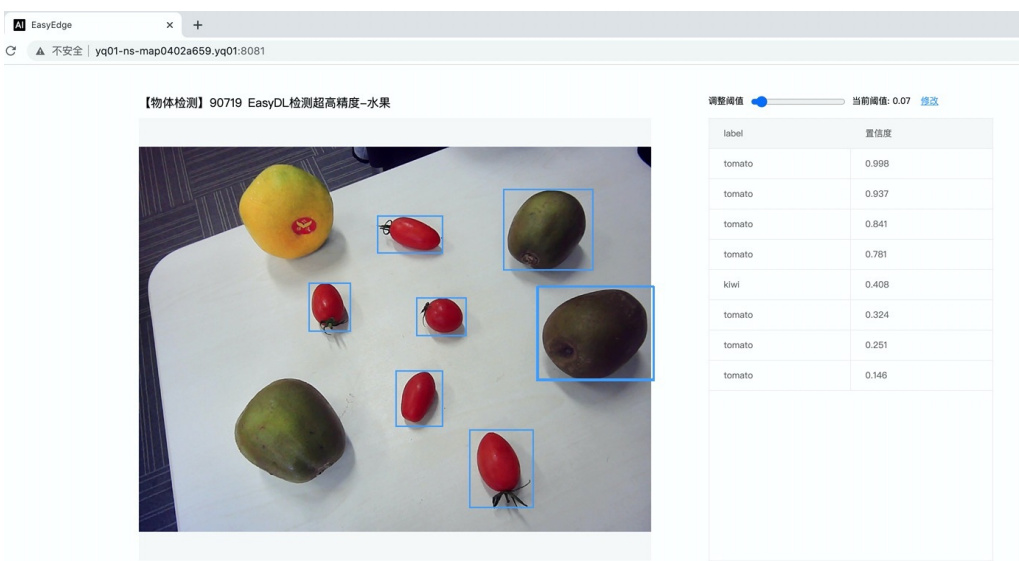


### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入 `http://{设备ip}:{服务端点, 默认8080}`，即可预览效果：



具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

### 云端管理说明

### 模型部署包管理

在[我的部署包](#)页面可以进行已发布的模型部署包的管理。

## 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。

端云协同服务 > 我的部署包

## 端云协同服务说明

[点击收起](#)

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「[我的本地设备](#)」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「[下发部署包到设备](#)」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

发布端云协同部署包

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
ecc	264	通用小型设备	Linux-通用ARM	V1	V1-已发布	0	<a href="#">下发到设备</a> <a href="#">发布新版本</a>
	246	服务器	Linux-通用X86 CPU	V2, V1	V2-已发布	1	<a href="#">下发到设备</a> <a href="#">发布新版本</a> <a href="#">服务详情</a>
	265	通用小型设备	Linux-通用ARM	-	V1-发布中	0	

每页显示  < 1 >

### 发布新版本

将最新训练的模型版本发布为服务，发布成功后，即可从云端下发到设备

服务名称	ecc
模型ID	264
选择新版本	<input type="text" value="V1"/>

新版本发布成功后，即可在「[下发部署包到设备](#)」页面或当前服务的「[服务详情](#)」页面，将新版本下发到本地设备上。

端云协同服务 > 我的部署包

## 端云协同服务说明

[点击收起](#)

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「[我的本地设备](#)」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「[下发部署包到设备](#)」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

发布端云协同部署包

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
ecc	264	通用小型设备	Linux-通用ARM	V1	V1-已发布	0	<a href="#">下发到设备</a> <a href="#">发布新版本</a>
	246	服务器	Linux-通用X86 CPU	V2, V1	V2-已发布	1	<a href="#">下发到设备</a> <a href="#">发布新版本</a> <a href="#">服务详情</a>
图像分类高精度_猫狗-265	265	通用小型设备	Linux-通用ARM	-	V1-发布中	0	



端云协同服务 &gt; 服务详情

服务名称 模型ID 246 当前可用版本 V2, V1 部署设备数 1/1  
 设备类型 服务器 应用平台 Linux-通用X86 CPU

[下发到更多设备](#)

设备名称	最新下发模型版本	部署到期时间	设备连接状态	服务下发状态	最新同步时间	操作
ubuntu-local-fjy	V2	2021-02-06	在线	V2下发成功	2021-01-07 14:45	<a href="#">查看服务配置</a> <a href="#">查看设备详情</a> <a href="#">下发新版本</a> <a href="#">移除设备</a>

## 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。

端云协同服务 > 服务详情

服务名称 模型ID 246 当前可用版本 V2, V1 部署设备数 1/1  
 设备类型 服务器 应用平台 Linux-通用X86 CPU

[下发到更多设备](#)

设备名称	最新下发模型版本	部署到期时间	设备连接状态	服务下发状态	最新同步时间	操作
ubuntu-local-fjy	V2	2021-02-06	在线	V2下发成功	2021-01-07 14:45	<a href="#">查看服务配置</a> <a href="#">查看设备详情</a> <a href="#">下发新版本</a> <a href="#">移除设备</a>

## 本地设备管理

在[我的本地设备](#)页面可以进行所有本地设备的管理。

## 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：

端云协同服务 > 我的本地设备

**我的本地设备** [点击收起](#)

在本页面新增设备、联网激活本地设备后，即可将「我的部署包」页面发布成功的部署包一键下发到设备上。设备联网时，可以查看设备上部署的服务、设备的运行状态、资源利用率等信息。

[新增设备](#)

设备名称	创建时间	设备类型	应用平台	设备连接状态	最新同步时间	操作
123123	2021-01-07 13:25	-	-	未激活	-	<a href="#">删除</a> <a href="#">激活设备</a> <a href="#">查看详情</a>
showcase-test	2020-12-15 17:57	服务器	Linux-AMD64(x86-64)	离线	2020-12-16 19:22	<a href="#">激活设备</a> <a href="#">查看详情</a>
linux-x86-zqw-2	2020-12-15 16:24	服务器	Linux-AMD64(x86-64)	离线	2020-12-16 19:42	<a href="#">激活设备</a> <a href="#">查看详情</a>
linux-x86-zqw	2020-12-15 15:29	服务器	Linux-AMD64(x86-64)	离线	2020-12-15 16:40	<a href="#">激活设备</a> <a href="#">查看详情</a>
firefly_rk3399pro	2020-12-15 14:39	-	-	离线	2020-12-15 21:33	<a href="#">激活设备</a> <a href="#">查看详情</a>
ubuntu-local-fjy	2020-12-14 21:38	服务器	Linux-AMD64(x86-64)	在线	2021-01-07 15:01	<a href="#">查看详情</a>
hfl-1	2020-12-14 19:24	服务器	Linux-AMD64(x86-64)	离线	2020-12-16 14:18	<a href="#">激活设备</a> <a href="#">查看详情</a>
edge新增	2020-12-11 14:52	-	-	未激活	-	<a href="#">删除</a> <a href="#">激活设备</a> <a href="#">查看详情</a>

设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况

端云协同服务 > 我的本地设备 > ubuntu-local-fjy

### 设备详情

设备名称 **ubuntu-local-fjy**      连接状态 **在线**      实时刷新  OFF  
 设备类型 **服务器**      应用平台 **Linux-AMD64(x86-64)**      最新同步时间 **2021-01-07 15:00**



CPU使用率  
**31.1%**



内存使用率  
**35.8%**

### 端云协同服务详情

服务名称	模型ID	CPU占比	内存使用情况	内存占比	操作
██████████	246	0.01%	156.7MB	0.93%	<a href="#">查看服务配置</a>

☞ 软硬一体方案部署

☞ 如何获取图像分类软硬一体产品

为进一步提升前端智能计算的用户体验，EasyDL推出了多款软硬一体方案。将高性能硬件与EasyDL图像分类/物体检测模型深度适配，可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松。[了解不同方案](#)

方案获取流程如下：

Step 1：在EasyDL训练专项适配所选硬件的图像分类/物体检测模型，迭代模型至效果满足业务要求

### 训练配置

\* 部署方式  公有云部署  EasyEdge本地部署  浏览器/小程序部署 限时免费 [如何选择部署方式?](#)

\* 选择设备  服务器  通用小型设备  专项适配硬件

选择硬件  Edgeboard(FZ)  Edgeboard(VMX)  翔影(Air/Pro) NEW  Jetson(Nano/TX2/Xavier) [了解不同方案](#)

\* 选择算法  高精度  高性能  AI市场已购模型

高级训练配置

名称	规格	算力	速度比例	价格
<input type="radio"/> GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	免费
<input type="radio"/> GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡¥0.36/分钟 (50小时*节点免费)
<input checked="" type="radio"/> GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡¥0.45/分钟 (53.28小时*节点免费)

训练费用 **免费**

当前GPU V100卡剩余免费资源：53.28(小时 \* 节点) 为保证训练任务顺利进行，请保证您的账户余额充足，可点击进行充值 [立即充值](#)，推荐购买特惠小时包，可享受至少8折优惠训练 [前往购买](#)

[开始训练](#)

Step 2：发布模型时选择对应硬件

## 发布模型

选择模型:	test02	▼	
部署方式:	专项硬件适配SDK	▼	
选择硬件:	Jetson(Nano/TX2/Xavier)	▼	<a href="#">了解更多</a>
选择版本:	V2	▼	

[提交](#)

Step 3 : 在AI市场购买方案获得硬件和用于激活专用SDK的专用序列号，参考文档集成后，即可实现离线AI预测

我的模型 > test02 > V1的专项硬件适配SDK服务详情

### 专项硬件适配SDK获取

专项硬件适配SDK是EasyDL软硬一体方案的软件部分，建议在AI市场购买整体方案，同时获得专用SDK激活序列号及专项适配硬件

如已在其他渠道购买硬件，可点击「获取序列号」前往控制台，支持申请专用的测试序列号、购买专用的永久有效序列号

专项硬件适配SDK	操作
EdgeBoard(VMX)专用SDK	<a href="#">前往AI市场购买</a> <a href="#">下载SDK</a> ▼ <a href="#">管理序列号</a>

如有其他硬件方案需求，请在百度智能云控制台内[提交工单](#)反馈。

🔗 [图像分类EdgeBoard\(FZ\)专用SDK集成文档](#)

#### 简介

本文档介绍 EasyEdge/EasyDL在EdgeBoard®边缘计算盒/Lite计算卡上的专用软件的使用流程。

EdgeBoard系列硬件可直接应用于AI项目研发与部署，具有高性能、易携带、通用性强、开发简单等四大优点。

详细硬件参数请在AI市场浏览。

EdgeBoard产品使用手册：<https://ai.baidu.com/ai-doc/HWCE/Yk3b86gvp>

#### 软核版本

CPP-SDK版本	对应软核
1.3.2、1.3.4、1.3.5	1.8.1
1.3.0、1.3.1、1.3.2、1.3.4	1.8
0.5.7-1.2.1	1.5
0.5.2+	1.4

SDK升级需配合EdgeBoard硬件软核升级，建议升级软核为SDK对应版本，否则可能出现结果错误或者其他异常。

可以通过 `dmesg | grep "DRIVER Version"` 命令获取EdgeBoard当前的软核版本

**Release Notes 注意\***：升级完成相应的软核之后需要重启机器生效。

sdk对应的软核说明：**如果客户使用的软核是mobile版本的，需要使用1.4的SDK；如果不是mobile 版本，可以选择1.5+（目前最高版本更新至1.8.1）版本的SDK使用。**

1.5+版本的软核以及sdk更新情况如下表所示：

时间	版本	说明	EdgeBoard非mobile对应的软核以及特性	
2021.1 2.20	1.3.5	升预测引擎为PaddleLite 1.8.1,推理库支持了Ubuntu18.04文件系统	<a href="https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk">https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk</a> (含有EB升级Ubuntu18.04系统的步骤)	
2021.1 0.15	1.3.2、 1.3.4、1.3.4	推理库支持了Ubuntu18.04文件系统	<a href="https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk">https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk</a>	
2021.0 6.29	1.3.1	视频流解析支持分辨率调整；预测引擎升级；	<a href="https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x">https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x</a>	
2021.0 5.14	1.3.0	新增视频流接入支持；展示已发布模型性能评估报告	<a href="https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x">https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x</a>	
2021.0 5.14	1.2.1	功能无更新	<a href="https://ai.baidu.com/ai-doc/HWCE/okqiwkm32">https://ai.baidu.com/ai-doc/HWCE/okqiwkm32</a>	
2020.1 0.29	0.5.7	预测引擎切换为PaddleLite 1.5		-
2019.1 2.27	0.4.5	引擎升级，支持zu5/zu3，支持EasyDL 高精度检测模型		-
2019.0 7.25	0.4.0	EdgeBoard SDK Release!		-

mobile软核以及sdk更新情况如下表所示：| 时间 | 版本 | 说明 | EdgeBoard mobile对应的软核以及特性 | | --- | --- | --- | --- | --- |  
|2021.05.14|1.2.1|功能无更新|<https://ai.baidu.com/ai-doc/HWCE/okqiwkm32>|<https://ai.baidu.com/ai-doc/HWCE/Lkqiwlziw>|

## 快速开始

开发者从EasyEdge/EasyDL下载的软件部署包中，包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行EdgeBoard计算盒。

部署包中包含多版本SDK：

- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.8*`：适用于EdgeBoard 1.5+软核
- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.4*`：适用于EdgeBoard 1.4软核

SDK文件结构

```

baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.5_*
├── README.txt
├── bin
│   ├── easyedge_image_inference
│   ├── easyedge_serving
│   └── easyedge_video_inference
├── include
│   └── easyedge
├── lib
│   ├── libeasyedge.so -> libeasyedge.so.1
│   ├── libeasyedge.so.1 -> libeasyedge.so.1.3.1
│   ├── libeasyedge.so.1.3.1
│   ├── libeasyedge_static.a
│   ├── libeasyedge_videoio.so -> libeasyedge_videoio.so.1
│   ├── libeasyedge_videoio.so.1 -> libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio_static.a
│   ├── libpaddle_full_api_shared.so -> libpaddle_full_api_shared.so.1.8.0
│   ├── libpaddle_full_api_shared.so.1.8.0
│   ├── libverify.so -> libverify.so.1
│   ├── libverify.so.1 -> libverify.so.1.0.0
│   └── libverify.so.1.0.0
├── now_sre.log
├── src
│   ├── CMakeLists.txt
│   ├── cmake
│   ├── common
│   ├── demo_image_inference
│   ├── demo_serving
│   └── demo_video_inference
└── thirdparty
    └── opencv

```

1.1.0+的SDK自带OpenCV，src编译的时候会引用thirdparty/opencv路径下的头文件和库文件。

## Demo使用流程

用户在AI市场购买计算盒之后，请参考以下步骤进行集成和试用。

### 1. 将计算盒连接电源

指示灯亮起，等待约1分钟。

- 参考[EdgeBoard使用文档](#)配置网口或串口连接。登录EdgeBoard计算盒。
- 加载驱动（开机加载一次即可）。

```
insmod /home/root/workspace/driver/{zu9|zu5|zu3}/fpgadv.ko
```

根据购买的版本，选择合适的驱动。若未加载驱动，可能报错：

```
Failed to to fpga device: -1
```

- 设置系统时间（系统时间必须正确）

```
date --set "2019-5-18 20:48:00"
```

### 2. (可选) 启动HTTP服务

部署包中附带了HTTP服务功能，开发者可以进入SDK根目录，运行easyedge\_serving程序启动HTTP服务。

```

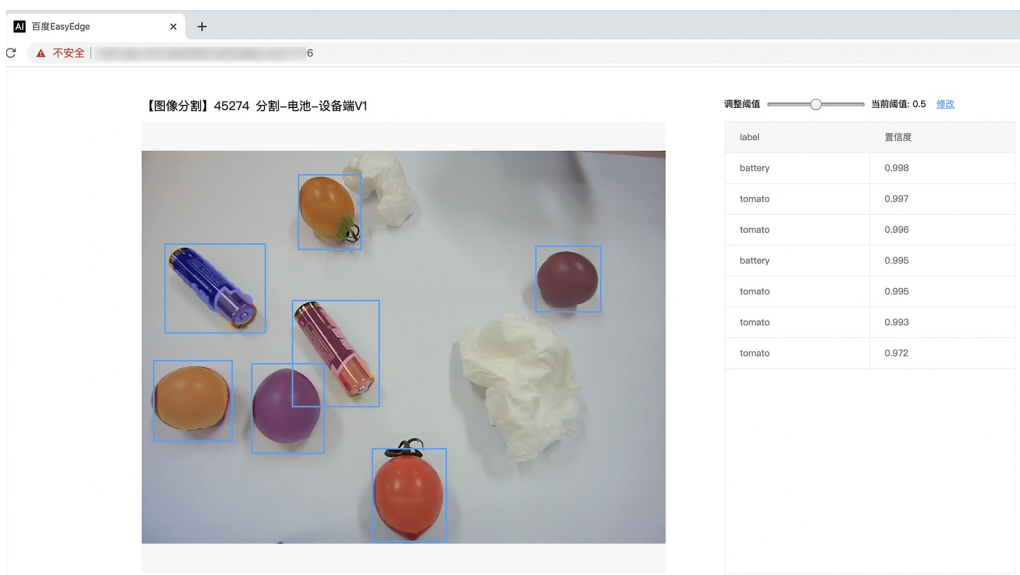
**./easyedge_serving {RES目录} "" {绑定的host, 默认0.0.0.0} {绑定的端口, 默认24401}**
cd ${SDK_ROOT}
export LD_LIBRARY_PATH=./lib
./demo/easyedge_serving ../../RES ""

```

日志显示

```
2019-07-18 13:27:05,941 INFO [EasyEdge] [http_server.cpp:136] 547974369280 Serving at 0.0.0.0:24401
```

则启动成功。此时可直接在浏览器中输入 `http://{EdgeBoard计算盒ip地址}:24401/`，在h5中测试模型效果。



同时，可以调用HTTP接口来访问盒子。具体参考下文接口说明。

EdgeBoard HTTP Server 目前使用的是单线程处理请求。

### 3. 编译运行Demo

编译：

```
cd src
mkdir build && cd build
cmake .. && make
```

运行

```
./easyedge_image_inference {RES资源文件夹路径} {测试图片路径}
```

便可看到识别结果。

使用说明

使用流程

激活成功之后，有效期内可离线使用。

1. 配置PaddleFluidConfig
2. 新建Predictor :`global_controller()->CreateEdgePredictor(config);`
3. 初始化 `predictor->init()`
4. 传入图片开始识别`predictor->infer(img, ...);`

目前EdgeBoard暂不支持并行多模型计算。

接口说明

设置序列号 请在网页控制台中申请序列号，并在init初始化前设置。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定CONTROLLER\_KEY\_AUTH\_MODE为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

## 预测图片

```
/**
 * @brief 同步预测接口
 * inference synchronous
 * Supported by most chip and engine
 * @param image: must be BGR, HWC format (opencv default)
 * @param result
 * @param threshold
 * @return
 */
virtual int infer(
    cv::Mat &image, std::vector<EdgeResultData> &result, float threshold = 0.1
) = 0;
```

## 识别结果说明

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // object detection field
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。
};
```

## 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考demo文件中使用opencv绘制矩形的逻辑。

## HTTP 私有服务请求说明

### http 请求参数

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	0.1

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Cpp label=C#

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

Cpp label=C++ 需要安装curl



```

#include <sys/stat.h>
#include <curl/curl.h>
#include <iostream>
#include <string>
#define S_ISREG(m) (((m) & 0170000) == (0100000))
#define S_ISDIR(m) (((m) & 0170000) == (0040000))

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"
", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"
", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s
", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

### Java请求示例

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

- 接口

class `VideoDecoding` :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};          // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;            // frame存储为视频文件的路径
    bool save_all{false};             // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

- 3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 错误说明

SDK所有主动报出的错误，均覆盖在EdgeStatus枚举中。同时SDK会有详细的错误日志，开发者可以打开Debug日志查看额外说明：

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

#### FAQ

##### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

可以通过安装`libcurl3 libcurl-openssl1.0-dev`来解决。如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库`easyedge_static.a`，自己指定需要的Library的版本。

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} paddle-mobile)
```

##### 2. error while loading shared libraries: libeasyedge.so.0.4.0: cannot open shared object file: No such file or directory

类似错误包括`libpaddle-mobile.so`找不到。

直接运行SDK自带的二进制可能会有这个问题，设置`LD_LIBRARY_PATH`为SDK部署包中的lib目录即可。开发者自行使用CMake编译的二进制可以有效管理.so的依赖。

##### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

##### 4. 预测过程中报内存不足“Killed”

此问题仅出现在ZU5，因为FZ5A带vcu，给他预留的内存过大导致，如果用不到VCU可以把这部分改小。修改`/run/media/mmcblk1p1/uEnv.txt`：

```
ethaddr=00:0a:35:00:00:09
uenvcmd=fatload mmc 1 0x3000000 image.ub && bootm 0x3000000

bootargs=earlycon console=ttyPS0,115200 clk_ignore_unused cpuidle.off=1 root=/dev/mmcblk1p2 rw rootwait cma=128M
```

注意中间空行要保留。

##### 5. 预测结果异常

如果购买的计算盒较早，驱动文件较旧，而SDK比较新（或SDK比较旧，但是计算盒较新），可能出现结果异常，如结果均为空或者nan。请参

考“软核版本”小节更新软核和驱动版本。

## 6. 编译过程报错file format not recognized

```
libeasymedge.so: file format not recognized; treating as linker script
```

下载的SDK zip包需要放到板子内部后，再解压、编译。

7. 提示 driver\_version(1.4.0) not match paddle\_lite\_version(1.5.1) 需更新驱动，否则可能导致结果异常。参考“软核版本”小节。

## 🔗 图像分类EdgeBoard(VMX)专用SDK集成文档

### 简介

本文档旨在介绍 EasyDL在EdgeBoard USB加速卡VMX（以下简称VMX加速卡或加速卡）上的专用软件的使用流程。EdgeBoard系列硬件适用于项目开发及部署，具有高性能、易携带、通用性强、开发简单等四大优点。您可在[AI市场](#)了解EdgeBoard相关系列产品，同时可以在[软硬一体方案](#)了解性能数据。

注意：本型号主要面向产品集成和企业项目，未同时售卖散热片和外壳，部分情况下芯片温度较高，**开发过程中，请勿用手触摸，谨防烫伤**

### 硬件介绍

VMX加速卡，采用Intel® Movidius™ 视觉 MyriadX处理器芯片，通过 USB3.0 通讯type-c接口方式，配合外围电路即可将该模组嵌入到第三方智能化产品中，采用标准 USB通讯协议，对接简单，开发速度快，具有强大的深度学习计算功能。可通过OpenVINO™和OpenCV软件库工具链移植算法，兼容百度PaddlePaddle支持Paddle2onnx和PaddleHub并集成EasyDL，使产品应用范围广，性能更稳定，增强用户体验。

VMX加速卡适用于深度学习加速，能够解决复杂的人工智能软硬件设计挑战，它可以集成基于视觉的加速器和推理引擎来实现深度边缘学习的解决方案。（3D/2D人脸识别、人头检测、人脸属性分析（性别、年龄）、人脸特征比对、手势及姿态识别、物体检测及分类、算法移植等功能。）

### 硬件配置与说明

核心板模块: Intel® Movidius™ MyriadX，内置内存LP-DDR4 4GBit。

#### • 硬件指标

#### CPU

- o Intel® Movidius Myriad X MA2485 Vision Processing Unit
- o Total performance of over 4 trillion operations per second (TOPS)
- o Over 1 TOPS performance on neural network inference w/ NCE accelerator
- o 16 Programmable 128-bit VLIW Vector Processors
- o 16 Configurable MIPI Lanes w/ enhanced Vision Accelerators
- o 2.5 MB of Homogenous On-Chip Memory w/ 4Gbit LPDDR4

#### Size

- o 38mm x 38mm

**Interface** o USB TYPE C (USB3.0) 辅助接口精简设计

**Boot** o USB 启动模式 - 内置 switch 缺省模式设置

**Power** o 平均功耗0.5W~2.2W

**Security** o 支持 eFuse 加密

### 运行说明

VMX加速卡包含独立的AI运算芯片，采用 USB Type-C通讯方式，通讯协议简单可靠，可连接不同芯片架构主机，包括 X86、ARM SOC等。加速卡运行需要通过TypeC接口连接宿主机执行，宿主机目前支持的软硬件环境包括：

- Linux: x86-64, armv7hf
- Windows: x86-64, Windows 10

使用过程中，请尽量避免直接接触板卡元器件；或者使用防静电锡纸包裹板卡。

### 快速开始 Linux

开发者从EasyDL训练模型之后，下载的软件部署包中，包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行。

## Release Notes

### Python SDK

时间	版本	说明
2020.12.18	1.2.0	性能优化；接口优化升级；推理引擎升级
2020.09.17	1.1.19	支持更多模型与平台

Python SDK适用于Linux x86-64和Windows平台。

2020-12-18: 【接口升级】 Python SDK序列号配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

### C++ SDK

时间	版本	说明
2021.06.29	1.3.1	视频流解析支持分辨率调整
2021.05.14	1.3.0	新增视频流接入支持；展示已发布模型性能评估报告
2020.12.18	1.0.0	性能优化；接口优化升级；推理引擎升级
2020.09.17	0.5.6	新增C++ SDK，支持Linux armv7hf（树莓派）架构的硬件接入VMX预测

C++ SDK适用于Linux x86-64、Linux armv7hf平台。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

**将加速卡连接宿主机** 请使用质量合规的usb线连接。连接之后，检查设备是否被操作系统识别：Linux 通过 `lsusb -v` 命令检查是否有 Myriad设备：

```
> sudo lsusb -v | grep -C 5 Myriad
bMaxPacketSize0    64
idVendor           0x03e7
idProduct          0x2485
bcdDevice          0.01
iManufacturer      1 Movidius Ltd.
iProduct           2 Movidius MyriadX
iSerial            3 03e72485
```

Windows 可以在设备管理器中查询。

如果使用 VirtualBox 之类的虚拟机，请在虚拟机加入 03e7:24 和 03e7:f63b 两个 usb 设备。

## 获取并安装依赖

### 1) 安装依赖

宿主机与sdk为以下情况：1  Windows x86-64：请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS 版本 2  Linux x86-64且使用Python SDK时必须：请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS 版本，安装时可忽略Configure the Model Optimizer及后续部分。

安装完毕，运行之前，请按照OpenVino的文档 设置环境变量

```
source /opt/intel/opencvino/bin/setupvars.sh
```

2) 从EasyDL 控制台获取SDK 在任意位置解压缩。

**获取序列号** 从[AI市场订单详情](#)或者[EasyDL控制台](#)获取序列号。

更换序列号、更换设备时，首次使用需要联网激活。激活成功之后，有效期内可离线使用。

请确保激活设备时使用的 操作系统账号与后续使用时运行的账号一致，否则会造成验证失败

## Python SDK

### 1. 安装wheel包

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp37-cp37m-linux_x86_64.whl
```

注意，请根据python的版本选择对应的whl文件，其中,1.2.0是SDK版本号，cp37表示是python3.7版本

--

注意，pip安装时请添加-U参数

### 2. 将步骤2中获得的序列号 填入demo.py

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

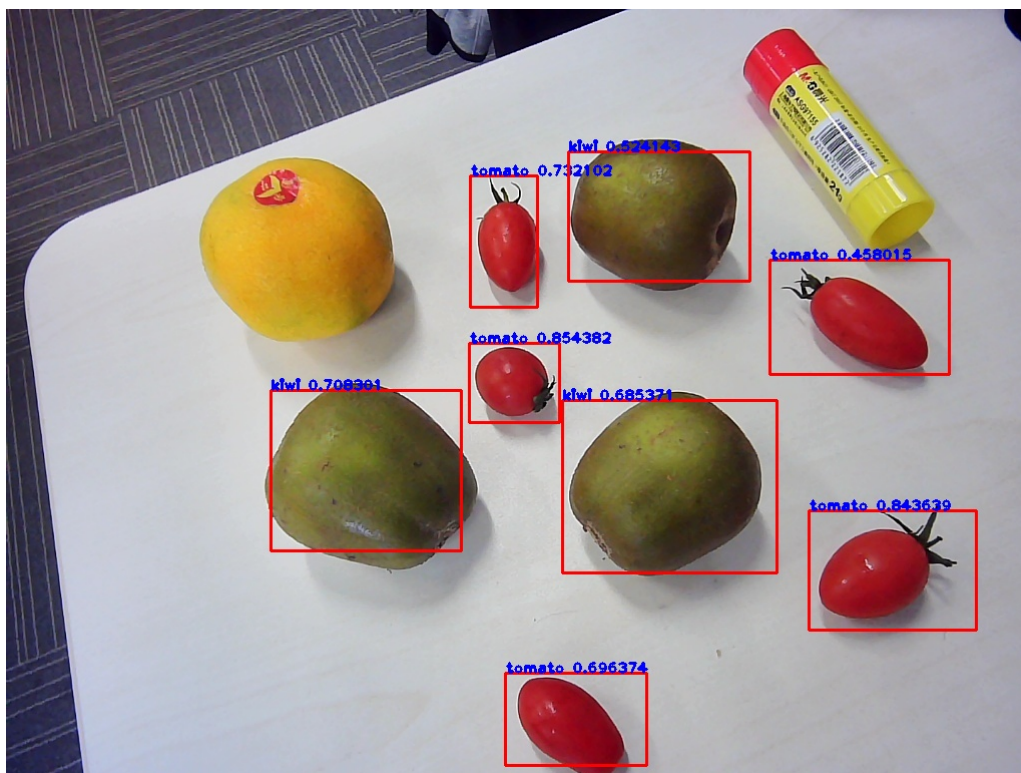
默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的设置），需要调用函数指定实例数鉴权模式，并且实例数鉴权模式下，支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，参考

```
pred.set_instance_auth_mode()
pred.set_instance_update_interval(200)
```

### 3. 测试demo.py

```
python3 demo.py {模型资源文件夹RES路径} {待识别的图片路径}
```

生成的样例结果图片如下：



## 使用流程

```
import BaiduAI.EasyEdge as edge
```

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.MOVIDIUS, engine=edge.Engine.OPENVINO)
pred.infer_image((numpy.ndarray的图片))
pred.close()
```

接口的详细说明请主要参考 SDK 中的接口注释

## 接口说明

### Program

- 初始化



```

def init(self,
    model_dir,
    device=Device.CPU,
    engine=Engine.NCSDK,
    config_file='conf.json',
    preprocess_file='preprocess_args.json',
    model_file='model',
    params_file='params',
    graph_file='graph.ncsmodel',
    label_file='label_list.txt',
    device_id=0,
    **kwargs
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device
        engine: BaiduAI.EasyEdge.Engine
        preprocess_file: str
        model_file: str
        params_file: str
        graph_file: str ncs的模型文件 或 PaddleV2的模型文件
        label_file: str
        device_id: int 设备ID
        thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success

    """

```

- 预测单张图像

```

def infer_image(self, img, threshold=None,
    channel_order='HWC',
    color_format='BGR',
    data_type='numpy'
):
    """
    Args:
        img: np.ndarray or bytes
        channel_order(string):
            channel order: HWC or CHW
        color_format(string):
            color format order: RGB or BGR
        threshold(float):
            only return result with confidence larger than threshold
        data_type(string): 仅在图像分割时有意义。 'numpy' or 'string'
            'numpy': 返回已解析的mask
            'string': 返回未解析的mask游程编码

    Returns:
        list

    """

```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中, data\_type为numpy时, 返回图像掩码的二维数组

```
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

## C++ SDK

### 使用说明

模型资源文件默认已经打包在开发者下载的SDK包中。请先将SDK包整体拷贝到具体运行的宿主机设备中，再解压缩编译；

在编译或运行demo程序前执行以下命令：

```
source ${cpp_kit位置路径}/thirdparty/opencv/bin/setupvars.sh
```

如果opencv预测引擎找不到设备需要执行以下命令：

```
sudo cp ${cpp_kit位置路径}/thirdparty/opencv/deployment_tools/inference_engine/external/97-myriad-usbboot.rules
/etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig ````
```

### 使用流程

```
// step 1: 配置运行参数
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num"); // 设置序列号
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread({图片路径});
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame，需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频，需在video_config中开启配置
}
```

**运行参数配置** 运行参数的配置通过结构体EdgePredictorConfig完成，其定义如下所示：

```
struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};
```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时部分参数也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

具体支持的运行参数可以参考开发工具包中的头文件。

### 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

### 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```

cv::Mat mask为图像掩码的二维数组
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域, 0代表非目标区域

```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码, 解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding, 此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};      // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};             // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};      // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;        // frame存储为视频文件的路径
    bool save_all{false};        // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定`CONTROLLER_KEY_AUTH_MODE`为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

- 1. 开启http服务 http服务的启动参考`demo_serving.cpp`文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

- 2. 请求http服务

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试。

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Java请求示例

- http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

其他配置

- 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```



效果如下：



## Linux FAQ

1. EasyDL 离线 SDK 与云服务效果不一致，如何处理？我们会逐渐消除这部分差异，如果开发者发现差异较大，可通过[工单](#)、[论坛](#)联系我们协助处理。

2. 硬件出现问题或者出现故障怎么办？软件使用有问题怎么处理？

- 如果持续在静电较多的环境中使用，建议使用防静电锡纸包裹板卡
- 如果硬件无法启动等故障，您可以通过商品页联系供应商处理；其它硬件问题，您可以邮件 [edgeboard-vmx.com](mailto:edgeboard-vmx.com)，我们将在0-2日内处理您的问题。为加快处理进度，您在邮件中，尽量描述清楚问题或者需求细节，避免来回沟通。
- 软件使用问题，请尽量通过[工单](#)、[论坛](#)联系我们协助处理。

3. 运行时报错：NC\_ERROR

```
Can not init Myriad device: NC_ERROR
```

一般是硬件没有插上，请确保Isusb能够找到该硬件。或者等待几秒后再试。

## 快速开始 Windows

1. 安装依赖

将操作系统升级到Windows 10

安装.NET Framework4.5

```
https://www.microsoft.com/zh-CN/download/details.aspx?id=42642
```

Visual C++ Redistributable Packages for Visual Studio 2013

```
https://www.microsoft.com/zh-cn/download/details.aspx?id=40784
```

Visual C++ Redistributable Packages for Visual Studio 2015

```
https://www.microsoft.com/zh-cn/download/details.aspx?id=48145
```

## 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num



点击“启动服务”，等待数秒即可启动成功，本地服务

默认运行在

```
http://127.0.0.1:24401/
```

其他任何语言只需通过HTTP调用即可。

## 接口调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=img).json()
```

C## 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

**返回参数** | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----| | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 | | x1, y1 | float | 0~1 | 物体检测，矩形的左上角坐标（相对长宽的比例值） | | x2, y2 | float | 0~1 | 物体检测，矩形的右下角坐标（相对长宽的比例值） |

关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

Windows FAQ

1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下：*.NET Framework 4.5* Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

2. 服务调用时返回为空，怎么处理？调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <http://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <http://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted? Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 其他问题 如果无法解决，可到论坛发帖：<http://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## 🔗 图像分类Jetson专用SDK集成文档

### 简介

本文档介绍EasyEdge/EasyDL的Jetson SDK的使用方法。Jetson SDK支持的硬件包括Jetson nano，Jetson TX2，Jetson AGX Xavier和Jetson Xavier NX。您可在[AI市场](#)了解Jetson相关系列产品，同时可以在[软硬一体方案](#)了解部署方案。

### 模型支持：

- EasyDL图像：图像分类高精度，图像分类高性能，物体检测高精度，物体检测均衡，物体检测高性能，目标跟踪单标签模型。
- BML：
  - 公开数据集预训练模型：SSD-MobileNetV1，YOLOv3-DarkNet，YOLOv3-MobileNetV1，ResNet50，ResNet101，SE-ResNeXt50，SE-ResNeXt101，MobileNetV2，EfficientNetB0\_small，EfficientNetB4，MobileNetV3\_large\_x1\_0，ResNet18\_vd，SE\_ResNet18\_vd，Xception71。
  - 百度超大规模数据集预训练模型：YOLOv3-DarkNet，MobileNetV3\_large\_x1\_0，ResNet50\_vd，ResNet101\_vd。
- EasyEdge：EasyEdge支持的模型较多，详见[查看模型网络适配硬件](#)。若模型不在此列表，可以尝试使用自定义网络生成端计算组件。

**软件版本支持** 使用EasyDL的Jetson系列SDK需要安装指定版本的JetPack和相关组件。所支持的JetPack版本会随着SDK版本的升级和新版本JetPack的推出而不断的更新。在使用SDK前请务必保证软件版本满足此处声明版本。目前所支持的JetPack版本包括：

- JetPack5.0.2
- JetPack5.0.1
- JetPack4.6
- JetPack4.5
- JetPack4.4 (deprecated，该版本SDK会在未来某个版本移除，请切换至新版本JetPack)
- JetPack4.2.2 (已移除，请切换至新版本JetPack)

安装JetPack时请务必安装对应的组件：

- 使用SDK Manager安装JetPack需要勾选TensorRT、OpenCV、CUDA、cuDNN等选项。
- 使用SD Card Image方式（仅对Jetson Nano和Jetson Xavier NX有效）则无需关心组件问题，默认会全部安装。

**Release Notes** | 时间 | 版本 | 说明 | | --- | --- | --- | | 2022.12.29 | 1.7.2 | 新增支持JetPack5.0.2；缓存机制优化；模型性能优化 | | 2022.07.28 | 1.6.0 | 新增支持JetPack5.0.1，新增目标追踪接入实时流的demo | | 2022.05.18 | 1.5.0 | 部分模型切换格式，max\_batch\_size含义变更，由输入图片数不大于该值变更为等于该值；移除适用于JetPack4.2.2的SDK；示例代码demo\_stream\_inference重构；示例代码移除frame\_buffer，新增更安全高效的safe\_queue | | 2021.12.22 | 1.3.5 | 新增支持JetPack4.6；支持在EasyEdge平台语义分割模型生成开发套件；修复缓存问题；支持自定义缓存路径 | | 2021.10.20 | 1.3.4 | 新增支持JetPack4.5；大幅提升EasyDL有损压缩加速模型的推理速度 | | 2021.06.29 | 1.3.1 | 视频流支持分辨率调整；支持将预测后的视频推流，新增推流demo | | 2021.05.13 | 1.3.0 | 新增视频流接入支持；EasyDL模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.1 | EasyEdge新增一系列模型的支持；性能优化 | | 2021.01.27 | 1.1.0 | EasyDL经典版高性能分类模型升级；

EasyDL经典版检测模型新增均衡选项；

EasyEdge平台新增Jetson系列端计算组件的生成；

问题修复 | | 2020.12.18 | 1.0.0 | 接口升级和一些性能优化 | | 2020.08.11 | 0.5.5 | 部分模型预测速度提升 | | 2020.06.23 | 0.5.4 | 支持JetPack4.4DP，支持EasyDL专业版更多模型 | | 2020.05.15 | 0.5.3 | 专项硬件适配SDK支持Jetson系列 |

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE 含义变更。变更前：预测输入图片数不大于该值均可。变更后：预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理，在输入图片数小于该值时依然可正常运行，但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值，尽可能保持最小。 【版本移除】 适用于JetPack4.4版本的SDK被标记为deprecated，SDK会在未来某个版本移除，建议切换至最新版本JetPack。适用于JetPack4.2.2版本的SDK被移除。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

2021-10-20: 【版本移除】适用于JetPack4.2.2版本的SDK被标记为deprecated，该版本代码已停止更新，SDK会在未来某个版本移除，请切换至新版本JetPack

**快速开始 安装依赖** 本SDK适用于JetPack4.5、JetPack4.6、JetPack5.0系列版本，请务必安装其中之一版本，并使用对应版本的SDK。注意在安装JetPack时，需同时安装CUDA、cuDNN、OpenCV、TensorRT等组件。

如已安装JetPack需要查询相关版本信息，请参考下文中的开发板信息查询与设置。

### 使用序列号激活

首先在官网获取序列号。



图片加载失败

将获取到的序列号填写到demo文件中或以参数形式传入。



图片加载失败

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定CONTROLLER\_KEY\_AUTH\_MODE为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

**编译并运行Demo** 模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

编译运行：

```
cd src
mkdir build && cd build
cmake ..
make -j$(nproc)
**make install 为可选，也可将lib所在路径添加为环境变量**
sudo make install
sudo ldconfig
./demo_batch_inference/easyedge_batch_inference {模型RES文件夹} {测试图片路径或仅包含图片的文件夹路径} {序列号}
```

demo运行示例：

```
baidu@nano:~/ljay/easydl/sdk/demo/build$ ./demo_batch_inference/easyedge_batch_inference ../../../../RES/
/ljay/images/mix008.jpeg
2020-08-06 20:56:30,665 INFO [EasyEdge] 548125646864 Compiling model for fast inference, this may take a while (Acceleration)
2020-08-06 20:57:58,427 INFO [EasyEdge] 548125646864 Optimized model saved to:
/home/baidu/.baidu/easyedge/jetson/mcache/24110044320/m_cache, Don't remove it
Results of image /ljay/images/mix008.jpeg:
2, kiwi, p:0.997594 loc: 0.352087, 0.56119, 0.625748, 0.868399
2, kiwi, p:0.993221 loc: 0.45789, 0.0730294, 0.73641, 0.399429
2, kiwi, p:0.992884 loc: 0.156876, 0.0598725, 0.3802, 0.394706
1, tomato, p:0.992125 loc: 0.523592, 0.389156, 0.657738, 0.548069
1, tomato, p:0.991821 loc: 0.665461, 0.419503, 0.805282, 0.573558
1, tomato, p:0.989883 loc: 0.297427, 0.439999, 0.432197, 0.59325
1, tomato, p:0.981654 loc: 0.383444, 0.248203, 0.506606, 0.400926
1, tomato, p:0.971682 loc: 0.183775, 0.556587, 0.286996, 0.711361
1, tomato, p:0.968722 loc: 0.379391, 0.0386965, 0.51672, 0.209681
Done
```

检测结果展示：



## 测试Demo HTTP 服务

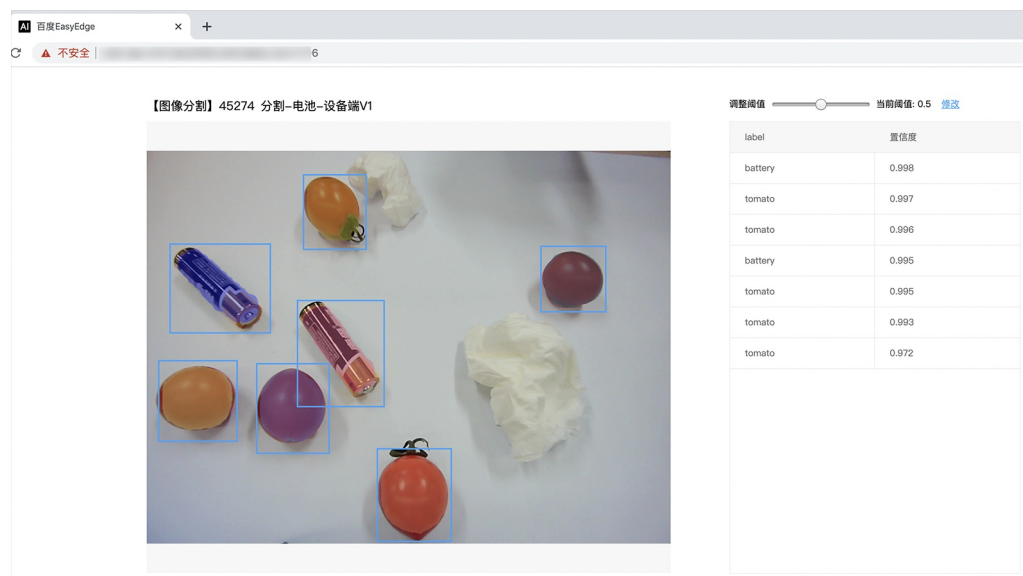
编译demo完成之后，会同时生成一个http服务，运行

```
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型运行参数
EdgePredictorConfig config;
config.model_dir = model_dir;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, serial_num);
config.set_config(params::PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE, 1); // 优化的模型可以支持的batch_size
config.set_config(params::PREDICTOR_KEY_GTURBO_FP16, false); // 置true开启fp16模式推理会更快, 精度会略微降低, 但取决于硬件是否支持fp16, 不是所有模型都支持fp16, 参阅文档
config.set_config(params::PREDICTOR_KEY_GTURBO_COMPILE_LEVEL, 1); // 编译模型的策略, 如果当前设置的max_batch_size与历史编译存储的不同, 则重新编译模型

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

### 初始化接口

```

auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

```

若返回非0, 请查看输出日志排查错误原因。

### 预测接口

```
/**
 * @brief
 * 单图预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& results
) = 0;

/**
 * @brief
 * 批量图片预测接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `max_batch_size`，其含义见下方参数配置接口的介绍。

**参数配置接口** 参数配置通过结构体EdgePredictorConfig完成。

```

struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};

```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

针对Jetson开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）

```



```

* 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
* 值类型: int
* 默认值: 1
*/
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产出的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时内存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的 GPU 卡号，对于 Jetson，此值无需更改。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 max\_batch\_size 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 compile\_level 来控制，当此值为 0 时，表示忽略当前设置的 max\_batch\_size 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 max\_batch\_size 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference。

**PREDICTOR\_KEY\_GTURBO\_FP16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持 fp16 的模式包括：EasyDL 图像分类高精度模型。

## 预测视频接口

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

class `VideoDecoding` :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct `VideoConfig`

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};          // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;            // frame存储为视频文件的路径
    bool save_all{false};             // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

#### 返回格式

预测成功后，从 `EdgeResultData`中 可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测或图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 图片宽度 = 检测框的左上角的横坐标 y1 图片高度 = 检测框的左上角的纵坐标 x2 图片宽度 = 检测框的右下角的横坐标 y2 图片高度 = 检测框的右下角的纵坐标

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里, 图片的解码运行在cpu之上, 可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量, 根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

### 2. 请求http服务

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片来进行测试。

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供, 则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img.json())
```

Java请求示例参考[这里](#)

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考接口使用-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

**多线程预测** Jetson 系列 SDK 支持多线程预测, 创建一个 predictor, 并通过 PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY 控制所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。需要注意的是多线程的启用会随着线程数的增加而降低单次 infer 的推理速度, 建议优先使用 batch inference 或权衡考虑使用。

#### 已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时, 部分结果错误

A: EasyDL图像分类高精度模型在有些显卡上可能存在此问题, 可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

#### 2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object

A: 如果遇到此问题, 请确认没有频繁调用 init 接口, 通常调用 infer 接口即可满足需求。

#### 3. 开启 fp16 后, 预测结果错误

A: 不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括: EasyDL图像分类高精度模型。目前不支持的将会在后面的版本陆续支持。

#### 4. 部分模型不支持序列化

A: 针对JetPack4.4、4.5版本, 部分模型无法使用序列化, 如已知的BML的MobileNetV1-SSD和物体检测高性能模型。需要每次加载模型的时候

编译模型，过程会比较慢。此问题将在后续JetPack版本中修复。目前JetPack4.6版本SDK已修复该问题。

**开发板信息查询与设置 查询L4T或JetPack版本** 查询JetPack版本信息，可以通过下面这条命令先查询L4T的版本。

```
**在终端输入如下命令并回车**
$ head -n 1 /etc/nv_tegra_release
**就会输出类似如下结果**
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看，板子当前的L4T版本为R32.4.3，对应JetPack4.4。注意，L4T的版本不是JetPack的版本，一般可以从L4T的版本唯一对应到JetPack的版本，下面列出了最近几个版本的对应关系：

```
L4T R32.6.1 --> JetPack4.6
L4T R32.5.1 --> JetPack4.5.1
L4T R32.5 --> JetPack4.5
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下，执行AI推理的速度是不一样的，如果对速度需求很高，可以把功率开到最大，但记得加上小风扇散热~

```
**1. 运行下面这条命令可以查询开发板当前的运行功率模式**
$ sudo nvpmode -q verbose
**$ NV Power Mode: MAXN**
**$ 0**
**如果输出为MAXN代表是最大功率模式**

**2. 若需要把功率调到最大，运行下面这条命令**
$ sudo nvpmode -m 0

**如果你进入了桌面系统，也可以在桌面右上角有个按钮可以切换模式**

**3. 查询资源利用率**
$ sudo tegrastats
```

#### FAQ 1. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**2. 运行SDK报错 Authorization failed 日志显示 Http perform failed: null respond** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

#### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

#### 4. 运行demo时报找不到libeasyedge\_extension.so

需要export libeasyedge\_extension.so所在的路径，如路径为/home/work/baidu/cpp/lib，则需执行：

```
export LD_LIBRARY_PATH=/home/work/baidu/cpp/lib:${LD_LIBRARY_PATH}
```

或者在编译完后执行如下命令将lib文件安装到系统路径：

```
sudo make install
```

如不能安装，也可手动复制lib下的文件到/usr/local/lib下。

## 5. 运行demo时报如下之一错误

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Compiling model for fast inference, this may take a while (Acceleration) Killed
```

**\*\*或\*\***

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Build graph failed
```

请适当降低PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE和PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY的值后尝试。

**6. 运行有损压缩加速的模型，运算精度较标准模型偏低** 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

## 🔗 图像分类辨影专用SDK集成文档

### 简介

本文档介绍EasyEdge/EasyDL的辨影软硬一体方案SDK的使用方法。支持的硬件包括辨影Air、辨影Pro。您可以在[软硬一体方案](#)了解部署方案。

### 模型支持：

- EasyDL图像：图像分类高精度，图像分类高性能，物体检测高精度，物体检测均衡，物体检测高性能
- BML：
  - 公开数据集预训练模型：SSD-MobileNetV1，YOLOv3-DarkNet，YOLOv3-MobileNetV1，ResNet50，ResNet101，SE-ResNeXt50，SE-ResNeXt101，MobileNetV2，EfficientNetB0\_small，EfficientNetB4，MobileNetV3\_large\_x1\_0，ResNet18\_vd，SE\_ResNet18\_vd，Xception71。
  - 百度超大规模数据集预训练模型：YOLOv3-DarkNet，MobileNetV3\_large\_x1\_0，ResNet50\_vd，ResNet101\_vd。
- EasyEdge：EasyEdge支持的模型较多，详见[查看模型网络适配硬件](#)。若模型不在此列表，可以尝试使用自定义网络生成端计算组件。

Release Notes | 时间 | 版本 | 说明 | |---| |---| |---| | 2022.08.01 | 1.3.5 | 新增支持辨影软硬一体方案部署 |

**辨影软件接入使用SDK** 辨影Air/Pro自带软件预置了大量飞桨开源模型，支持EasyDL/BML模型SDK一键导入使用，详细的辨影使用说明见购买后获得的使用说明书

- 辨影推理主界面



- 辨影设置界面。在应用中可选预置模型能力，也可选择EasyDL/BML导入的模型SDK



快速开始 接下来的文档内容将会描述辨影SDK的集成开发教程，仅需要使用辨影自带软件的用户无需关注

使用序列号激活



首先请在[EasyDL智能云官网](#)获取序列号。

将获取到的序列号填写到demo文件中或以参数形式传入。



默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定CONTROLLER\_KEY\_AUTH\_MODE为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

**编译并运行Demo** 模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

编译运行：

```
cd src
mkdir build && cd build
cmake ..
make
**make install 为可选，也可将lib所在路径添加为环境变量**
sudo make install
sudo ldconfig
./demo_batch_inference/easyedge_batch_inference {模型RES文件夹} {测试图片路径或仅包含图片的文件夹路径} {序列号}
```

demo运行示例：

```
baidu@nano:~/ljay/easydl/sdk/demo/build$ ./demo_batch_inference/easyedge_batch_inference ../../../../RES/
/ljay/images/mix008.jpeg
2020-08-06 20:56:30,665 INFO [EasyEdge] 548125646864 Compiling model for fast inference, this may take a while (Acceleration)
2020-08-06 20:57:58,427 INFO [EasyEdge] 548125646864 Optimized model saved to:
/home/baidu/.baidu/easyedge/jetson/mcache/24110044320/m_cache, Don't remove it
Results of image /ljay/images/mix008.jpeg:
2, kiwi, p:0.997594 loc: 0.352087, 0.56119, 0.625748, 0.868399
2, kiwi, p:0.993221 loc: 0.45789, 0.0730294, 0.73641, 0.399429
2, kiwi, p:0.992884 loc: 0.156876, 0.0598725, 0.3802, 0.394706
1, tomato, p:0.992125 loc: 0.523592, 0.389156, 0.657738, 0.548069
1, tomato, p:0.991821 loc: 0.665461, 0.419503, 0.805282, 0.573558
1, tomato, p:0.989883 loc: 0.297427, 0.439999, 0.432197, 0.59325
1, tomato, p:0.981654 loc: 0.383444, 0.248203, 0.506606, 0.400926
1, tomato, p:0.971682 loc: 0.183775, 0.556587, 0.286996, 0.711361
1, tomato, p:0.968722 loc: 0.379391, 0.0386965, 0.51672, 0.209681
Done
```

检测结果展示：



## 测试Demo HTTP 服务

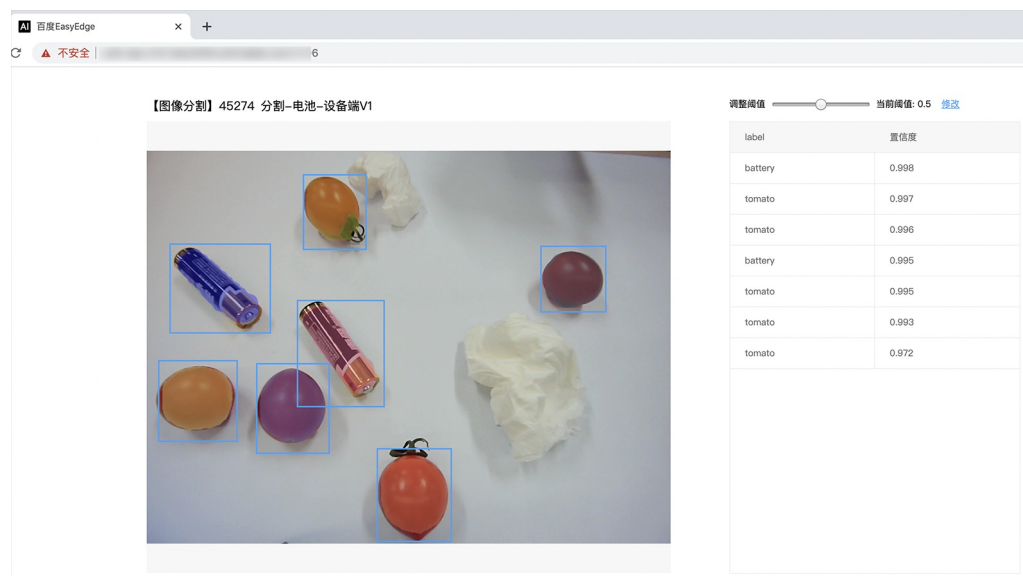
编译demo完成之后，会同时生成一个http服务，运行

```
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型运行参数
EdgePredictorConfig config;
config.model_dir = model_dir;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, serial_num);
config.set_config(params::PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE, 1); // 优化的模型可以支持的最大batch_size，实际单次推理的图片数不能大于此值
config.set_config(params::PREDICTOR_KEY_GTURBO_FP16, false); // 置true开启fp16模式推理会更快，精度会略微降低，但取决于硬件是否支持fp16，不是所有模型都支持fp16，参阅文档
config.set_config(params::PREDICTOR_KEY_GTURBO_COMPILE_LEVEL, 1); // 编译模型的策略，如果当前设置的max_batch_size与历史编译存储的不同，则重新编译模型

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame，需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频，需在video_config中开启配置
}

```

### 初始化接口

```

auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

```

若返回非0，请查看输出日志排查错误原因。

### 预测接口

```
/**
 * @brief
 * 单图预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& results
) = 0;

/**
 * @brief
 * 批量图片预测接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `max_batch_size`，其含义见下方参数配置接口的介绍。

**参数配置接口** 参数配置通过结构体EdgePredictorConfig完成。

```

struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};

```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

针对Jetson开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）

```

```

* 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
* 值类型: int
* 默认值: 1
*/
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产出的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数不可大于此值，但可以是不大于此值的任意图片数。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 max\_batch\_size 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 compile\_level 来控制，当此值为 0 时，表示忽略当前设置的 max\_batch\_size 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 max\_batch\_size 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference。

**PREDICTOR\_KEY\_GTURBO\_FP16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持fp16的模式包括：EasyDL图像分类高精度模型。

## 预测视频接口

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 VideoDecoding，此类提供了获取视频帧数据的便利函数。通过 VideoConfig 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK infer 接口的参数进行预测。

class VideoDecoding :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};         // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;           // frame存储为视频文件的路径
    bool save_all{false};            // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于`/dev/video0`的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

#### 返回格式

预测成功后，从 `EdgeResultData`中可以获得对应的分类信息、位置信息。



```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测或图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 图片宽度 = 检测框的左上角的横坐标 y1 图片高度 = 检测框的左上角的纵坐标 x2 图片宽度 = 检测框的右下角的横坐标 y2 图片高度 = 检测框的右下角的纵坐标

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里, 图片的解码运行在cpu之上, 可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量, 根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

### 2. 请求http服务

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片来进行测试。

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供, 则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Java请求示例参考[这里](#)

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考接口使用-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

**多线程预测** 辨影系列 SDK 支持多线程预测, 创建一个 predictor, 并通过 PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY 控制所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。需要注意的是多线程的启用会随着线程数的增加而降低单次 infer 的推理速度, 建议优先使用 batch inference 或权衡考虑使用。

#### 已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时, 部分结果错误

A: EasyDL图像分类高精度模型在有些显卡上可能存在此问题, 可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

#### 2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object

A: 如果遇到此问题, 请确认没有频繁调用 init 接口, 通常调用 infer 接口即可满足需求。

#### 3. 开启 fp16 后, 预测结果错误

A: 不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括: EasyDL图像分类高精度模型。目前不支持的将会在后面的版本陆续支持。

#### 4. 部分模型不支持序列化

A: 针对JetPack4.4版本, 部分模型无法使用序列化, 如已知的BML的MobileNetV1-SSD和物体检测高性能模型。需要每次加载模型的时候编译模

型，过程会比较慢。此问题将在后续JetPack版本中修复。

**开发板信息查询与设置 查询L4T或JetPack版本** 查询JetPack版本信息，可以通过下面这条命令先查询L4T的版本。

```
**在终端输入如下命令并回车**
$ head -n 1 /etc/nv_tegra_release
**就会输出类似如下结果**
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看，板子当前的L4T版本为R32.4.3，对应JetPack4.4。注意，L4T的版本不是JetPack的版本，一般可以从L4T的版本唯一对应到JetPack的版本，下面列出了最近几个版本的对应关系：

```
L4T R32.6.1 --> JetPack4.6
L4T R32.5.1 --> JetPack4.5.1
L4T R32.5 --> JetPack4.5
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下，执行AI推理的速度是不一样的，如果对速度需求很高，可以把功率开到最大，但记得加上小风扇散热~

```
**1. 运行下面这条命令可以查询开发板当前的运行功率模式**
$ sudo nvpmode -q verbose
**$ NV Power Mode: MAXN**
**$ 0**
**如果输出为MAXN代表是最大功率模式**

**2. 若需要把功率调到最大，运行下面这条命令**
$ sudo nvpmode -m 0

**如果你进入了桌面系统，也可以在桌面右上角有个按钮可以切换模式**

**3. 查询资源利用率**
$ sudo tegrastats
```

#### FAQ 1. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

#### 2. 运行SDK报错 Authorization failed 日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

#### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

#### 4. 运行demo时报找不到libeasyedge\_extension.so

需要export libeasyedge\_extension.so所在的路径，如路径为/home/work/baidu/cpp/lib，则需执行：

```
export LD_LIBRARY_PATH=/home/work/baidu/cpp/lib:${LD_LIBRARY_PATH}
```

或者在编译完后执行如下命令将lib文件安装到系统路径：

```
sudo make install
```

如不能安装，也可手动复制lib下的文件到/usr/local/lib下。

## 5. 运行demo时报如下之一错误

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Compiling model for fast inference, this may take a while (Acceleration) Killed
**或**
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Build graph failed
```

请适当降低PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE和PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY的值后尝试。

**6. 运行有损压缩加速的模型，运算精度较标准模型偏低** 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

## 端云协同部署

### 端云协同服务说明

#### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 联网状态下在平台管理设备运行状态、资源利用率

目前本地服务器的应用平台支持Linux-AMD64(x86-64)，具体使用流程请参考下方文档。

#### 使用流程

#### Step 1 发布端云协同部署包

在[我的部署包](#)页面点击「发布端云协同部署包」

**图像分类模型** [操作文档](#) [教学视频](#) [常见问题](#) [提交工单](#) 收起 ^

定制图像分类模型，可以识别一张图整体是什么物体/状态/场景。  
在各分类图片之间差异明显的情况下，训练数据每类仅需20-100张，最快10分钟可训练完毕

模型中心

我的模型

- 创建模型
- 训练模型
- 校验模型
- 发布模型

EasyData数据服务

- 数据总览
- 标签组管理
- 在线标注
- 云服务数据回流

EasyEdge本地部署

- 纯离线服务
- 端云协同服务

**我的部署包**

- 我的本地设备
- 下发部署包到设备

**端云协同服务 > 我的部署包** 点击收起

**端云协同服务说明**

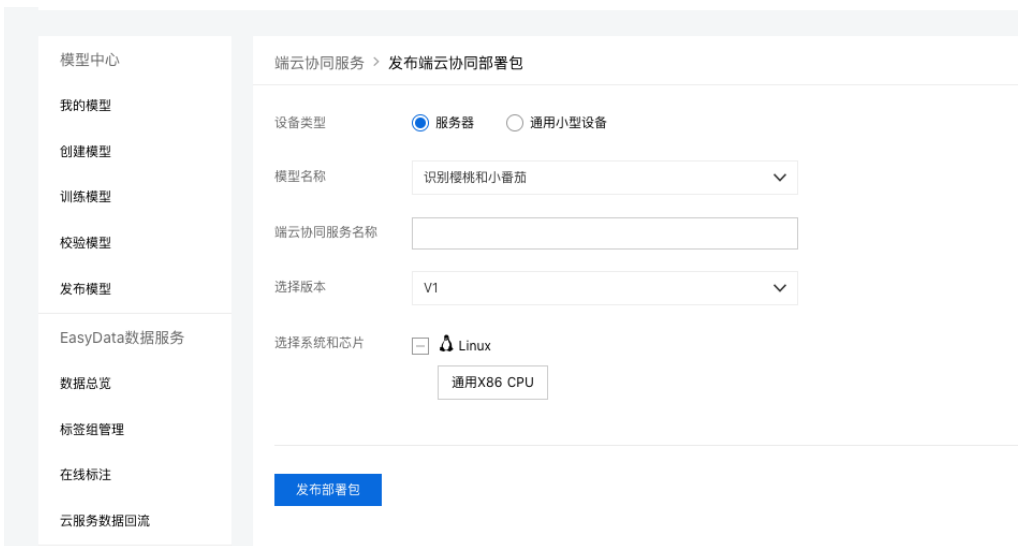
- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「[我的本地设备](#)」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「[下发部署包到设备](#)」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
暂无可用数据 请稍后再试							

填写服务名称，选择模型版本并提交发布



在列表查看部署包发布状态



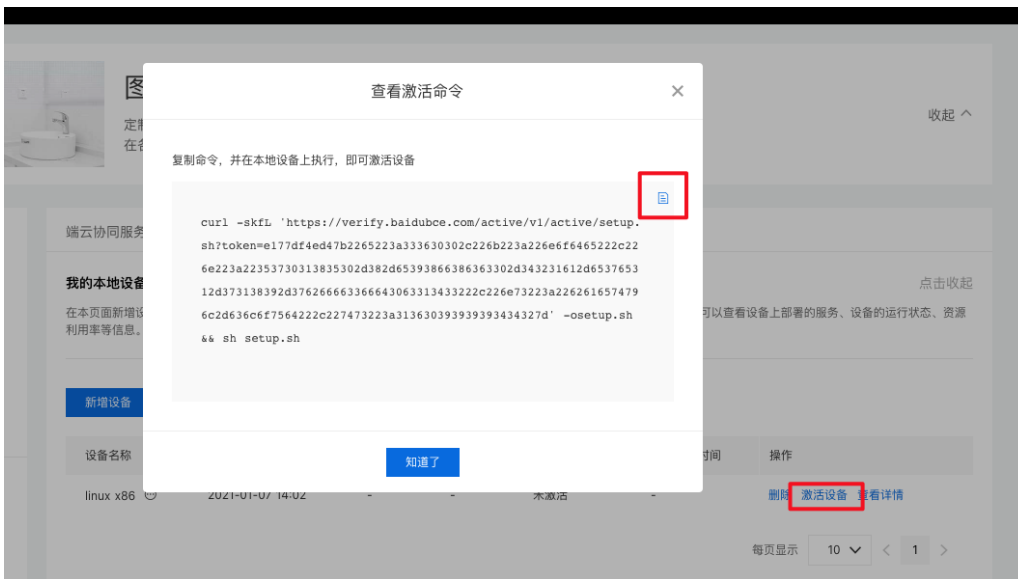
## Step 2 新增设备并激活

在**我的本地设备**页面新增设备





在列表中，点击设备对应的「激活设备」操作，复制激活命令并在本地设备上执行即可

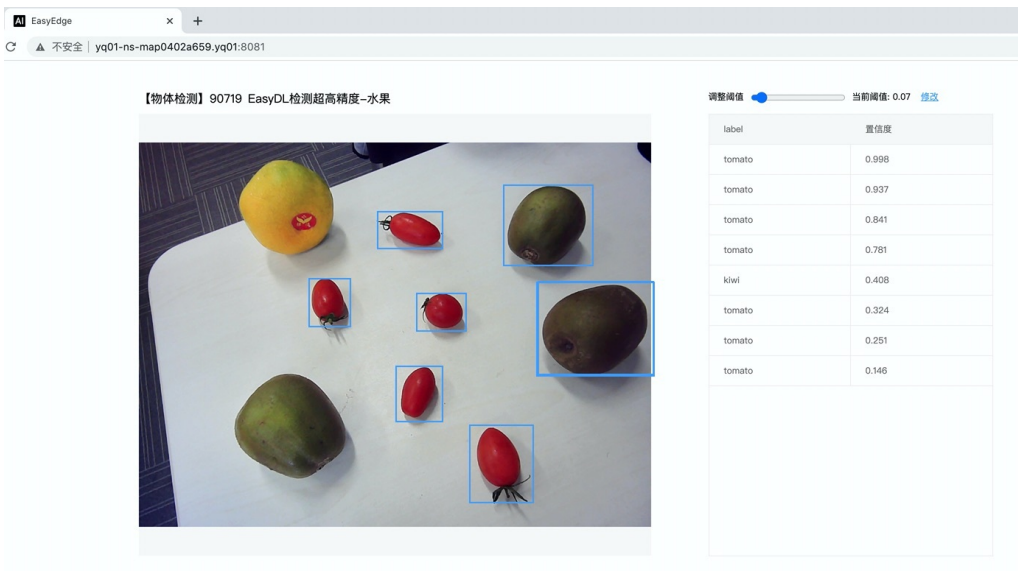


### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入http://{设备ip}:{服务端点，默认8080}，即可预览效果：



具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

### 云端管理说明

### 模型部署包管理

在[我的部署包](#)页面可以进行已发布的模型部署包的管理。

### 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。



新版本发布成功后，即可在「下发部署包到设备」页面或当前服务的「服务详情」页面，将新版本下发到本地设备上。



## 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。



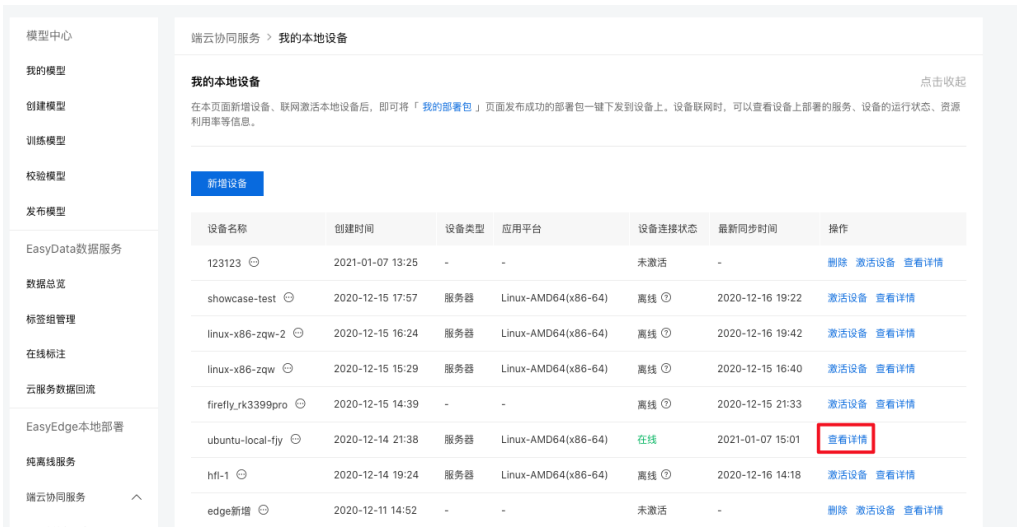
## 本地设备管理

在[我的本地设备](#)页面可以进行所有本地设备的管理。

## 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：





设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况



**申请延期** 端云协同部署包有效期为1个月，到期后可点击申请延期，工作人员审核后会增加测试时间



### 端云协同-JetsonNano部署文档

**端云协同-Jetson系列硬件** 端云协同支持的 Jetson 系列硬件包括 Jetson nano , Jetson TX2 , Jetson AGX Xavier 和 Jetson Xavier NX.

#### 准备环境

- JetPack** : 安装JetPack 4.4 版本 (目前端云协同仅支持 JetPack4.4 版本) , 并注意在安装 JetPack 时务必安装对应的组件 :
  - 使用 SDK Manager 安装 JetPack 需要勾选 TensorRT、OpenCV、CUDA、cuDNN 等选项。
  - 使用 SD Card Image 方式 (仅对 Jetson Nano 和 Jetson Xavier NX 有效) 则无需关心组件问题, 默认会全部安装。
- docker** : 安装 nvidia-docker 或 docker (版本 >= 19.03) , 一般 Jetson 系列硬件自带操作系统都已包含 nvidia-docker 或 docker , 可通过以下命令确认 :
 

```

// nvidia-docker
nvidia-docker version
// docker

```

`docker version` 修改启动参数 `/etc/docker/daemon.json` 文件, 将`"default-runtime"`改为`"nvidia"`, 并重启 `nvidia-docker` 或 `docker`。

```

cat baidu@xavier-nx:~$ cat /etc/docker/daemon.json
{
  "runtimes": {
    "nvidia": {
      "path": "nvidia-container-runtime",
      "runtimeArgs": []
    }
  },
  "insecure-repositories": [
    "localhost:5000",
    "172.17.0.1:5000"
  ],
  "experimental": true,
  "default-runtime": "nvidia"
}

```

3. 依赖库文件: 下载文件 `easyedge_runtime_j44.csv`, 并将该文件置于 Jetson 宿主机的 `/etc/nvidia-container-runtime/host-files-for-container.d/` 目录内。

激活设备

- 1. 在 EasyDL/BML/EasyEdge 平台「端云协同服务」-「我的本地设备」页面新增设备。
- 2. 在设备列表中点击设备对应的「激活设备」操作, 复制激活命令。
- 3. 在 Jetson 设备上, 执行激活命令。激活过程中提示选择 `containerd` 或者 `docker` 时, 选择 `docker`, 如下图:

```

Hitool@fire2:~/zlh/tmp$ curl -s -k https://verify.baodubce.com/active/v1/active/setup.sh?token=e285ed51007b2265223a33363030c2c226b223a226e0f6465222c226e223a2235383734372d312d513131373226693261392d523061306252363437376133639222c226e7322602261657479962d61366f756422c227473223a313633323838363430337d -o setup.sh && sh setup.sh
K8S/K3S is not installed yet, do you want us to install K3S for you? Yes/No (default: Yes):Yes
K3S could run with containerd/docker, which do you want us to install for you? containerd for Yes, docker for No (default: Yes):No

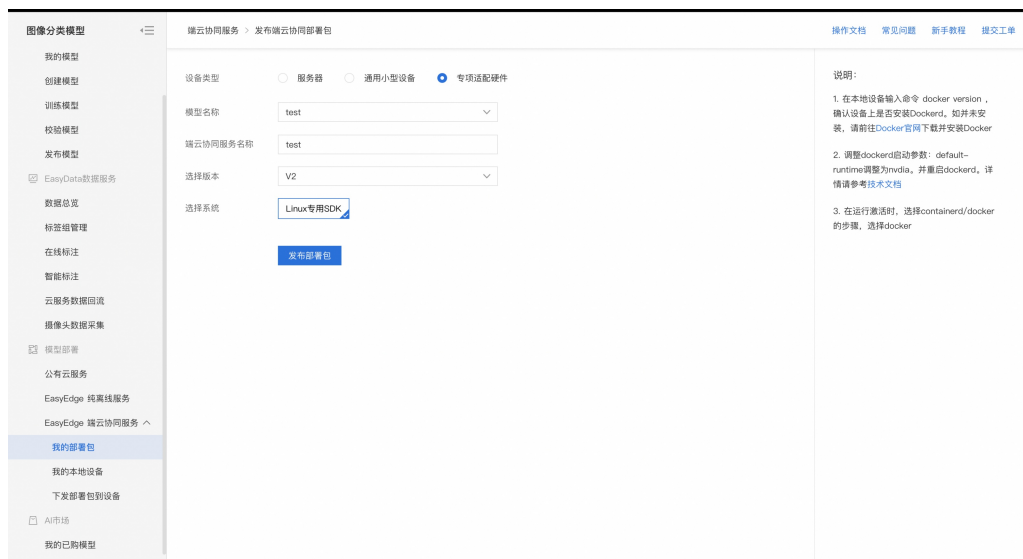
```

下发部署

1. 在端云协同服务-我的部署包页面点击发布端云协同部署包



2. 选择模型名称和版本, 发布为专项适配硬件部署包



3. 点击导航栏中的下发部署包到设备，即可将刚才生成的部署包下发到已经激活的设备当中

🔗 浏览器或小程序部署

🔗 浏览器或小程序部署

### 浏览器或小程序部署

简介 本文档介绍EasyDL的浏览器/小程序部署SDK的使用方法，

#### SDK支持范围 浏览器部署

PC浏览器: Chrome、Safari、Firefox

手机浏览器: Baidu App、Safari、Chrome、UC and QQ Browser

#### 小程序部署

小程序: 百度小程序、微信小程序

#### 支持的操作系统

系统: MacOS、Windows

demo文件结构 SDK解压缩之后，目录结构如下

```
|--public
| |--model
| |--model.json
| |--chunk_n.dat
|--src
| |--components
| |--App.vue
| |--config.json
| |--env.d.ts
| |--label.json
| |--main.ts
| |--modelInfo.json
| |--usePredict.ts
|--index.html
|--package.json
|--README.md
|--tsconfig.json
|--tsconfig.node.json
|--vite.config.ts
|--yarn.lock
```

demo基于vite，其中public/model下的model.json、chunk\_1.dat...chunk\_n.dat为模型文件，src下为业务代码，index.html为入口文件

快速开始 依赖node及npm，如果没有node，请前往[node官网](#)下载长期维护版本

安装依赖：npm install

启动项目：npm run dev

启动后控制台输出

```
vite v2.8.4 dev server running at:  
> Local: http://localhost:3000/  
> Network: use `--host` to expose
```

到浏览器打开 <http://localhost:3000/> 即可体验demo

**模型预测结果示例** 图像分类示例

```
[0.4450492858886719, 0.3961234986782074, 0.0122891990467906, 0.14653800427913666]
```

数组的index为对应的标签，值为置信度

物体检测示例

```
[[1, 0.2247152328491211, 0.11200979351997375, 0.07523892819881439, 0.8540866374969482, 0.5503567457199097], [2, 0.1224712328491211, 0.511200979351997375, 0.27523892819881439, 0.8540866374969482, 0.5503567457199097],...]
```

输出结果是一个二维数组，第二维的结果为：[标签，置信度，矩形框x1坐标，矩形框y1坐标，矩形框x2坐标，矩形框y2坐标]

**浏览器开发**

参考src/usePredict文件

```
// 加载推理引擎  
import {Runner, env} from '@paddlejs/paddlejs-core';  
// 使用webgl计算方案(暂不能使用wasm、webgpu等计算方案)  
import '@paddlejs/paddlejs-backend-webgl';  
...  
// 注册引擎  
const runner = new Runner({  
  modelPath: '/model',  
  keepRatio: config.rescale_mode === 'keep_ratio',  
  mean: config.img_mean.reduce((memo, v) => [...memo, +(v / 255).toFixed(3)]), [] as number[]),  
  std: config.scale.reduce((memo, v) => [...memo, +((1 / 255 / v).toFixed(3))], [] as number[]),  
  bgr: config.colorFormat === 'BGR',  
  feedShape: {  
    fw: config.resize[0],  
    fh: config.resize[1]  
  }  
});  
...  
// init runner  
await runner.init();  
...  
// predict and get result  
await runner.predict(img);
```

更多可参考[PaddleJS工程页](#)

**小程序开发**

**微信小程序**

微信小程序需添加 [Paddle.js微信小程序插件](#)

步骤：

小程序管理界面 --> 设置 --> 第三方设置 --> 插件管理 --> 添加插件 --> 搜索 wx7138a7bb793608c3 并添加

**掌上百度小程序**

手百小程序需添加paddlejs百度智能小程序动态库 **引入动态库代码包**

代码示例：

```
{
  "dynamicLib": {
    // 定义一个别名，小程序中用这个别名引用动态库。
    "paddlejs": {
      "provider": "paddlejs"
    }
  }
}
```

### 使用动态库

在使用页面的json文件里配置如下信息：

```
{
  "usingSwanComponents": {
    "paddlejs": "dynamicLib://paddlejs/paddlejs"
  }
}
```

从而页面中可以使用此组件：

```
<view class="container">
  <view>下面这个自定义组件来自于动态库</view>
  <paddlejs />
</view>
```

### 示例

index.swan

```
<view class="container">
  <!-- index.wxml -->
  <image style="width:100%; height: 300px; " src="{{imgPath}}"></image>
  <button bindtap="chooseImage">选择图片</button>
  <button bindtap="doPredict" class="btn" type="primary">新鲜度预测</button>
  <!-- 返回结果 -->
  <view class="result" s-if="resultType">预测结果：{{resultType}}</view>
  <view class="result" s-if="resultVal">预测可信用度：{{resultVal}}</view>
  <paddlejs options="{{options}}" status="{{status}}" imgBase64="{{imgBase64}}" bindchildmessage="predict" />
</view>
```

index.js

```
Page({
  data: {
    imgPath: '',
    content: '',
    resultType: '',
    resultVal: '',
    isShow: true,
    options: { // 模型配置项
      modelPath: 'http://localhost:3000/model',
      fileCount: 3,
      needPreheat: true,
      feedShape: {
        fw: 224,
        fh: 224
      },
    },
    fetchShape: [1, 7, 1, 1],
    fill: [255, 255, 255, 255],
    scale: 256,
    targetSize: { height: 224, width: 224 },
    mean: [0.485, 0.456, 0.406],
    std: [0.229, 0.224, 0.225]
  },
  status: '' // 初始值为'', 变为'predict'时会触发模型预测
},
```

```

/**
 * 选择图片
 */
chooseImage: function () {
  const me = this;
  this.setData({
    ishow: false
  });
  swan.chooseImage({
    count: 1,
    sizeType: ['original', 'compressed'],
    sourceType: ['album', 'camera'],
    success(res) {
      const path = res.tempFilePaths[0];
      swan.getFileSystemManager().readFile({
        filePath: path,
        encoding: 'base64',
        success: res => {
          me.setData({
            imgBase64: res && res.data,
            imgPath: path
          });
        },
        fail: res => {
          console.log(res);
        }
      });
    }
  });
},
predict(e) {
  const status = e && e.detail && e.detail.status;
  if (status === 'loaded') {
    this.setData({status: 'loaded', isShow: false});
  }
  else if (status === 'complete') {
    const data = e.detail.data;
    const maxItem = this.getMaxItem(data);
    this.setData({status: '', resultType: maps[maxItem.index], resultVal: maxItem.value});
  }
},
doPredict() {
  this.setData({status: 'predict'});
},
getMaxItem(datas = []) {
  let max = Math.max.apply(null, datas);
  let index = datas.indexOf(max);
  return {value: max, index};
},
});

```

### Prop

名称	类型	默认值	是否必选	描述
options	string		是	模型配置项，参考src/usePrdict
imgBase64	string		是	要预测的图像的base64
status	string	"	是	当前状态，status变化触发组件调用相应的api，当status变为predict时，组件会读取imgBase64作为输入的图像，调用模型预测API

### 🔗 模型加速整体说明

**功能简介** 当您发布时纯离线服务时，平台已结合最新的量化、剪枝、蒸馏技术，推出丰富的模型压缩加速方案，以提高您的SDK部署效率。

**覆盖范围：**服务器、通用小型设备、专项适配硬件均支持该功能。

具体原理：针对目标芯片，对模型做深度优化压缩加速，加速后模型在推理速度、内存占用、体积大小等指标上表现更优。发布加速模型可能需要一段时间，同时会有微小的精度损失。发布完成后可通过性能报告对比具体加速效果。

**使用流程** 选择加速方式 结合选择的系统与芯片不同，分别为您提供不同的压缩方式。

纯云服务器 > 发布新服务

操作文档 教学视频 常见问题 提交工单

服务器 通用小型设备 专项适配硬件

1 选择部署形式 2 填写个人信息

集成方式  SDK  API

选择模型 dog-cat-test

选择版本 V2

选择系统和芯片  Linux  Windows

通用X86 CPU 英伟达GPU 华为 Atlas 300 百度 昆仑XPU

模型加速

基础-无加速  
无加速

精度无损压缩加速  
在精度尽可能无损的前提下加速模型

精度微损压缩加速-中  
在部分芯片上，内存/显存占用降低，推理速度可以获得一定提升

下一步

说明：

- 本地服务器部署支持将模型部署于本地的CPU、GPU服务器上，提供API和SDK两种集成方式：[查看文档](#)
- 本地服务器SDK：将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活
- 集成步骤：① 申请SDK并在服务详情页下载SDK → ② 在控制台申请激活序列号 → ③ 根据开发文档集成SDK，并联网激活使用。如存在设备无法联网，需要在纯离线的环境下激活的情况，请[提交工单](#)联系我们。
- 个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

提示：基础SDK默认作为勾选项存在，可后续与您的加速SDK进行效果与性能比对，方便您进一步挑选

**查看发布状态** 点击完成发布后，将自动跳转至列表页，可分别查看不同加速方案下的模型发布进度及发布时间。

服务器 通用小型设备 专项适配硬件

输入模型名称

SDK API

此处发布、下载的SDK为未授权SDK，需要前往控制台[获取序列号](#)激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	操作
dog-cat-test	115215-V2	通用X86 CPU-Linux	基础版	● 发布中	2021-05-13 20:49	下载SDK
			精度无损压缩加速	● 发布中	2021-05-13 20:49	下载加速版SDK

## 常见问题

### 数据相关问题

**需要上传多少张图片才能训练出效果较好的模型？**

- 每个分类至少需要准备20张以上。如果想要较好的效果，建议每个分类准备不少于100张图片。

**上传图片的总量有限制吗？**

- 每个账号下所有数据集的图片总数不能超过10万张。

### 训练相关问题

**数据处理失败或者状态异常怎么办？**

- 如是是图像分类模型上传处理失败，请先检查已上传的分类命名是否正确，是否存在中文命名、或者增加了空格；然后检查下数据图片量是否超过上限（10万张）；再检查图片中是否有损坏。如果自查没有发现问题，请在百度智能云控制台内[提交工单](#)反馈

**模型训练失败怎么办？**

- 如果遇到模型训练失败的情况，请先尝试重新训练，如多次重新训练后仍然失败，请在百度智能云控制台内[提交工单](#)反馈

**已经上线的模型还可以继续优化吗？**

- 已经上线的模型依然可以持续优化，操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练，然后在模型列表中更新线上服务，完成模型的优化

#### Step 1 重新训练

点击我的模型列表——找到需要重新训练的模型——点击训练，进行新版本模型训练

#### Step 2 重新发布

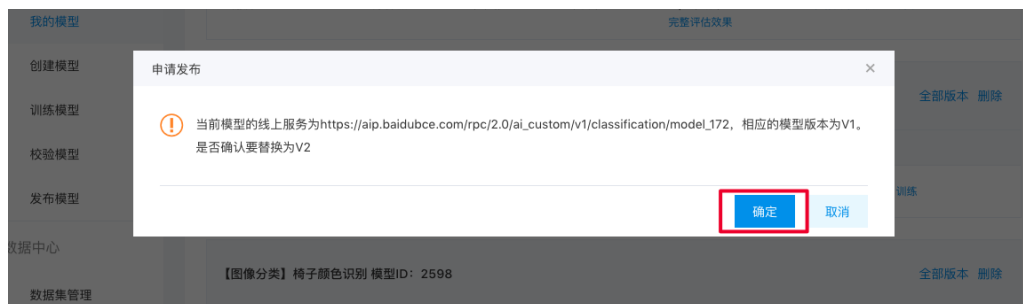
点击我的模型列表——找到新训练好的模型版本——点击申请发布

【图像分类】百美椅子训练 模型ID: 230						全部版本 删除
应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V2	训练完成	未申请	未发布	top1准确率87.61% top5准确率100.00% <a href="#">完整评估效果</a>	<b>申请发布</b> 校验 训练
离线SDK	V1	训练完成	未申请	未发布	top1准确率85.84% top5准确率100.00% <a href="#">完整评估效果</a>	申请发布 训练

每页显示 12 < 1 >

### Step 3 确认发布

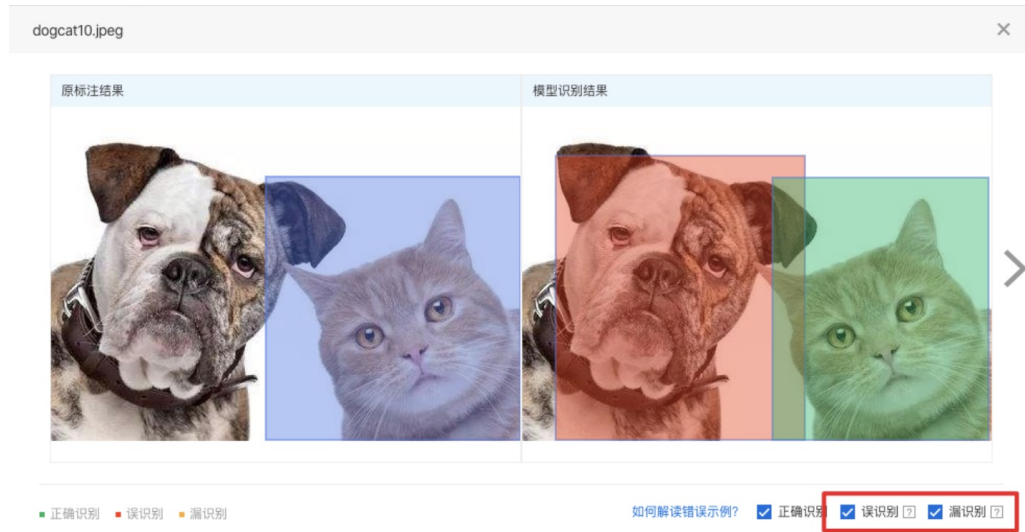
在出来的弹窗中点击确定



### 🔗 模型效果相关问题

如何通过「完整评估结果」里的错误示例优化模型？

- 错误示例中，左侧是正确的结果，右侧是模型的识别结果
- 观察模型识别有误的图片有哪些共同点，并有针对性地补充训练数据。比如：当图片比较亮的时候模型都能识别正确，但比较暗的时候模型就识别错了。这时就需要补充比较暗的图片作为训练数据



我的数据有限，如何优化效果？

- 在训练配置页面-数据增强策略中配置更多数据增强的算子，来增加训练数据。也可在精度提升配置包-数据增强策略中选择自动数据增强策略，从而自动补充适合场景的增强数据
- 如果您是通过将模型发布为公有云服务进行应用，即可通过云服务数据管理功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

实际调用服务时模型效果变差？

- 训练图片和实际场景要识别的图片拍摄环境应一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片
- 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 如果使用的是云服务，可以开通云服务数据管理功能，将实际调用云服务识别的图片加入训练集，不断迭代模型



\*\*如果训练数据已经达到以上要求，且单个分类/标签的图片量超过200张以上，效果仍然不佳，请在百度智能云控制台内[提交工单](#)反馈

## 模型上线/部署相关问题

每个账号可以上线几个模型？是否可以删除已上线的模型？

- 不限制发布模型数量，已上线模型无法删除

线上的部署方式支不支持我的硬件？

部署类型	支持的硬件示例
通用ARM	绝大多数安卓、苹果手机；瑞芯微RK32、RK32、RK35系列、树莓派等开发板
英特尔神经计算棒	NCS 1代、NCS 2代
海思NNIE	Hi3559AV100/Hi3559CV100等
华为昇腾Atlas开发板	Atlas200计算盒、Atlas300 计算卡
比特大陆SE计算盒	Bitmain SE5
通用x86CPU	绝大多数英特尔和AMD CPU
通用x86CPU加速版	英特尔志强、酷睿、凌动系列CPU
高通骁龙	骁龙660以后芯片的手机
华为NPU	mate10，mate10pro，P20，mate20，荣耀v20等
华为达芬奇NPU	mate30，p40，nova6，荣耀v30等
英伟达GPU	消费级显卡GeForce系列、RTX系列、TITAN，专业显卡Quadro、Tesla系列
英伟达Jetson	TX2、Nano、Xavier、Xavier NX

## 物体检测

### 整体介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL图像。

EasyDL图像支持定制图像分类、物体检测、图像分割三类模型。三类模型的功能区别如下：

- 图像分类：识别一张图中是否是某类物体/状态/场景，适用于图片内容单一、需要给整张图片分类的场景
- 物体检测：检测图中每个物体的位置、名称。适合图中有多个主体要识别、或要识别主体位置及数量的场景
- 图像分割：对比物体检测，支持用多边形标注训练数据，模型可像素级识别目标。适合图中有多个主体、需识别其位置或轮廓的场景

以下是关于物体检测模型的技术文档。

#### 应用场景

- 视频监控：如检测是否有违规物体、行为出现
- 工业质检：如检测图片里微小瑕疵的数量和位置
- 医疗诊断：如医疗细胞计数、中草药识别等

#### 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作。在数据已经准备好的情况下，最快几分钟即可获得定制模型。

[新手教程](#)详细介绍每一步的操作方式。如果文档没有解决您的问题，请在百度智能云控制台内[提交工单](#)反馈。



## 数据准备

### 创建数据集

在训练之前需要在数据中心【创建数据集】，导入并标注数据。

如果训练数据需要多人分工标注，可以创建多个数据集。将训练数据分批上传到这些数据集后，再将数据集"共享"给自己的小伙伴，同步进行标注。

### 设计标签

在上传之前确定想要识别哪几种物体，并上传含有这些物体的图片。每个标签对应想要在图片中检测出的一种物体

注意：标签的上限为1000种

### 准备图片

基于设计好的标签准备图片：

- 每种要识别的物体在所有图片中出现的数量需要大于50
- 如果某些要区分的物体具有相似性，需要增加更多图片
- 一个模型的图片总量限制4张~10万张
- 单张图片中的目标数不能超过1000个

如有特殊需求，请[提交工单](#)联系我们

### 图片格式要求：

- 目前支持图片类型为png、jpg、bmp、jpeg，图片大小限制在14M以内
- 图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px

### 图片内容要求：

- 训练图片和实际场景要识别的图片拍摄环境一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片；如果是需要识别白天光照下的物体，就不能使用夜晚拍摄的图片数据
- 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

如果需要寻求第三方数据采集团队协助数据采集，请在百度智能云控制台内[提交工单](#)反馈

### 上传数据集并在线标注

在完成了设计标签与准备数据后，可以通过以下方式导入数据：

- 导入未标注的数据，在线进行数据标注
- 直接导入标注好的数据

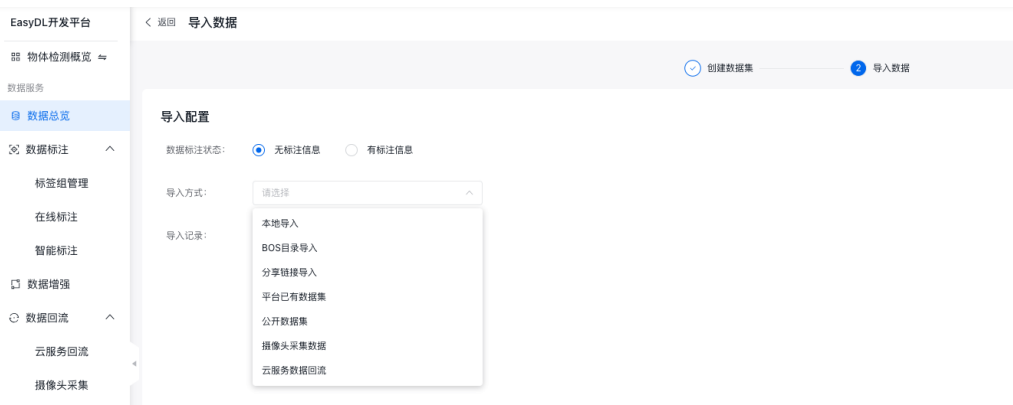
### 导入未标注数据

#### 本地数据

支持上传图片、压缩包，或通过[API导入](#)

#### 已有数据集

支持选择百度云BOS导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他图像类模型的数据集



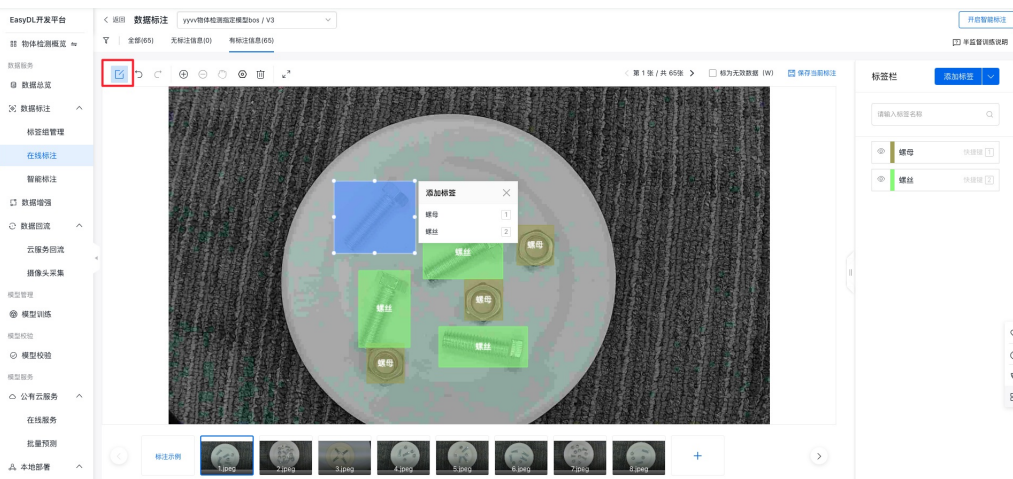
### 在线标注

上传未标注数据后，即可进入「标注数据集」页面进行在线标注

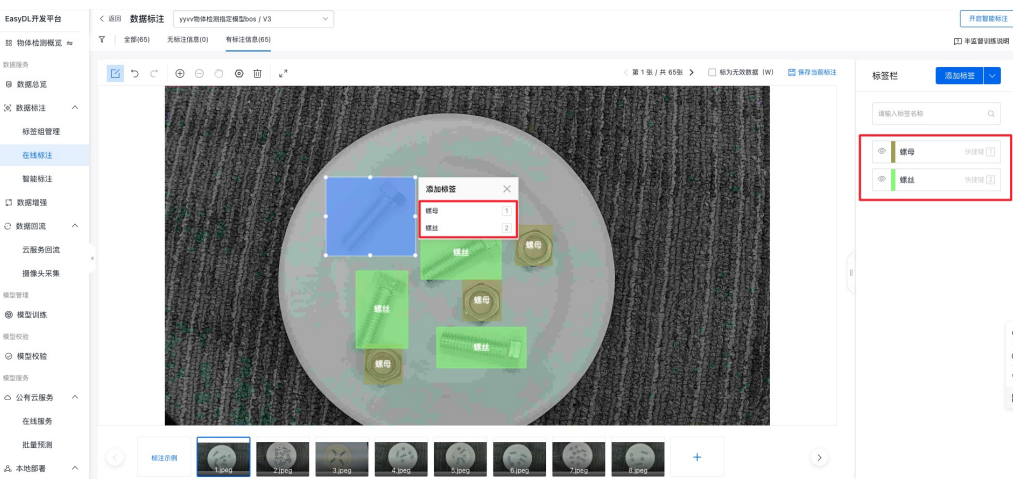
### 标注方法

注意：单张图片的标注框数不能超过500个，如有特殊需求，请[提交工单](#)联系我们

Step 1 首先在标注框上方找到工具栏，点击标注按钮在图片中拖动画框，圈出要识别的目标

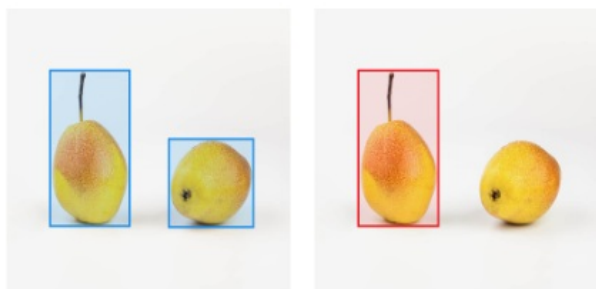


Step 2 然后在右侧的标签栏中，增加新标签，或选择已有标签



### 标注技巧

- 所有图片中出现的目标物体都需要被框出（框可以重叠）



全部框出

部分框出

- 框应包含整个物体，且尽可能不要包含多余的背景



包含整个物体

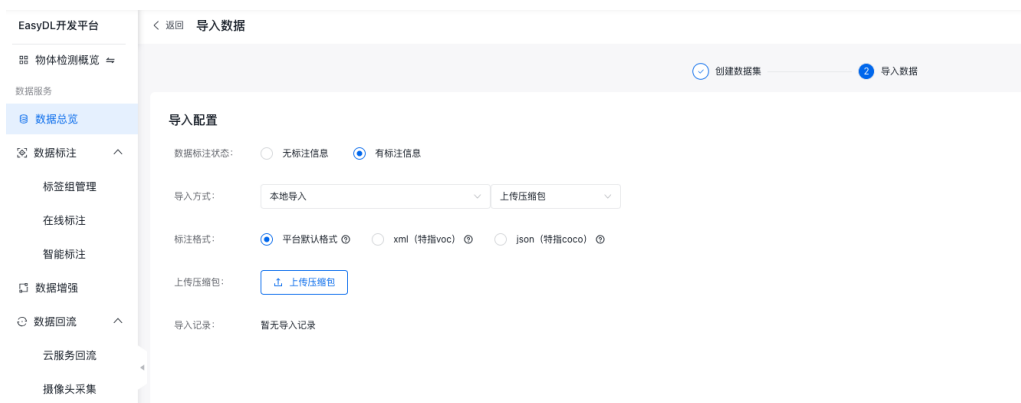
未框全或包含多余背景

- 如果图片中存在很多相同标签的目标物体，可以使用右侧的锁定按钮。锁定标签后，只需要在左侧框选目标物体即可，不用再重复选择标签
- 若需要标注的图片量较大时（如超过100张），可以启动智能标注来降低标注成本

## 导入已标注数据

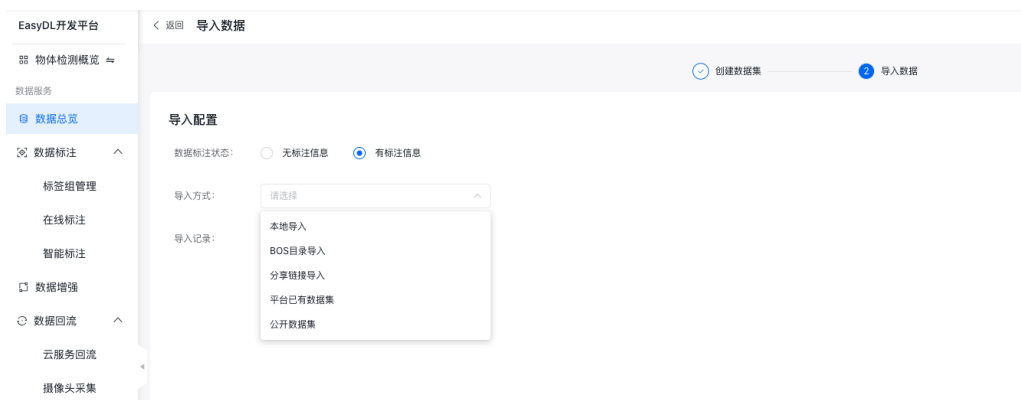
### 本地数据

支持上传压缩包，或通过API导入



### 已有数据集

支持选择百度云BOS导入、分享链接导入、平台已有数据集导入



## 数据集智能标注

使用智能标注功能可降低数据的标注成本。启动后，系统会从数据集所有图片中筛选出最关键的图片并提示需要优先标注。通常情况下，只需标注数据集30%左右的数据即可训练模型。与标注所有数据后训练相比，模型效果几乎等同

整体流程以物体检测的智能标注流程为例：

### 创建智能标注任务

启动物体检测数据集的智能标注前，请先检查以下是否已满足以下条件：

- 所有需要识别的标签都已创建
- 每个标签的标注框数不少于10个
- 所有需要标注的图片都已加入数据集，且所有不相关的图片都已删除

若已满足，即可从导航栏进入「数据服务」-「智能标注」，创建智能标注任务，系统会基于您选择数据类型及数据量级，自动预估任务运行时长



### 系统筛选难例

系统会分批筛选出最关键需标注的图片，即难例图片。

Tips：难例筛选需要一定时间，在此期间您可以正常进行其他未标注图片的标注



### 用户确认难例

智能标注任务启动后，系统为您自动筛选难例，您可以通过总览页查看进度按钮查看当前难例筛选进度，同时，进度图中也会全局展示您处于难例筛选的具体哪一环节，以便您的操作后续。筛选难例完成后，绿色进度条会进展到确认难例阶段，您可以点击【确认难例】完成对预标注结果的人工确认。

创建智能标注任务 ● 图像智能标注任务 ○ 文本智能标注任务

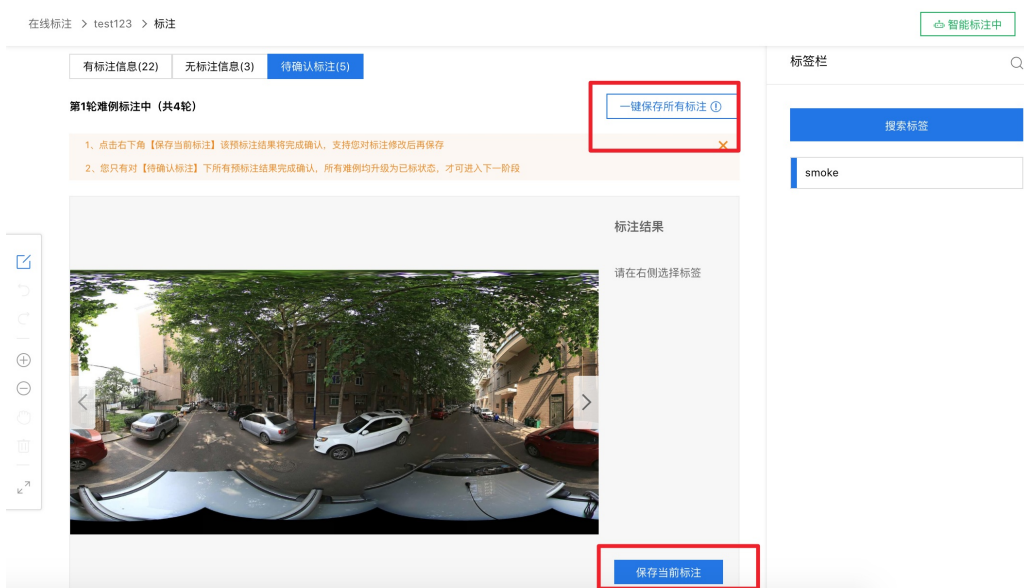
序号	数据集ID	数据集名称	版本	智能标注状态	操作
1	3107	tj-智能标注-检测-demo	V2	已中止	<a href="#">重新启动</a> <a href="#">查看记录</a>
2	2751	xyf_test_data1	V1	已中止	<a href="#">重新启动</a> <a href="#">查看记录</a>
3	256	test123	V1	运行中	<a href="#">查看进度</a> <a href="#">难例确认</a> <a href="#">中止任务</a> <a href="#">查看记录</a>
4	2744	zzy-测试智能标注			<a href="#">查看记录</a>
5	3111	tj-智能标注-检测-1110			<a href="#">查看记录</a>
6	2831	py3升级-智能标注-物体检测			<a href="#">查看记录</a>
7	1965	物体检测-多人标注用			<a href="#">查看记录</a>
8	2981	赵鸾专属数据集1			<a href="#">查看记录</a>

当前您处于第1轮难例阶段（共4轮）  
已为您筛选出本轮难例图片，请确认该轮难例图片

知道了

我们为您的人工确认提供两种模式：

- 单张确认，在该模式下支持您对预标注结果进行修正后点击保存
- 一键保存所有标注，为提升您的确认效率，默认您对难例的预标注结果全部满意，即可进入下一阶段



标注难例的预训练模型，也会对您无标注信息下的图片进行预标注结果的展示，您有余力的情况下，可以完成标注确认，确认后该张图片将升级为已标状态，该环节并非是您进入智能标注下一阶段的必备要求。



### 评估难例效果，完成任务

当您对难例完成确认后，您可以根据本轮次预标注的结果是否满意，判断您是否还需要进入下一轮难例筛选阶段，如果满意本轮难例的预标注效果，系统将自动为您系统其他的未标图片打标签。



### 第1轮难例标注中（共4轮）

- 1、点击右上角【保存当前标注】该预标注结果将完成确认，支持您对标注修改后再保存
- 2、您只有对【待确认标注】下所有预标注结果完成确认，所有难例均升级为已标状态，才可进入下一阶段



### 中止任务

当您在任务运行中想要中止任务时，可实时点击标注页面右上方【中止任务】按钮，任务将被提前结束。



### 其他操作提示

- 在智能标注任务中，有任务上限吗？

支持五条智能标注任务同时运行，超过该上限您需要中止其他任务

- 智能标注中可以增删标签吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签 如果确实需要增删标签，可以先结束智能标注

- 智能标注中可以增删图片吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片。如果确实需要增删图片，可以先结束智能标注

- 智能标注中可以修改已标注图片的标注框吗？

可以。但为了保证智能标注的效果，建议不要大量改动。如果确实需要修改大量标注，建议先结束智能标注

- 为什么我已经人工标注了很多图片，但系统预标注依然不准？

系统预标注的结果会受以下因素影响：智能标注期间，对“已标注”图片的标签进行大量改动；曾结束智能标注，并对标签、图片进行增删

- 多个数据集是否可以同时启动智能标注？

目前每个账号同一时间仅支持对一个数据集启动智能标注

- 共享中的数据集是否可以启动智能标注？

暂不支持。智能标注中的数据集也暂不支持共享

- 智能标注失败了怎么办？

可以先尝试稍后重新启动，如多次失败请[提交工单](#)联系我们

## 问题反馈

您在使用EasyData过程中可以通过以下任何方式联系我们：

- 在社区咨询

在论坛发帖提交问题，也可以在论坛与其他用户一起交流。[前往论坛](#)

- 提交工单

如果使用EasyData遇到其他任何问题或任何bug，您可以点此[提交工单](#)

- 添加微信小助手留言

请在微信搜索“BaiduEasyDL”，并备注暗号“EasyData”，添加小助手后留言。请在微信搜索“BaiduEasyDL”，并备注暗号“EasyData”，添加小助手后留言。

## 🔗 数据集多人标注

如果训练数据需要多人分工标注，可以创建多个数据集。将训练数据分批上传到这些数据集后，再将数据集“共享”给自己的小伙伴，同步进行标注。

共享方式如下：

### 1. 在「数据集管理」页面，点击需要共享的数据集对应操作栏中的「共享」

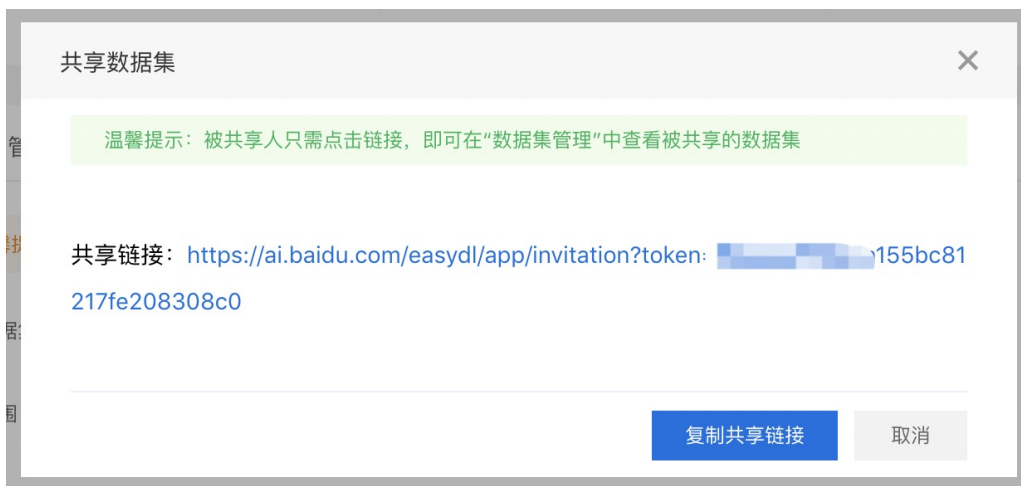


### 2. 在共享页面，勾选被共享数据集的授权使用范围，生成共享链接。如需被共享人标注数据，则需勾选「修改」



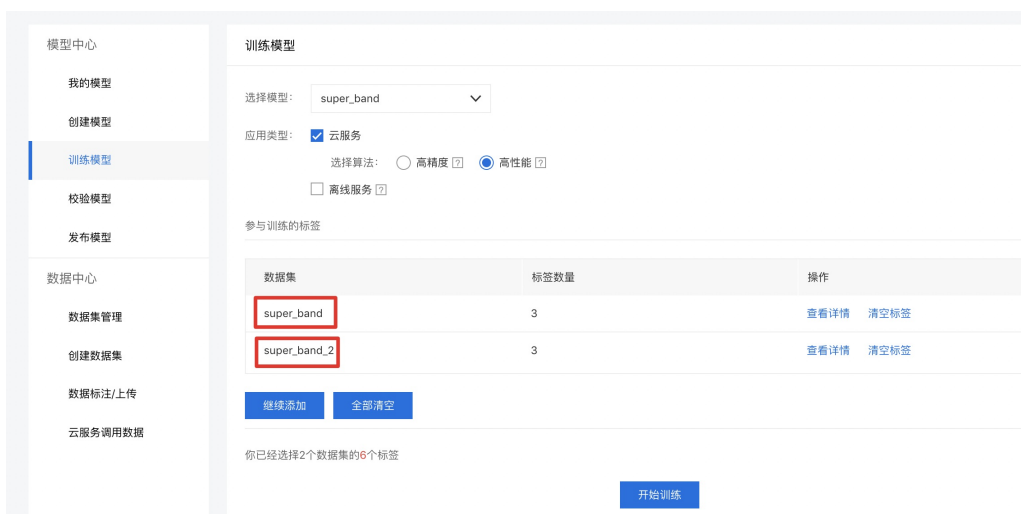
### 3. 复制共享链接，并发送给小伙伴





4. 被共享人打开链接后, 即可在「数据集管理」页面看到被共享的数据集, 并进行被授权的操作

5. 训练模型时, 在「训练模型」页面添加训练数据时, 可从多个数据集 (如多个被共享的数据集) 选择数据



## 🔗 数据集管理API

本文档主要说明当您线下已有大量的已经完成标注的图片数据, 如何通过调用API完成图片及标注的便捷上传和管理。EasyDL图像数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据, 只是在部分接口入参存在差异, 使用及接口地址完全一致。

### 数据集创建API

#### 接口描述

该接口可用于创建数据集。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法 : POST

请求URL : <https://aip.baidubce.com/rpc/2.0/easydl/dataset/create>

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

#### 数据集列表API

##### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

##### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

#### 分类（标签）列表API

##### 接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

##### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

#### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
append Label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类10000个汉字</b>
entity_name	是	string	文件名
labels	否	array(object)	标签/分类数据。若为空，则只上传图片，不上传标签/分类。若不为空，则应在数组中包含以下前面带+的参数
+label_name	是	string	标签/分类名称（由中文、数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

##### 接口描述

该接口可用于删除数据集。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 分类（标签）删除API

##### 接口描述

该接口可用于删除分类（标签）。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/label/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
2	Service temporarily unavailable	服务暂不可用, 请再次请求, 如果持续出现此类错误, 请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
3	Unsupported openapi method	调用的API不存在, 请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数, 请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法, 请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 🔗 数据质检

**功能概述** 该功能旨在对您数据集中的图像数据进行质量检测, 通过提供客观指标, 为您对数据集的下一步操作(标注、清洗等)进行参照引导。

整体质检报告将包括对原图、标注信息两个层面的指标进行统计, 本期先上线原图维度的质检指标, 标注层面的质检指标敬请期待。

### 使用流程 Step 1 功能入口

您可从数据总览页操作列点击【质检报告】或查看页面点击【质检报告】进入该功能页面

数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
5	● 已完成	图像分类	0% (0/5)	-	<a href="#">查看与标注</a> <a href="#">导出</a> <a href="#">删除</a> <a href="#">质检报告</a>

[< 返回](#)
[数据集详情](#)

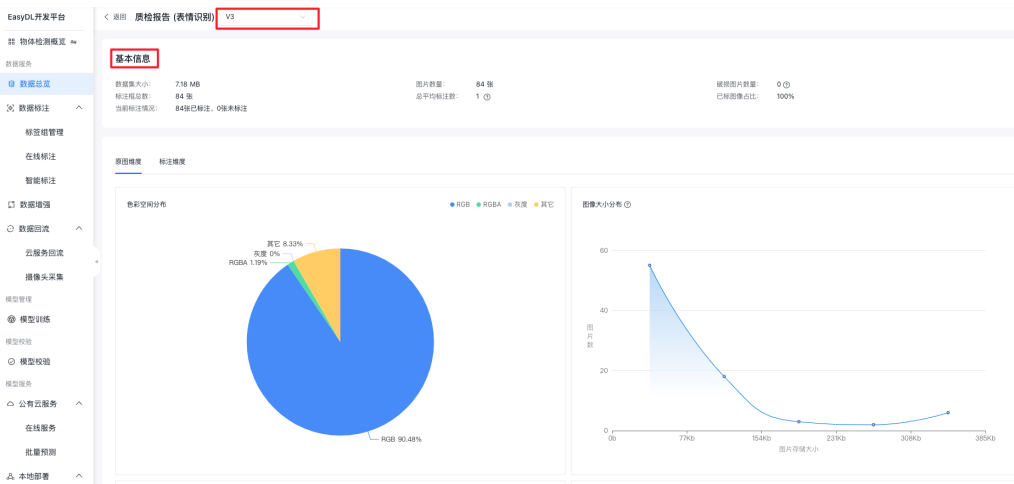
[导入](#)
[质检报告](#)
[数据标注](#)

Y 全部(4)
有标注信息(4)
无标注信息(0)

**Step 2 指标查看** 本期报告分为整体指标和分布指标两类。整体指标包括数据集存储大小、图片数量、破损图像数三类; 分布指标包括色彩分布空间、图像存储大小分布、高宽比分布、分辨率分布、色偏分布五类。

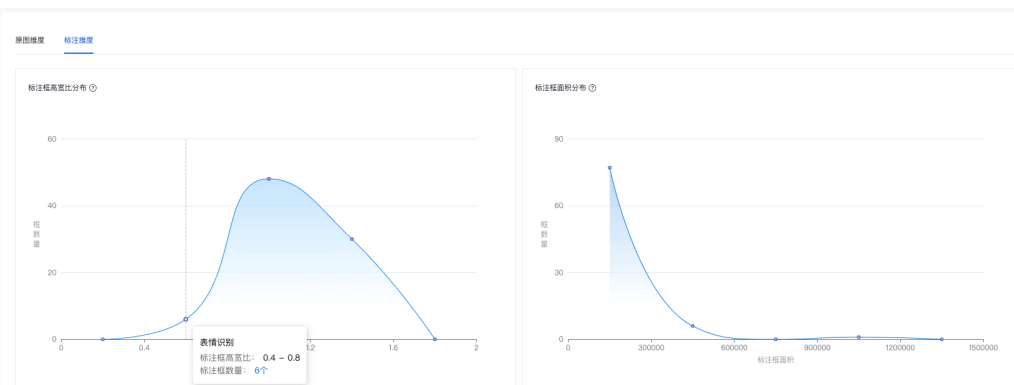
可以通过切换数据集版本查看不同版本下质检报告。





Step 3 对应处理 可通过hover具体指标数值进行相关操作，以高宽比分布为例：

第一步，高宽比在0.4-0.8的标注框，hover显示有6张图片，支持点击



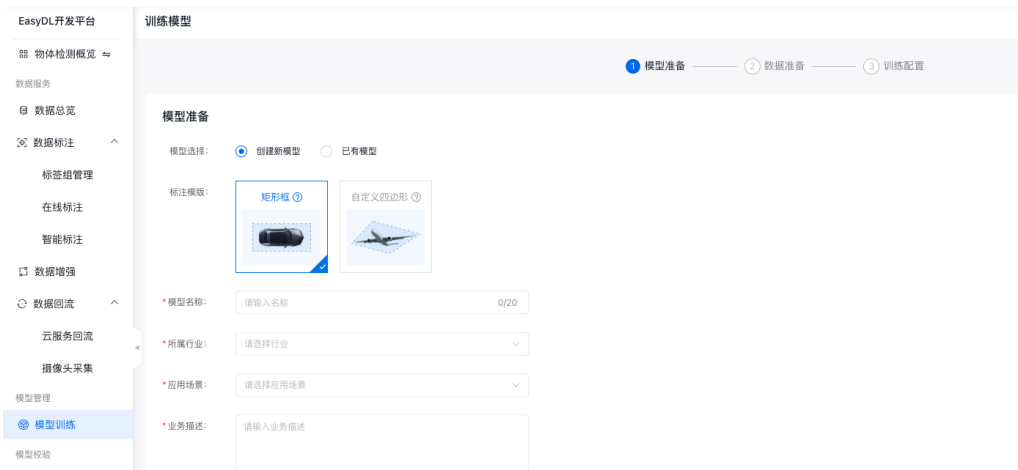
第二步，点击后进入符合该指标的图片操作页，可针对筛选后图片进行删除、标注等操作

### 模型训练

#### 物体检测创建模型

在导航【模型训练】中，点击训练模型，填写模型名称、所属行业、应用场景等信息，即可进入数据准备环节

操作示例：



- 注：1. 创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
- 2. 目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练。
- 3. 如果您是用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

🔗 物体检测训练操作说明

数据提交后，可以在导航中找到【训练模型】，按以下步骤操作，启动模型训练：

- 注：1. 启动训练前请确保数据已经标注完成，否则无法启动训练
- 2. 下述训练功能点中，标注为星号（\*）的功能为非必要选择项，可根据实际需求考虑是否使用



① 选择模型

选择此次训练的模型 ② 添加数据

**半监督训练\*** 半监督深度学习是半监督学习和深度学习结合的产物，可以理解为在深度学习算法中使用无标签样本。

模型取得优异表现离不开大量有标记样本。在现实生活中，有标记样本获取代价高昂，而无标签样本却很容易获得。由此想把半监督学习引入到深度学习中。

当打开半监督训练开关后，可以将未标注的数据添加至训练数据中参加训练。同时，这些未标注的数据在半监督训练完成后将会自动生成对应的标签信息，如在「保存自动生成标签」字段下选择了“是”，则可在EasyData数据服务对应数据集中查看并确认对应的标签结果

注：开启半监督训练后会增加部分训练时间，一般不会大于对应全量标注数据训练的训练时间两倍，请根据实际需求考虑后选择。例如「80

已标注样本+20未标注样本」半监督训练与「100已标注样本」常规训练的训练时间对比，前者训练时间会更长，但不会大于后者训练时长的两倍

\*添加数据集 + 请选择

数据集	版本	分类数量	数据量	标注状态	操作
testss	V1	1	1	已标注	<a href="#">移除</a>
testss	V1	1	38	未标注	<a href="#">移除</a>

保存自动生成标签  是  否

### 添加训练数据

- 先选择数据集，再按标签选择数据集里的图片，可从多个数据集选择图片
- 训练时间与数据量大小有关，1000张图片可能需要几个小时训练，请耐心等待

### Tips :

- 如果包含同一个标签的数据分散在不同的数据集里，可以在训练时同时从这些数据集里选择，模型训练时会按标签名称合并

**添加自定义验证集\*** AI模型在训练时，每训练一批数据会进行模型效果检验，以某一张验证图片作为验证数据，通过验证结果反馈去调节训练。可以简单地把AI模型训练理解为学生学习，训练集则为每天的上课内容，验证集即为每周的课后作业，质量更高的每周课后作业能够更好的指导学生学习和找寻自己的不足，从而提高成绩。同理AI模型训练的验证集也是这个功效。

注：学生的课后作业应该与上课内容对应，这样才能巩固知识。因此，验证集的标签也应与训练集完全一致。

**添加自定义测试集\*** 如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果。

注：期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可。

### 配置数据增强策略

深度学习模型的成功很大程度上要归功于大量的标注数据集。通常来说，通过增加数据的数量和多样性往往能提升模型的效果。当在实践中无法收集到数目庞大的高质量数据时，可以通过配置数据增强策略，对数据本身进行一定程度的扰动从而产生"新"数据。模型会通过学习大量的"新"数据，提高泛化能力。

你可以在「默认配置」、「手动配置」、「自动数据增强」3种方式中进行选择，完成数据增强策略的配置。

#### 默认配置

如果你不需要特别配置数据增强策略，就可以选择默认配置。后台会根据你选择的算法，自动配置必要的的数据增强策略。

#### 手动配置

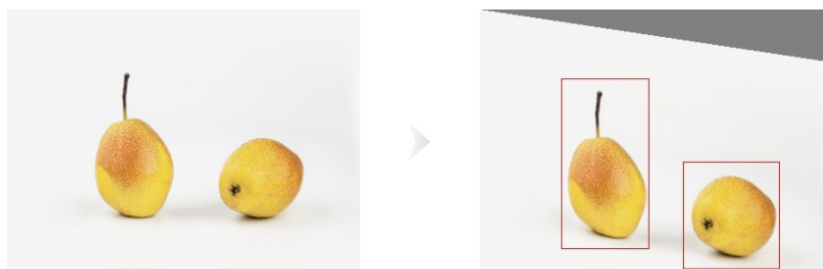
EasyDL提供了大量的数据增强算子供开发者手动配置。你可以通过下方的算子功能说明或训练页面的效果展示，来了解不同算子的功能：

算子名	功能
ShearX_BBox	剪切图像的水平边
ShearX_Only_BBoxes	剪切标注框内图像的水平边
ShearY_BBox	剪切图像的垂直边
ShearY_Only_BBoxes	剪切标注框内图像的垂直边
TranslateX_BBox	按指定距离（像素点个数）水平移动图像及标注框
TranslateX_Only_BBoxes	按指定距离（像素点个数）水平移动标注框内的图像
TranslateY_BBox	按指定距离（像素点个数）垂直移动图像及标注框
TranslateY_Only_BBoxes	按指定距离（像素点个数）垂直移动标注框内的图像
Rotate_BBox	按指定角度旋转图像及标注框
Rotate_Only_BBoxes	按指定角度旋转标注框内的图像
AutoContrast	自动优化图像对比度
Contrast	调整图像对比度
Equalize	将图像转换为灰色值均匀分布的图像
Equalize_Only_BBoxes	将标注框内的图像转换为灰色值均匀分布的图像
Solarize	为图像中指定阈值之上的所有像素值取反
Solarize_Only_BBoxes	为标注框内的图像中指定阈值之上的所有像素值取反
Solarize_add	为图像中指定阈值之下的所有像素值加上像素偏移值
Posterize	减少每个颜色通道的bits至指定位数
Color	调整图像颜色平衡
Brightness	调整图像亮度
Sharpness	调整图像清晰度
Cutout	通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例
BBox_Cutout	通过在标注框附近进行随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例
Cutout_Only_BBoxes	只在标注框内通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例
Flip_Only_BBoxes	对标注框内的图像进行左右翻转

## 效果展示



剪切图像的垂直边，能更好地识别发生了垂直方向变形的图像



关闭

## 自动数据增强

在训练方式选择「精度提升配置包」选项后，此处数据增强策略提供「自动数据增强」选项。自动数据增强算法会根据您数据的特性，自动选择数据增强算子。使用付费机型训练的用户请注意，自动数据增强算法可能会增加模型训练时间。

模型训练完成后，可在「我的模型-查看版本配置」中，查看配置记录：

【物体检测】 zh_detect_default 模型ID: 10065						
部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V4	训练完成	未申请	未发布	mAP 80.06% 精确率 75.00% 召回率 85.71% 完整评估结果	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>

## 配置建议

算子的配置建议贴合实际场景。

比如，数字识别的数据集中，因为对数字的旋转很有可能导致错误样本的产生，所以不建议对数字数据集进行旋转操作。再比如，检测数据集中，如果标注量比较少，就可以通过随机平移的算子增强数据集，模型也更容易学习到目标物体的平移不变性。

## ③ 训练配置

### 部署方式

可选择「公有云API」、「EasyEdge本地部署」

#### 如何选择部署方式

**选择设备** 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择-如果您选择了「公有云API」，则可按需选择训练方式 **增量训练\*** 增量训练：在模型迭代训练时，用户在原训练数据上增加了训练数据，可通过加载原训练数据训练的模型参数进行模型训练。这样可以让模型收敛速度变快，训练时间变短，同时在数据集质量较高的情况下，可能获得的模型效果也会更好。

注：仅可选择同一部署方式下的训练的模型作为基准模型版本

### 训练方式

- 「常规训练」包括EasyDL历史提供的「高精度」、「高性能」等模型选择，以及常规的模型训练配置
- 「精度提升配置包」选用百度自有超大规模预训练模型，让模型有更好的精度效果。并提供按云调用时延选择网络模型的形式，根据您的实际应用场景需求，选择更合适的模型。

**自动超参搜索\*** 自动超参搜索目前仅在精度提升配置包的选项下提供。选择开启自动超参搜索后，算法会多次实验，自动搜寻出适合模型训练的各种参数，来达到高精度的模型效果。

注：开启自动超参搜索后会增加3倍以上的训练时间，请根据实际需求考虑后选择

**高级训练配置\*** 高级训练配置开关默认关闭，建议对深度学习有一定了解的用户根据实际情况考虑使用。高级训练配置目前提供「输入图片分辨率」、「epoch」、两个配置项

- 输入图片分辨率：可以根据具体应用场景选择输入图片分辨率，如检测目标在图片中较小，就可适当增加输入图片分辨率，增强检测目标在数据层面的特性。推荐值为该类算法任务输入图片分辨率普遍最优值。
- epoch：训练集完整参与训练的次数。如有训练数据集较大，模型训练不充分，模型精度较低的情况，可适当设置较大epoch值（大于100），使模型训练更完整。

### 选择算法

不同的部署方式下，可以选择不同的算法。每个算法旁边有一个小问号，可以查看详细说明。

例如：选择「公有云API」后，可以在「超高精度」、「高精度」、「高性能」3种算法中选择。鼠标移动到「高精度」右侧的问号上，可以看到对高精度算法的详细说明。



通常，高精度模型在识别准确率上表现较好，但在识别速度上表现较弱。高性能模型反之。

在「精度提升配置包」中提供「小目标检测」算法供用户选择，当检测目标小于图片的5%，使用小目标检测算法可获得效果不错的模型。

注：小目标检测算法目前仅支持本地服务器部署



此外，如果你已从AI市场购买了模型算法，也可以基于已购模型的算法训练：[前往AI市场购买](#)>

#### ④ 训练模型

点击「开始训练」，训练模型。

- 训练时间与数据量大小有关，1000张图片可能需要几个小时训练，请耐心等待。
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面。
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

#### 🔗 物体检测模型效果评估

可通过模型评估报告或模型校验了解模型效果：

- 模型评估报告：训练完成后，可以在【我的模型】列表中看到模型效果，以及详细的模型评估报告。
- 模型在线校验：可以在左侧导航中找到【校验模型】，在线校验模型效果。校验功能示意图：

### 校验模型

选择模型: super\_band
应用类型: 云服务 (目前仅支持)
选择版本: V7

当前模型mAP平均精度 89.13% [评估报告](#)



[点击添加图片](#)

识别结果 [如何优化效果?](#)

调整阈值 


 当前阈值: 0.3

预测标签	置信度>30%
1. person	98.41%
2. positive	94.63%

申请上线
纠正识别结果

## 模型评估报告

### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、mAP、精确率、召回率。这部分模型效果的指标是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如图片数量低于100个），参与评估的数据可能不超过30个，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

**注意：**若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口批量测试，获取更准确的模型效果。

#### 整体评估

cat\_dog V2效果优异，建议针对识别错误的图片示例继续优化模型效果。由于目前训练集数据量较少，该结论仅供参考，建议扩充训练集得到更准确的评估效果。 [如何优化效果?](#)



查看模型评估结果时，需要思考在当前业务场景，更关注精确率与召回率哪个指标。是更希望减少误识别，还是更希望减少漏识别。前者更需要关注精确率的指标，后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况，对于希望准确率与召回率兼具的场景，F1-score越接近1效果越好。评估指标说明如下

**F1-score：**对某类别而言为精确率和召回率的调和平均数，评估报告中指各类别F1-score的平均数

**mAP：**mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务，每一类object都可以计算出其精确率(Precision)和召回率(Recall)，在不同阈值下多次计算/试验，每个类都可以得到一条P-R曲线，曲线下的面积就是average

**精确率：**正确预测的物体数与预测物体总数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

**召回率：**正确预测的物体数与真实物体数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

**模型调优建议** 在模型评估中，EasyDL将会通过智能算法对误识别的样本进行归因分析，可推断出误识别的样本对某个模型评估指标的具体影响以及影响程度，并提供对应优化的方案。同时还可针对某个具体表现不好的标签进行归因分析，针对性优化识别效果

**模型调优建议**

归因粒度 **基于整个模型** 基于单个标签

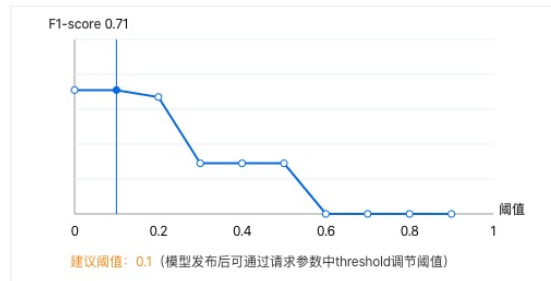
序号	受影响指标	影响程度	根因分析	调优对策
1	F1-Score	中	“高宽比”对“F1-Score”的效果有“一定”影响,不同特征区间的“F1-Score”方差达到“0.0352”	在【添加数据】->【数据增强策略】中配置“ShearX,ShearY”进行增强。
2	F1-Score	中	“分辨率”对“F1-Score”的效果有“一定”影响,不同特征区间的“F1-Score”方差达到“0.0321”	在【添加数据】->【数据增强策略】中配置“ShearX,ShearY”进行增强。
3	F1-Score	中	“色偏”对“F1-Score”的效果有“一定”影响,不同特征区间的“F1-Score”方差达到“0.0205”	在【添加数据】->【数据增强策略】中配置“Color,Posterize”进行增强。
4	F1-Score	中	“亮度”对“F1-Score”的效果有“一定”影响,不同特征区间的“F1-Score”方差达到“0.0188”	在【添加数据】->【数据增强策略】中配置“Brightness”进行增强。
5	F1-Score	中	“饱和度”对“F1-Score”的效果有“一定”影响,不同特征区间的“F1-Score”方差达到“0.018”	在【添加数据】->【数据增强策略】中配置“Color”进行增强。

### 详细评估

在这个部分可以看到不同阈值下的F1-score、模型识别错误的图片示例，以及使用混淆矩阵定位易混淆的标签。

#### 详细评估

不同阈值下F1-score表现



不同标签的mAP及对应的识别错误的图片

<p>cat <span style="display: inline-block; width: 85%; height: 10px; background-color: #007bff;"></span> 85%</p> <p>dog <span style="display: inline-block; width: 90%; height: 10px; background-color: #007bff;"></span> 90%</p>	<p>cat的错误结果示例 (点击查看识别错误详情)</p>
---	--------------------------------

### 识别错误图片示例

通过分标签查看模型识别错误的图片，直寻找其中的共性，进而有针对性的扩充训练数据；或发现是标注错误，从而直接点击修改标注来将标注修正

如下图所示，可以通过勾选「误识别」、「漏识别」来分别查看两种错误识别的情况：





### • 误识别：红框内没有目标物体（准备训练数据时没有标注），但模型识别到了目标物体

观察误识别的目标有什么共性：例如，一个检测电动车的模型，把很多自行车误识别成了电动车（因为电动车和自行车外观上比较相似）。这时，就需要在训练集中为自行车特别建立一个标签，并且在所有训练集图片中，将自行车标注出来。

可以把模型想象成一个在认识世界的孩童，当你告诉他电动车和自行车分别是什么样时，他就能认出来；当你没有告诉他的时候，他就有可能把自行车认成电动车。

### • 漏识别：橙框内应该有目标物体（准备训练数据时标注了），但模型没能识别出目标物体

观察漏识别的目标有什么共性：例如，一个检测会议室参会人数的模型，会漏识别图片中出现的白色人种。这大概率是因为训练集中缺少白色人种的标注数据造成的。因此，需要在训练集中添加包含白色人种的图片，并将白色人种标注出来。

黄色人种和白色人种在外貌的差别上是比较明显的，由于几乎所有的训练数据都标注的是黄色人种，所以模型很可能认不出白色人种。需要增加白色人种的标注数据，让模型学习到黄色人种和白色人种都属于「参会人员」这个标签。

以上例子中，我们找到的是识别错误的图片中，目标特征上的共性。除此之外，还可以观察识别错误的图片在以下维度是否有共性，比如：图片的拍摄设备、拍摄角度，图片的亮度、背景等等。

### 定位易混淆标签

支持按识别错误样本量的绝对数值/相对数值查看混淆矩阵，同时支持下载完整的混淆矩阵进行更深入的分析。

#### 定位易混淆标签

下方是Auto2022031600:32:46 V1模型的混淆矩阵，每一个橙色的方格都对应一组易混淆的标签（最多展示10个易混淆的标签）

展示了数据标注与模型预测不符数量前10的标签，点击标签可在下方查看示例图，帮助您有针对性地设计特征，使得类别更具区分性。

[↓ 下载完整混淆矩阵](#)

序号	标签名称	误识别标签TOP5及其数量	精确率	测试集数量	召回率	f1-score
1	Cacca	[8] 背景类	100.0%	10	20%	33%
2	Lopl		25.0%	2	100%	40%
3	Caccg	[20] 背景类 [1] Cksg	93.2%	72	76%	84%
4	Ssdwl	[14] Ssdvr	87.7%	83	86%	87%
5	Ckksa	[2] 背景类	100.0%	10	80%	89%

### 🔗 物体检测模型如何提升效果

一个模型很难一次性就训练到最佳的效果，可能需要结合模型评估报告和校验结果不断扩充数据和调优。

为此我们设计了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，获得更好的模型效果。

**注意：如果模型已经是上线状态（包括已付费的模型服务），依然支持模型迭代。只需要在训练完毕后发布新的版本，就可以获得更新后的模型服务。**

想要提升模型效果，可以尝试以下两种方法：

#### 检查并优化训练数据

1. 检查是否存在训练数据过少的情况，建议每个标签标注50个框以上，如果低于这个量级建议扩充。

2. 检查不同标签的标注框数量是否均衡，建议不同标签的标注框数数据量级相同，并尽量接近，如果有的标签框数很多，有的标签框数很少，会影响模型整体的识别效果。
3. 通过模型效果评估报告中的错误识别示例，有针对性地扩充训练数据。
4. 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

## 云服务调用数据管理

开通云服务调用数据管理功能后，可查找云服务模型识别错误的图片，纠正结果并将其加入模型迭代的训练集，实现训练数据的持续丰富和模型效果的持续优化

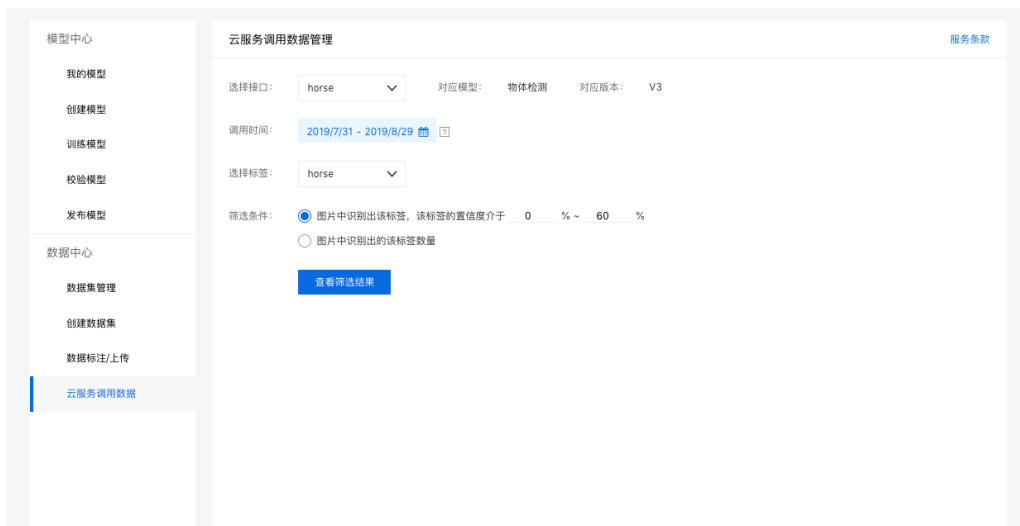
具体使用流程如下：

1. 为已上线接口开通云服务调用数据服务



2. 通过选择调用时间、标签，并设置筛选条件，查看疑似错误识别的图片

注意：数据将从开通功能后开始存储，最多存储30天的数据。当天调用的数据暂不支持即时查看，可在第二天查看



3. 将接口识别错误的图片添加到指定数据集（建议新建数据集）并纠正结果。后续训练模型时，只需增加包含接口数据的数据集，即可提升模型效果

**尝试不同的训练配置** 可前往训练配置页面尝试不同的配置组合，因不同数据集在不同的算法上可能表现不一致，所以建议您多尝试不同的算法选型后综合挑选精度最高的模型使用，你可以选择如下的配置项：

- 增量训练
- 精度提升配置包
- 自动超参搜索
- 自定义验证集

- 数据增强策略
- 在高级训练配置中增加输入图片分辨率

注：如您需检测的目标在图中占比小于5%，建议您选择「精度提升配置包」中的小目标检测算法，但出于对算法性能的考虑，目前小目标检测算法仅支持本地部署



## 模型发布

### 🔗 物体检测模型发布整体说明

训练完成后，可将模型部署在公有云服务器、通用小型设备、本地服务器，或直接购买软硬一体方案，灵活适配各种使用场景及运行环境

#### 公有云在线服务

训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合

具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

支持查找云端模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，不断优化模型效果

**纯离线服务** 训练完成的模型整体打包为纯离线服务，可下载在本地稳定调用。纯离线服务按部署硬件芯片不同分为本地服务器部署、通用小型设备部署。为了提供更好的算法与硬件推理效果，EasyDL提供软硬一体方案部署。纯离线服务的整体支持与评测信息可详见[算法与性能评测大表](#)

#### 本地服务器部署

可将训练完成的模型部署在私有CPU/GPU服务器上，支持服务器API和服务器SDK两种集成方式

模型服务性能表现更好，适用于对性能要求较高的场景，例如工业质检、流水线产品分拣等

#### 通用小型设备

训练完成的模型被打包成适配智能硬件的SDK，可进行设备端离线计算。满足推理阶段数据敏感性要求、更快的响应速度要求

支持iOS、Android、Linux、Windows四种操作系统，基础接口封装完善，满足灵活的应用侧二次开发

#### 软硬一体方案

高性能硬件与模型深度适配，多种方案可选。可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松。[了解更多](#)

## 端云协同服务

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新

断网状态下模型离线计算 (http服务, 可调用与公有云API功能相同的接口)

联网状态下在平台管理设备运行状态、资源利用率

公有云部署

如何发布物体检测API

在新手教程中点击链接过来的用户请注意, 您仍需要完整训练模型后, 按如下操作指引, 方可使用公有云服务

训练完毕后可以在左侧导航栏中找到【发布模型】, 依次进行以下操作即可发布公有云API:

- 选择模型
- 选择部署方式「公有云部署」
- 选择版本
- 自定义服务名称、接口地址后缀
- 申请发布

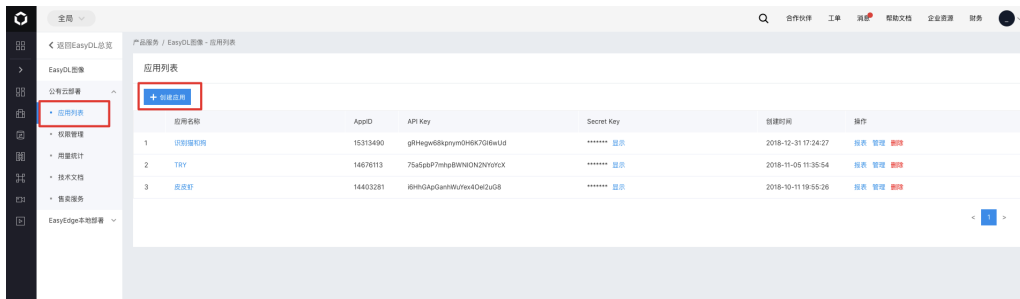
申请发布后, 通常的审核周期为T+1, 即当天申请第二天可以审核完成。如果需要加急、或者遇到莫名被拒的情况, 请在百度智能云控制台内[提交工单反馈](#)

发布模型界面示意:



### 接口赋权

在正式使用之前, 还需要做的一项工作为接口赋权, 需要登录[EasyDL版控制台](#)中创建一个应用, 获得由一串数字组成的appid, 然后就可以参考接口文档正式使用了



同时支持在「公有云服务管理」-「权限管理」中为第三方用户配置权限

示意图如下:



### 🔗 物体检测API调用文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

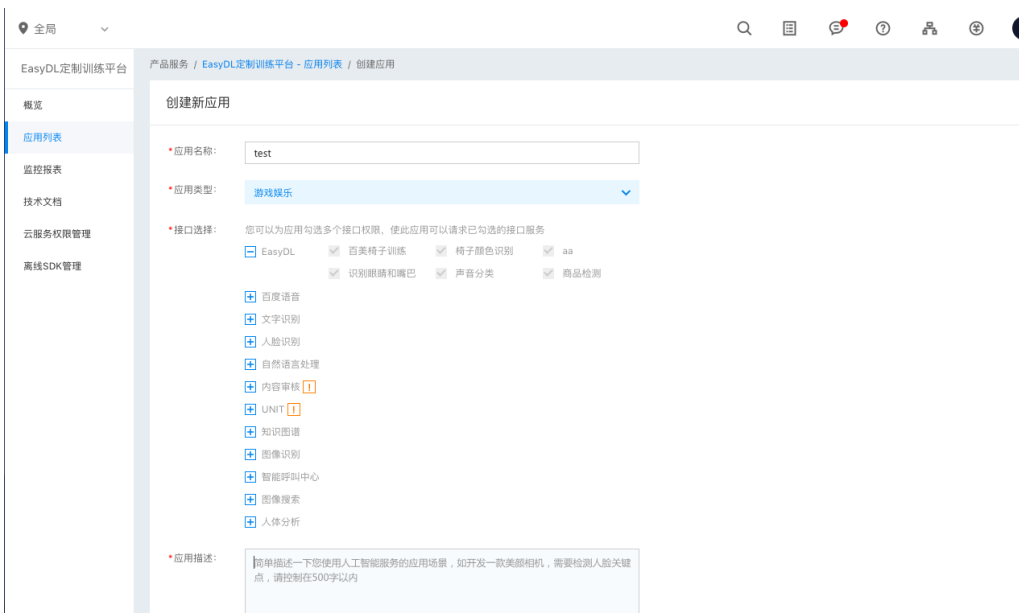
- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#) ,与其他开发者进行互动

### 接口描述

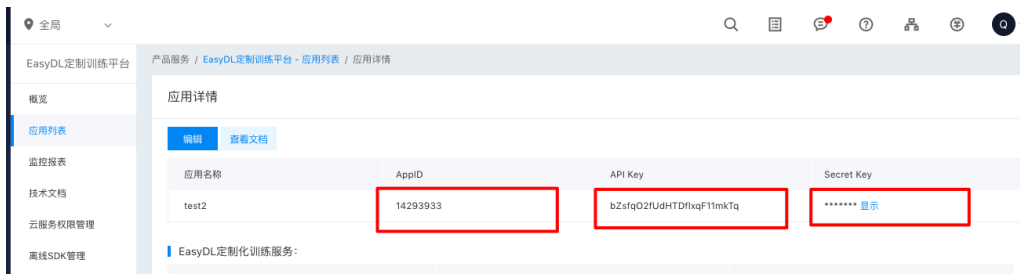
基于自定义训练出的物体检测模型，实现定制图像识别。

### 接口鉴权

#### 1、在EasyDL控制台创建应用



#### 2、应用详情页获取AK SK



**请求说明**

**请求示例**

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后申请上线，上线成功后可在服务列表中查看并获取url。

URL参数：

参数	值
input_type	当取值为 url 时，需在请求参数中传入图片的URL string
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

**请求参数**

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
threshold	否	number	-	默认值为推荐阈值，请在我的模型列表-模型效果查看推荐阈值
url	否	string	-	如果请求URL参数中增加“input_type=url”，则该参数必传，否则“image”参数必传。参数内容为URL string，用户需确保该string是有效的图片URL，否则会下载失败

**请求代码示例**

提示一：使用示例代码前，请记得替换其中的示例Token、图片地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

PHP
Java
Python3
C++

```

<?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = '', $param = '')
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
}

```

### 返回说明

### 物体检测-矩形框标注

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	识别结果数组
+name	否	string	分类名称
+score	否	number	置信度
+location	否		
++left	否	number	检测到的目标主体区域到图片左边界的距离
++top	否	number	检测到的目标主体区域到图片上边界的距离
++width	否	number	检测到的目标主体区域的宽度
++height	否	number	检测到的目标主体区域的高度

### 物体检测-自定义四边形标注

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	物体检测目标信息
+name	否	string	目标物体标签
+score	否	number	置信度
+location	否	object	目标物体所在位置
++points	否	list(object)	目标物体所在四边形的顶点信息
+++x	否	number	顶点横坐标
+++y	否	number	顶点纵坐标
error_code	否	number	错误码, 当请求错误时返回
error_msg	否	string	错误描述信息, 当请求错误时返回

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误示例



需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 本地服务器部署

### 如何在本地服务器部署

训练完毕后，可以选择将模型通过「纯离线服务」或「端云协同服务」部署，具体介绍如下：

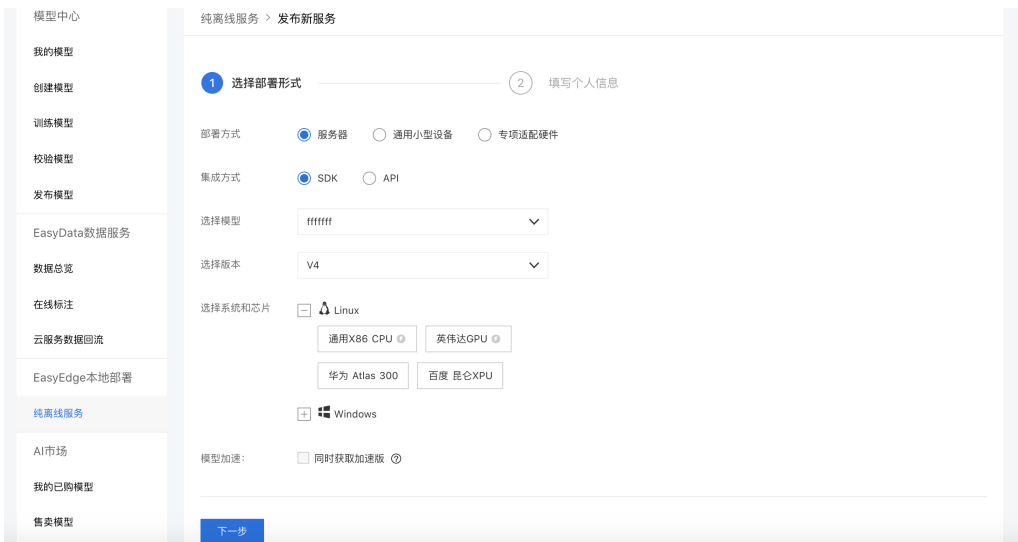
#### 纯离线服务部署

可以在左侧导航栏中找到「纯离线服务」，依次进行以下操作即可将模型部署到本地服务器：

- 选择部署方式「服务器」
- 选择集成方式
- 选择模型、版本、系统和芯片

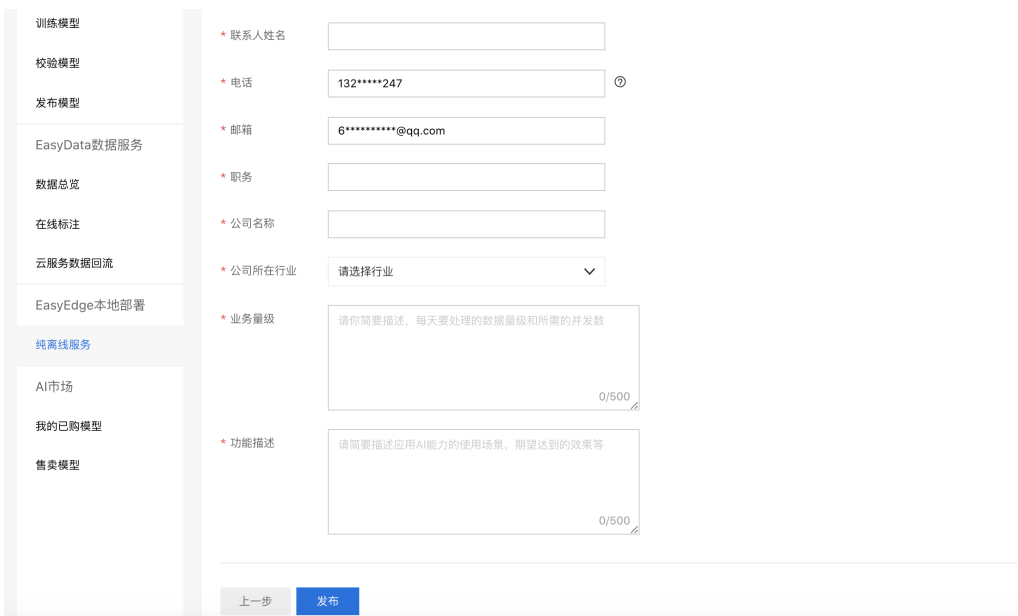


• 点击下一步



• 填写部分信息（注：个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用）

• 点击发布



① 私有API

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

点击「发布」后，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

② 服务器端SDK

将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活

1、点击「发布」后，前往[控制台](#)申请服务器端SDK的试用序列号

2、点击「新增测试序列号」，根据模型类型选择「序列号类型」，填写「新增设备数」（所得序列号数量），点击确定即可



### 3、离线SDK的激活和使用，请参考文档完成集成



### 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供，基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

### 本地服务器部署价格说明

EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

如需购买永久使用授权，服务器SDK用户请在[控制台](#)点击「购买正式授权」，并按照对应步骤激活。

服务器API用户请微信搜索“BaiduEasyDL”添加小助手咨询，通过线下签订合同购买使用。

### 更多参考

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[纯离线SDK说明](#)

[纯离线SDK简介](#)

本文档主要说明定制化模型发布后获得的服务器端SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### SDK说明

物体检测服务器端SDK支持Linux、Windows两种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux		CPU: x86_64 NVIDIA GPU: x86_64 HUAWEI Atlas 300: x86_64
Windows	64位 Windows7 及以上	NVIDIA GPU: x86_64  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015  GPU依赖： CUDA 9.x + cuDNN 7.x

#### 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)查看评测信息表

#### 激活&使用步骤

离线SDK的激活与使用分以下三步：

- ① 下载SDK后，在[控制台](#)获取序列号
- ② 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

- ③ 正式使用

#### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在[百度云控制台](#)内[提交工单](#)反馈

#### 1、激活失败怎么办？

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在[百度云控制台](#)内[提交工单](#)反馈

#### Windows集成文档

##### 简介

本文档介绍物体检测服务器端Windows SDK的使用方法。

- 硬件支持：
  - NVIDIA GPU（普通版，加速版）
- 操作系统支持
  - 64位 Windows 7 及以上
  - 64位 Windows Server 2012及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- GPU基础版（EasyEdge-win-x86-nvidia-gpu）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib：<http://www.winimage.com/zLibDll/zlib123dllx64.zip>，解压后将dll\_x64/zlibwapi.dll 拷贝到cuda的bin目录下) + 硬件计算能力(<https://developer.nvidia.com/cuda-gpus#compute>)达6.1及以上
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + 硬件计算能力达7.5及以上
- GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.x.x
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.x.x
- GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.3.1 + 硬件计算能力达6.1及以上
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.1.6 + 硬件计算能力达7.5及以上
- GPU加速版（x86-nvidia-gpu-torch）
  - CUDA 11.0.x + cuDNN 8.0.5.x
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级，下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 优化模型算法；GPU CUDA9.0 CUDA10.0 标记为待废弃状态 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | GPU基础版推理引擎优化升级；GPU加速版支持自定义模型文件缓存路径；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | 修复已知问题 | | 2021-08-19 | 1.3.2 | 新增支持EasyDL小目标检测，新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | 修复已知问题 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020-12-18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020-10-29 | 1.1.19 | 修复已知问题 | | 2020-09-17 | 1.1.18 | 支持更多模型 | | 2020.08.11 | 1.1.17 | 支持专业版更多模型 | | 2020.06.23 | 1.1.16 | 支持专业版更多模型 | | 2020.05.15 | 1.1.15 | 更新加速版tensorrt版本，支持高精度检测 | | 2020.03.13 | 1.1.14 | 支持声音分类 | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | 支持物体检测高精度算法的CPU加速版，EasyDL 专业版支持 SDK 加速版 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版！ |

## 快速开始

### 1. 安装依赖

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

### 如果使用GPU版SDK，请安装CUDA + cuDNN

<https://developer.nvidia.com/cuda>  
<https://developer.nvidia.com/cudnn>

### 如果使用GPU版加速版SDK，请安装TensorRT

<https://developer.nvidia.com/tensorrt>

根据cuda版本下载，下载后把lib目录下的所有dll，拷贝到SDK的dll目录下

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，SDK默认使用cuda9版本，如果需要cuda10请运行EasyEdge CUDA10.0.bat切换到cuda10版本，之后打开EasyEdge.exe，选择鉴权模式，输入Serial Num，点击“启动服务”，等待数秒即可启动成功，本地服务默认运行在

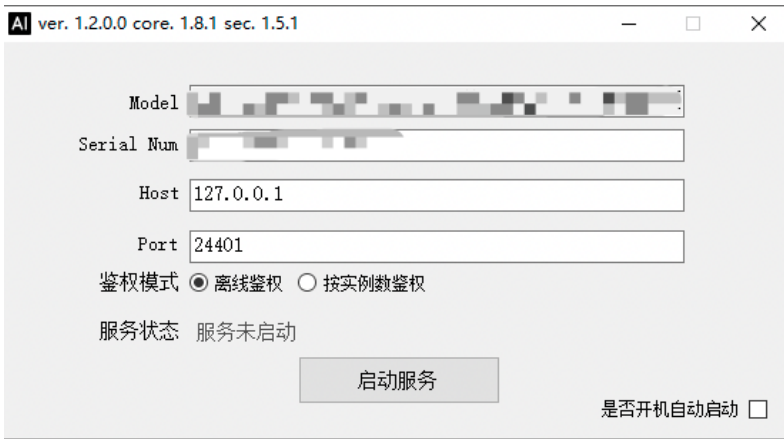
<http://127.0.0.1:24401/>

其他任何语言只需通过HTTP调用即可。

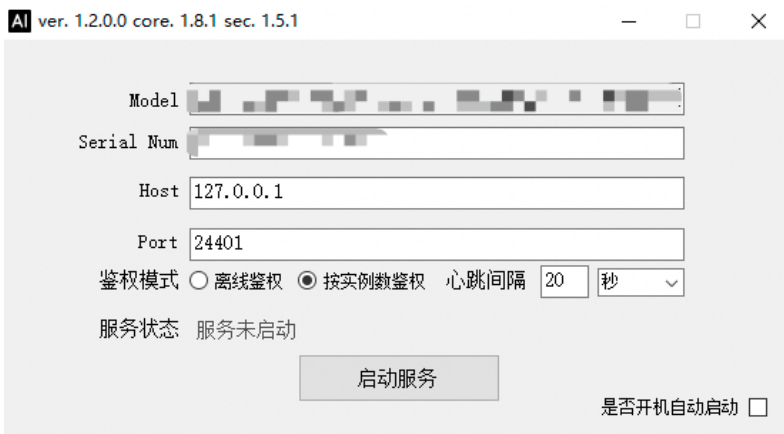
如启动失败，可参考如下步骤排查：



#### 2.1 离线鉴权（默认鉴权模式）首次联网激活，后续离线使用



2.2 按实例数鉴权 周期性联网激活，离线后会释放所占鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

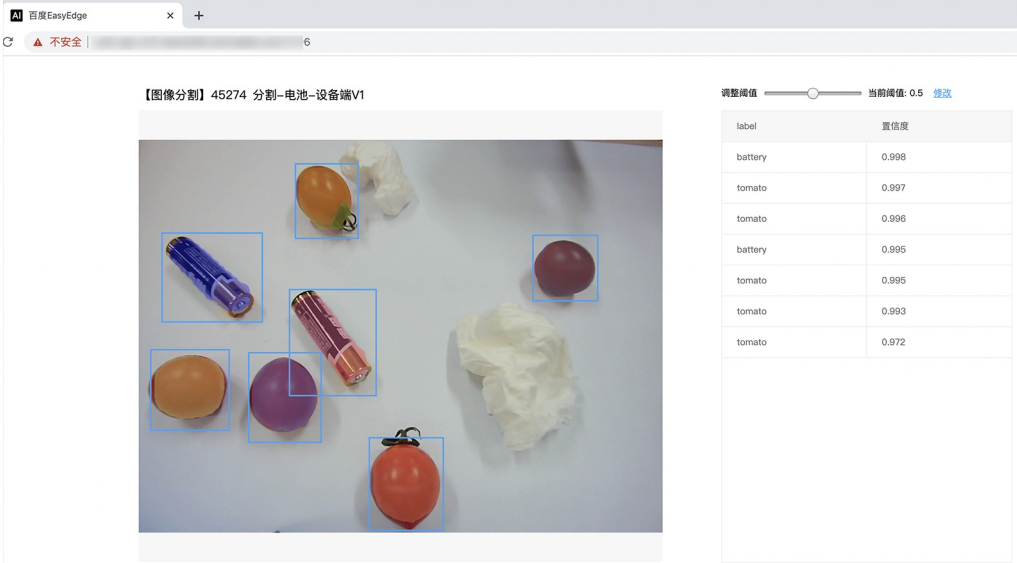
```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

🔗 2.3 序列号激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。



label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

使用说明

调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                        data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl



```

**include <sys/stat.h>**
**include <curl/curl.h>**

**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----|-----| | confidence | float | 0~1 | 检测的置信度 | | label | string | | 检测的类别 | | index | number | | 检测的类别 | | x1, y1 | float | 0~1 | 矩形的左上角坐标 (相对长宽的比例值) | | x2, y2 | float | 0~1 | 矩形的右下角坐标 (相对长宽的比例值) |

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

## 集成指南

### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

### 基于c++ dll集成

#### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

#### 集成方法

参考src目录中的CMakeLists.txt进行集成

### 基于c# dll集成

#### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

#### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
.NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

GPU依赖，版本必须如下：  
\* CUDA 11.0.x + cuDNN 8.4.x 或者 CUDA 11.7.x + cuDNN 8.4.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖，版本必须如下：  
\* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.x.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖，版本必须如下：  
\* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.3.1

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

其他问题 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：图像分类，物体检测，图像分割，目标追踪
- 硬件支持：
  - CPU 基础版: - intel x86\_64 \* - AMD x86\_64 - 龙芯 loongarch64 - 飞腾 aarch64
  - CPU 加速版 - Intel Xeon with Intel®AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE - AMD Core Processors with AVX2
  - NVIDIA GPU: x86\_64 PC
  - 寒武纪 Cambricon MLU270
  - 比特大陆计算卡SC5+
  - 百度昆仑XPU K200
    - x86\_64 - 飞腾 aarch64 - 百度昆仑XPU R200
    - x86\_64 - 飞腾 aarch64
  - 华为Atlas 300
  - 海光DCU: x86\_64 PC
  - 寒武纪 MLU370 on x86\_64
- 操作系统支持：Linux

根据开发者的选择，实际下载的版本可能是以下版本之一：

- EasyDL图像
  - x86 CPU 基础版
  - x86 CPU 加速版
  - Nvidia GPU 基础版
  - Nvidia GPU 加速版
  - x86 mlu270基础版
  - x86 SC5+基础版
  - Phytium MLU270基础版
  - Phytium XPU基础版
  - Phytium Atlas300I基础版
  - Hygon DCU基础版

性能数据参考[算法性能及适配硬件](#)

\*intel 官方合作，拥有更好的适配与性能表现。

## Release Notes

时间	版本	说明
2023.0 8.31	1.8.3	Atlas系列Soc支持语义分割模型，Atlas Cann升级到6.0.1，昆仑XPU后端推理引擎升级
2023.0 6.29	1.8.2	模型压缩能力升级
2023.0 4.24	1.8.1	支持物体检测自定义四边形模型精度无损压缩发布，CPU GPU 双CPU

5.17	1.8.1	支持物体检测自定义四边形框坐精度无损压缩及分布x86 CPU版SDK
2023.0 3.16	1.8.0	支持图像分类精度提升包本地部署
2022.1 2.29	1.7.2	模型性能优化；推理库性能优化
2022.1 0.27	1.7.1	新增语义分割模型http请求示例；升级海光DCU SDK，需配套rocm4.3版本使用；Linux GPU基础版下线适用于CUDA10.0及以下版本的SDK；Linux GPU加速版升级推理引擎版本
2022.0 9.15	1.7.0	Linux GPU加速版升级预测引擎；Linux GPU加速版适用于CUDA9.0、CUDA10.0的SDK为deprecated，未来移除；新增实例分割高性能模型离线部署；性能优化
2022.0 7.28	1.6.0	Linux CPU普通版、Linux GPU普通/加速版、Jetson新增目标追踪模型接入实时流的demo
2022.0 5.27	1.5.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2022.0 5.18	1.5.0	GPU加速版max_batch_size参数含义变更；修复GPU加速版并发预测时部分图片结果预测错误及耗时增加问题；CPU普通版预测引擎升级；新增版本号头文件；新增飞腾Atlas300I支持，并且在Easdl新增多种加速版本；示例代码移除frame_buffer，新增更安全高效的safe_queue；新增Tensor In/Out接口和Demo
2022.0 4.25	1.4.1	EasyDL, BML升级支持paddle2模型
2022.0 3.25	1.4.0	新增支持海光服务器搭配海光DCU加速卡；
2021.1 2.22	1.3.5	GPU加速版支持自定义模型文件缓存路径；新增支持飞腾MLU270服务器、飞腾XPU服务器
2021.1 0.20	1.3.4	CPU加速版推理引擎优化升级，新增支持飞腾CPU、龙芯CPU服务器、比特大陆计算卡SC5+ BM1684、寒武纪MLU270；大幅提升EasyDL GPU加速版有损压缩加速模型的推理速度
2021.0 8.19	1.3.2	CPU、GPU普通版及无损加速版新增支持EasyDL小目标检测，CPU普通版、GPU普通版支持检测模型的batch预测
2021.0 6.29	1.3.1	CPU普通版、GPU普通版支持分类模型的batch预测，CPU加速版支持分类、检测模型的batch预测；GPU加速版支持CUDA11.1；视频流解析支持调整分辨率；预测引擎升级
2021.0 5.13	1.3.0	新增视频流接入支持；模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告
2021.0 3.09	1.2.1	GPU新增目标追踪支持，http server服务支持图片通过base64格式调用，EasyDL高性能检测模型和均衡检测模型CPU加速版新增量化压缩模型
2021.0 1.27	1.1.0	EasyDL经典版分类高性能模型升级；部分SDK不再需要单独安装OpenCV
2020.1 2.18	1.0.0	1.0版本发布！安全加固升级、性能优化、引擎升级、接口优化等多项更新
2020.1 1.26	0.5.8	EasyDL经典版分类模型CPU加速版里新增量化压缩模型
2020.1 0.29	0.5.7	新增CPU加速版支持：EasyDL经典版高精度、超高精度物体检测模型和EasyDL经典版图像分割模型
2020.0 9.17	0.5.6	性能优化，支持更多模型
2020.0 8.11	0.5.5	提升预测速度；支持百度昆仑芯片
2020.0 5.15	0.5.3	优化性能，支持专业版更多模型
2020.0 4.16	0.5.2	支持CPU加速版；CPU基础版引擎升级；GPU加速版支持多卡多线程
2020.0 3.12	0.5.0	x86引擎升级；更新本地http服务接口；GPU加速版提速，支持批量图片推理

2020.0 1.16	0.4.7	ARM引擎升级；增加推荐阈值支持
2019.1 2.26	0.4.6	支持海思NNIE
2019.1 1.02	0.4.5	移除curl依赖；支持自动编译OpenCV；支持EasyDL 专业版 Yolov3; 支持EasyDL经典版高精度物体检测模型升级
2019.1 0.25	0.4.4	ARM引擎升级，性能提升30%；支持EasyDL专业版模型
2019.0 9.23	0.4.3	增加海思NNIE加速芯片支持
2019.0 8.30	0.4.2	ARM引擎升级；支持分类高性能与高精度模型
2019.0 7.25	0.4.1	引擎升级，性能提升
2019.0 7.25	0.4.0	支持Xeye, 细节完善
2019.0 6.11	0.3.3	paddle引擎升级；性能提升
2019.0 5.16	0.3.2	新增NVIDIA GPU支持；新增armv7l支持
2019.0 4.25	0.3.1	优化硬件支持
2019.0 3.29	0.3.0	ARM64 支持；效果提升
2019.0 2.20	0.2.1	paddle引擎支持；效果提升
2018.1 1.30	0.1.0	第一版！

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE 含义变更。变更前：预测输入图片数不大于该值均可。变更后：预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理，在输入图片数小于该值时依然可正常运行，但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值，尽可能保持最小。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES。更新模型时，如果SDK版本号有更新，请务必同时更新SDK，旧版本的SDK可能无法正确适配新发布出来部署包中的RES模型。

## 快速开始

SDK在以下环境中测试通过

- x86\_64, Ubuntu 16.04, gcc 5.4
- x86\_64, Ubuntu 18.04, gcc 7.4
- Tesla P4, Ubuntu 16.04, cuda 9.0, cudnn 7.5
- x86\_64, Ubuntu 16.04, gcc 5.4, XTCL r1.0
- aarch64, Kylin V10, gcc 7.3
- loongarch64, Kylin V10, gcc 8.3

- Bitmain SC5+ BM1684, Ubuntu 18.04, gcc 5.4
- x86\_64 MLU270 , Ubuntu 18.04, gcc 7.5
- phytium MLU270 , Kylin V10 , gcc 7.3.0
- phytium XPU , Kylin V10 , gcc 7.3.0
- hygon DCU, CentOS 7.8 gcc 7.3.0
- XPU K200, x86\_64, Ubuntu 18.04
- XPU K200 aarch64, Ubuntu 18.04
- XPU R200, x86\_64, Ubuntu 18.04
- XPU R200 aarch64, Ubuntu 18.04
- MLU370, x86\_64, Centos7.6.1810

依赖包括

- cmake 3+
- gcc 5.4 (需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.11 (可选)
- cuda && cudnn (使用NVIDIA-GPU时必须, SDK内提供多个Cuda版本推理套件, 根据需要安装依赖的Cuda和Cudnn版本)
- XTCL 1.0.0.187 (使用昆仑服务器时必须)
- Rocm4.3, Miopen 2.14(使用海光DCU服务器时必须)

## 1. 安装依赖

以下步骤均可选, 请开发者根据实际运行环境选择安装。

**(可选) 安装cuda&cudnn**

**在NVIDIA GPU上运行必须(包括GPU基础版, GPU加速版)**

对于GPU基础版, 若开发者需求不同的依赖版本, 请在[PaddlePaddle官网](#) 下载对应版本的libpaddle\_fluid.so或参考其文档进行编译, 覆盖lib文件夹下的相关库文件。

**(可选) 安装TensorRT**

**在NVIDIA GPU上运行GPU加速版必须**

下载包中提供了对应 cuda9.0、cuda10.0、cuda10.2、cuda11.0+四个版本的 SDK, cuda9.0 和 cuda10.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.0.0.11, cuda10.2 及以上的 SDK 默认依赖的 TensorRT 版本为 TensorRT8.4, 请在[这里](#)下载对应 cuda 版本的 TensorRT, 并把其中的lib文件拷贝到系统lib目录, 或其他目录并设置环境变量。

**(可选) 安装XTCL 使用昆仑服务器及对应SDK时必须** 请安装与1.0.0.187版本兼容的XTCL。必要时, 请将运行库路径添加到环境变量。

**(可选) 安装Rocm、Miopen**

**使用海光DCU服务器对应SDK时必须**

海光DCU SDK依赖Rocm 4.3和Miopen 2.14版本, 推荐使用easyedge镜像

(registry.baidubce.com/easyedge/hygon\_dcu\_infer:1.0.2.rocm4.3), SDK镜像内运行, 镜像拉取方式(wget https://aipe-easyedge-public.bj.bcebos.com/dcu\_docker\_images/hygon\_dcu\_rocm4.3.tar.gz && docker load -i hygon\_dcu\_rocm4.3.tar.gz), 关于海光DCU使用更多细节可参考[paddle文档](#)

**2. 使用序列号激活** 请在官网获取序列号



SDK内bin目录下提供预编译二进制文件，可直接运行(二进制运行详细说明参考下一小节)，用于图片推理和模型http服务，在二进制参数的serial\_num(或者serial\_key)处填入序列号可自动完成联网激活（请确保硬件首次激活时能够连接公网，如果确实不具备联网条件，需要使用纯离线模式激活，请下载使用百度智能边缘控制台纳管SDK)

```

**SDK内提供的一些二进制文件，填入序列号运行可自动完成激活，以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}

```

如果是基于源码集成，设置序列号方法如下

```

global_controller()->set_licence_key("")

```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量或者源码设置）实例数鉴权环境变量设置方法

```

export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30

```

实例数鉴权源码设置方法

```

global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)

```

### 3. 基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；在Intel CPU上运行CPU加速版，如果thirdparty里包含openvino文件夹的，必须在编译或运行demo程序前执行以下命令：source \${c\_cpp\_kit位置路径}/thirdparty/openvino/bin/setupvars.sh 或者执行 source \${c\_cpp\_kit位置路径}/thirdparty/openvino/setupvars.sh(openvino-2022.1+) 如果SDK内不包含setupvars.sh脚本，请忽略该提示

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号首次激活需要使用，激活后可不用填序列号也能运行二进制)

```

**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=./lib ./easyedge_image_inference ../../RES /xxx/cat.jpeg "1111-1111-1111-1111"

```

demo运行效果：



```
> ./easyedge_image_inference ../.././RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

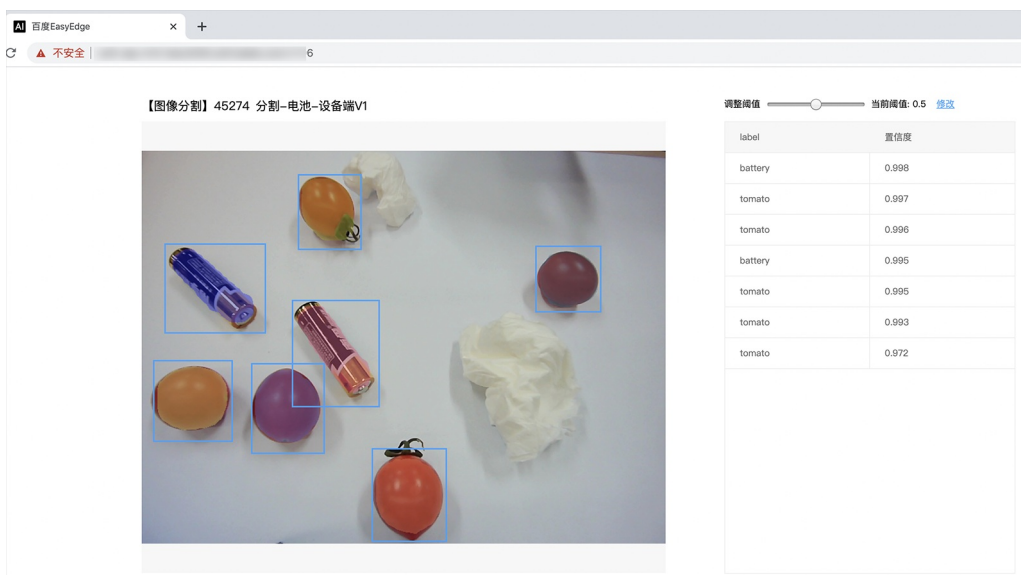
```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=./lib ./edgekit_serving --cfg=./edgekit_serving.yml

**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=./lib ./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

```
HTTP(or Webservice) is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，http://{设备ip}:24401，选择图片来进行测试，网页右侧会展示模型推理结果



对于目标追踪的模型，请选择一段视频，并耐心等待结果



同时，可以调用HTTP接口来访问服务。

请求http服务 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

python示例代码如下



```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }

        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

### 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。编译demo项目 SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```

cd src
mkdir build && cd build
cmake .. && make
./easiedge_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easiedge_image_inference {模型RES文件夹} {测试图片路径}

```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEDGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 1: 配置模型资源目录
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 在这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}
}
```

## 输入图片不限制大小

**SDK参数配置** SDK的参数通过 EdgePredictorConfig::set\_config和global\_controller()->set\_config配置。set\_config的所有key在easyedge\_xxxx\_config.h中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过EdgePredictorConfig::set\_config设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过global\_controller()->set\_config设置

以序列号为例，KEY的说明如下：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";
```

使用方法如下：

```
EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");
```

具体支持的运行参数配置列表可以参考开发工具包中的头文件的详细说明。

相关配置均可以通过环境变量的方法来设置，对应的key名称加上前缀EDGE\_即为环境变量的key。如序列号配置的环境变量key为EDGE\_PREDICTOR\_KEY\_SERIAL\_NUM，如指定CPU线程数的环境变量key为EDGE\_PREDICTOR\_KEY\_CPU\_THREADS\_NUM。注意：通过代码设置的配置会覆盖通过环境变量设置的值。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image, std::vector<std::vector<EdgeResultData>>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测、图像分割时才有意义
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割的模型, 该字段才有意义
    // 请注意: 图像分割时, 以下两个字段会比较大, 使用完成之后请及时释放EdgeResultData
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask

    // 目标追踪模型, 该字段才有意义
    int trackid; // 轨迹id
    int frame; // 处于视频中的第几帧
    EdgeTrackStat track_stat; // 跟踪状态
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

cv::Mat mask为图像掩码的二维数组

```

{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}

```

其中1代表为目标区域, 0代表非目标区域

### 关于图像分割mask\_rle

该字段返回了mask的游程编码, 解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding, 此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};      // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的is_needed置为false
    int input_fps{0};             // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};      // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;        // frame存储为视频文件的路径
    bool save_all{false};        // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

#### http服务

1. 开启http服务 http服务的启动可以参考`demo_serving.cpp`文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

#### 2. http接口详细说明

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片或视频来进行测试。

http 请求方式一：无额外编码 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
```

Python请求示例 (视频测试, 注意: 区别于图片预测, 需指定Content-Type; 否则会调用图片推理接口)

```
import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data).json()
```

http 请求方法二: json格式, 图片传base64格式字符串 HTTP方法: POST Header如下:

参数	值
Content-Type	application/json

Body请求填写:

- 图像分类网络: body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据, base64编码, 要求base64图片编码后大小不超过4M,最短边至少15px,最长边最大4096px,支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量, 不填该参数, 则默认返回全部分类结果

- 物体检测和实例分割网络: Body请求示例:



```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)

```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型，同其他CV模型不同，语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
        with open("gray_result.png", "wb") as fb:
            fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    },
  ]
}
```

其他配置

### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



### 2. CPU线程数设置

CPU线程数可通过 EdgePredictorConfig::set\_config配置

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_CPU_THREADS_NUM, 4);
```

### 3. 批量预测设置

```
int batch_size = 2; // 使用前修改batch_size再编译、执行
while (get_next_batch(imgs, img_files, batch_size, start_index)) {
  ...
}
```

**GPU 加速版 预测接口** GPU 加速版 SDK 除了支持上面介绍的通用接口外，还支持图片的批量预测，预测接口如下：

```

/**
 * @brief
 * GPU加速版批量图片推理接口
 * @param image: must be BGR, HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& result
) = 0;

/**
 * @brief
 * GPU加速版批量图片推理接口, 带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;

```

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE`，其含义见下方参数配置接口的介绍。

**运行参数选项** 在上面的内容中我们介绍了如何使用 `EdgePredictorConfig` 进行运行参数的配置。针对GPU加速版开发工具包，目前 `EdgePredictorConfig` 的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产生的max_batch_size不相等时，则重新编译模型（推荐）
 * 2: 无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
 * 值类型: int
 * 默认值: 1
 */

```

```

static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名, 默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置; 序列号不设置留空时, SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**: 首次加载模型会先对模型进行编译优化, 通过此值可以设置优化后的产出文件名, 这在多进程加载同一个模型的时候是有用的。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**: 首次加载模型经过编译优化后, 产生的优化文件会存储在这个位置, 可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**: 设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**: 此值用来控制批量图片预测可以支持的最大图片数, 实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**: 设置需要使用的 GPU 卡号。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**: 模型编译等级。通常模型的编译会比较慢, 但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制。当此值为 0 时, 表示忽略当前设置的 `max_batch_size` 而仅使用历史产出 (无历史产出时则编译模型); 当此值为 1 时, 会比较历史产出和当前设置的 `max_batch_size` 是否相等, 如不等, 则重新编译; 当此值为 2 时, 无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**: 通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量, 其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源, 建议结合实际使用控制此值, 使用多少则设置多少。注意: 此值的增加会降低单次 infer 的速度, 建议优先考虑 batch inference 和 multi predictor。

**PREDICTOR\_KEY\_GTURBO\_FP16**: 默认是 fp32 模式, 置 true 可以开启 fp16 模式预测, 预测速度会有所提升, 但精度也会略微下降, 权衡使用。注意: 不是所有模型都支持 fp16 模式。目前已知不支持fp16的模型包括: 图像分类高精度模型。

**多线程预测** GPU 加速版 SDK 的多线程分为单卡多线程和多卡多线程两种。单卡多线程: 创建一个 predictor, 并通过

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY** 控制单卡所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。多卡多线程: 多卡的

支持是通过创建多个 predictor，每个 predictor 对应一张 GPU 卡，predictor 的创建和 init 的调用放在主线程，通过多线程的方式调用 infer 接口。

**已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时，部分结果错误** A：EasyDL图像分类高精度模型在有些显卡上可能存在此问题，可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

**2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object** A：部分显卡存在此问题，如果遇到此问题，请确认没有频繁调用 init 接口，通常调用 infer 接口即可满足需求。

**3. 开启 fp16 后，预测结果错误** A：不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括：图像分类高精度模型。目前不支持的将在后面的版本陆续支持。

**昆仑服务器** 昆仑服务器SDK支持将EasyDL的模型部署到昆仑服务器上。SDK提供的接口风格一致，简单易用，轻松实现快速部署。Demo的测试可参考上文中的测试Demo部分。

**参数配置接口** 在上面的内容我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对昆仑服务器开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * 使用哪张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 设置需要同时预测的图片数量
 * 值类型：int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_KUNLUN_BATCH_SIZE = "PREDICTOR_KEY_KUNLUN_BATCH_SIZE";
```

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的加速卡的卡号。

**PREDICTOR\_KEY\_KUNLUN\_BATCH\_SIZE**：设置单次预测可以支持的图片数量。

使用方法：

```
int batch_size = 1;
config.set_config(easyedge::params::PREDICTOR_KEY_KUNLUN_BATCH_SIZE, batch_size);
```

**模型调优** 通过设置如下环境变量，可以在初始化阶段对模型调优，从而让预测的速度更快。

```
export XPU_CONV_AUTOTUNE=5
```

## FAQ

### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

## 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

## 3. NVIDIA GPU预测时，报错显存不足 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请根据显存大小和模型配置。调整合适的初始 fraction\_of\_gpu\_memory。参数的含义参考[这里](#)。

## 4. 如何将我的模型运行为一个http服务？目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

## 5. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

## 6. 运行SDK报错 Authorization failed

情况一：日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

## 7. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

## 8. 运行二进制时，提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 运行二进制时提示 libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 同上面8的问题类似，没有正确设置动态库的查找路径，可通过设置LD\_LIBRARY\_PATH为sdk的thirdparty/opencv/lib文件夹解决

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/thirdparty/opencv/lib
(tips: 上面冒号后面接的thirdparty/opencv/lib路径以实际项目中路径为准, 比如也可能是../thirdparty/opencv/lib)
```

10. 编译时报错: **file format not recognized** 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中, 再解压缩、编译

11. 进行视频解码时, 报错符号未找到、格式不支持、解析出的图片为空、无法设置抽帧 请确保安装OpenCV时, 添加了-DWITH\_FFMPEG=ON选项 (或者GStream选项), 并且检查OpenCV的安装日志中, 关于Video I/O段落的说明是否为YES。

```
-- Video I/O:
-- DC1394:          YES (ver 2.2.4)
-- FFMPEG:          YES
-- avcodec:         YES (ver 56.60.100)
-- avformat:        YES (ver 56.40.101)
-- avutil:          YES (ver 54.31.100)
-- swscale:         YES (ver 3.1.101)
-- avresample:      NO
-- libv4l/libv4l2:  NO
-- v4l/v4l2:        linux/videodev2.h
```

如果为NO, 请搜索相关解决方案, 一般为依赖没有安装, 以apt为例:

```
apt-get install yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
```

12. GPU加速版运行有损压缩加速的模型, 运算精度较标准模型偏低 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除, 并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true, 使用FP16的运算精度重新评估模型效果。若依然不理想, 可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false, 从而使用更高精度的FP32的运算精度。

## Linux集成文档-Python

### 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法, 适用于 EasyDL 和 BML。

EasyDL 通用版:

- 网络类型支持: 图像分类, 物体检测, 图像分割, 声音分类, 表格预测
- 硬件支持:
  - Linux x86\_64 CPU (基础版, 加速版)
  - Linux x86\_64 Nvidia GPU (基础版, 加速版)
- 语言支持: Python 3.5, 3.6, 3.7

BML:

- 网络类型支持: 图像分类, 物体检测, 声音分类
- 硬件支持:
  - Linux x86\_64 CPU (基础版)
  - Linux x86\_64 Nvidia GPU (基础版)
- 语言支持: Python 3.5, 3.6, 3.7

### Release Notes

时间	版本	说明
2023-03-16	1.3.7	迭代升级，新增支持文本类模型；新增GPU 多卡多进程推理demo
2022.10.27	1.3.5	新增华为Atlas300、飞腾Atlas300 Python SDK，支持图像分类、物体检测、人脸检测、实例分割
2022.09.15	1.3.3	EasyDL CPU普通版新增支持表格预测
2022.05.27	1.3.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2021.12.22	1.2.7	声音分类模型升级
2021.10.20	1.2.6	CPU基础版、CPU加速版、GPU基础版推理引擎优化升级
2021.08.19	1.2.5	CPU基础版、CPU无损加速版、GPU基础版新增支持EasyDL小目标检测
2021.06.29	1.2.4	CPU、GPU新增EasyDL目标跟踪支持；新增http server服务启动demo
2021.03.09	1.2.2	EasyDL CPU加速版新增支持分类、高性能检测和均衡检测的量化压缩模型
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.03.12	1.1.14	新增声音识别python sdk
2020.02.12	1.1.13	新增口罩模型支持
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持 SDK 加速版
2019.12.04	1.1.10	支持图像分割
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.08.29	1.1.8	CPU 加速版支持
2019.07.19	1.1.7	提供模型更新工具
2019.05.16	1.1.3	NVIDIA GPU 支持
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】序列号的配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。
- 使用声音分类SDK需要安装额外依赖 \* pip 安装 `resampy pydub six librosa` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已基于sdk中无需额外安装，linux系统需要手动安装）
- 使用表格预测SDK需要安装额外依赖 `pip安装brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3 catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2 psutil==5.7.2 pymml==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0` 另外ml算法安装（目前只支持python3.7） `pip install BaiduAI_TabularInfer-0.0.0-cp37-cp37m-linux_x86_64.whl` 安装 **paddlepaddle**
- 使用x86\_64 CPU 基础版 预测时必须安装（目标跟踪除外）：



```
python -m pip install paddlepaddle==2.2.2 -i https://mirror.baidu.com/pypi/simple
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU 基础版预测时必须安装（目标跟踪除外）：

```
python -m pip install paddlepaddle-gpu==2.2.2.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA10.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2 -i https://mirror.baidu.com/pypi/simple #CUDA10.2的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post110 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.0的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post111 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post112 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.2的PaddlePaddle
```

不同cuda版本的环境，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持安装 OpenVINO 使用x86\_64 CPU 加速版 SDK 预测时必须安装。

- 1) 请参考 [OpenVINO toolkit 文档](#)安装 2021.4版本，安装时可忽略Configure the Model Optimizer及后续部分
- 2) 运行之前，务必设置环境变量

```
source /opt/intel/opencvino_2021/bin/setupvars.sh
```

#### 安装 cuda、cudnn

- 使用Nvidia GPU 加速版预测时必须安装。依赖的版本为 cuda9.0、cudnn7。版本号必须正确。

#### 安装 pytorch (torch >= 1.7.0)

- 目标跟踪模型的预测必须安装pytorch版本1.7.0及以上（包含：Nvidia GPU 基础版、x86\_64 CPU 基础版）。
- 目标跟踪模型Nvidia GPU 基础版还需安装依赖cuda、cudnn。

关于不同版本的pytorch和CUDA版本的对应关系：[pytorch官网](#) 目标跟踪模型还有一些列举在requirements.txt里的依赖（包括torch >= 1.7.0），均可使用pip下载安装。

```
pip3 install -r requirements.txt
```

## 2. 安装 easyedge python wheel 包 安装说明

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。安装说明：[华为 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Atlas300-{版本号}-cp36-cp36m-linux_x86_64.whl
```

安装说明：[飞腾 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Phytium.Atlas-{版本号}-cp36-cp36m-linux_aarch64.whl
```

## 3. 使用序列号激活



## 获取序列号

此处发布、下载的SDK为未授权SDK，需要前往控制台获取序列号激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标test	134318-V1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英特尔GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
		MIPS Linux	基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布		

## 修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

## 4. GPU 加速版 使用 GPU 加速版，在安装完 whl 之后，必须：

1. 从[这里](#)下载 TensorRT7.0.0.11 for cuda9.0，并把解压后的 lib 放到 C++ SDK 的 lib 目录或系统 lib 目录
2. 运行时，必须在系统库路径中包含 C++ SDK 下的lib目录。如设置LD\_LIBRARY\_PATH

```
cd ${SDK_ROOT}
```

### \*\*1. 安装 python wheel 包\*\*

```
tar -xzf python/*.tar.gz
pip install -U {对应 Python 版本的 wheel 包}
```

### \*\*2. 设置 LD\_LIBRARY\_PATH\*\*

```
tar -xzf cpp/*.tar.gz
export EDGE_ROOT=$(readlink -f $(ls -h | grep "baidu_easyedge_linux_cpp"))
export LD_LIBRARY_PATH=$EDGE_ROOT/lib
```

### \*\*3. 运行 demo\*\*

```
python3 demo.py {RES文件夹路径} {测试图片路径}
```

如果是使用 C++ SDK 自带的编译安装的 OpenCV，LD\_LIBRARY\_PATH 还需要包括 C++ SDK 的 build 目录下的 `thirdparty/lib` 目录

如果没有正确设置 LD\_LIBRARY\_PATH，运行时可能报错：

```
ImportError: libeasyedge.so.0.4.3: cannot open shared object file: No such file or directory
ImportError: libopencv_core.so.3.4: cannot open shared object file: No such file or directory
```

## 5. 测试 Demo

### 5.1 图片预测

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



### 5.2 视频预测（适用于目标跟踪）

输入对应的模型文件夹（默认为RES）和测试视频文件路径 / 摄像头id / 网络视频流地址，运行：

```

**video_type: 输入源类型 type:int**
**1 本地视频文件**
**2 摄像头的index**
**3 网络视频流**
**video_src: 输入源地址, 如视频文件路径、摄像头index、网络流地址 type: string**
python3 demo.py {model_dir} {video_type} {video_src}

```

6. 测试Demo HTTP 服务 输入对应的模型文件夹（默认为RES）、序列号、设备ip和指定端口号，运行：

```
python3 demo_serving.py {model_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

后，会显示：

```
Running on http://0.0.0.0:24401/
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里 `http_client_test()`函数请求http服务进行推理。

【图像分割】 45274 分割-电池-设备端V1

调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

图片加载失败

## 使用说明

使用流程 `demo.py`

```

import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image((numpy.ndarray的图片))
pred.close()

```

`demo_serving.py`

```

import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir={RES文件夹路径}, license=serial_key)
**请参考同级目录下demo.py里:**
**pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)**
**对以下参数device\device_id和engine进行修改**
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)

```

## 初始化

- 接口

```
def init(self,
    model_dir,
    device=Device.CPU,
    engine=Engine.PADDLE_FLUID,
    config_file='conf.json',
    preprocess_file='preprocess_args.json',
    model_file='model',
    params_file='params',
    label_file='label_list.txt',
    infer_cfg_file='infer_cfg.json',
    device_id=0,
    thread_num=1
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device, 比如 : Device.CPU
        engine: BaiduAI.EasyEdge.Engine, 比如 : Engine.PADDLE_FLUID
        config_file: str
        preprocess_file: str
        model_file: str
        params_file: str
        label_file: str 标签文件
        infer_cfg_file: 包含预处理、后处理信息的文件
    device_id: int 设备ID
        thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success
    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

## 预测图像

- 接口

```
def infer_image(self, img,
                threshold=0.3,
                channel_order='HWC',
                color_format='BGR',
                data_type='numpy'):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

### 预测视频（目前仅限目标跟踪模型调用）

- 接口

```
def infer_frame(self, frame, threshold=None):
    """
    视频推理(抽帧之后)
    :param frame:
    :param threshold:
    :return:
    """
```

- 返回格式dict

字段	类型	说明
pos	dict1	当前帧每一个类别的追踪目标的像素坐标(tlwh)
id	dict2	当前帧每一个类别的追踪目标的id
score	dict3	当前帧每一个类别的追踪目标的识别置信度
label	dict4	class_idx(int)与label(string)的对应关系
class_num	int	追踪类别数

### 预测声音

- 使用声音分类SDK需要安装额外依赖 `pip 安装 resampy pydub` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已集成在sdk中无需额外安装，linux系统需要手动安装）

- 接口

```
def infer_sound(self, sound_binary,
                threshold=0.3):
    """

    Args:
        sound_binary: sound_binary
        threshold: confidence

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类的置信度
label	string		分类的类别
index	number		分类的类别

**升级模型** 适用于经典版升级模型，执行`bash update_model.sh`，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

#### FAQ

**Q: EasyDL 离线 SDK 与云服务效果不一致，如何处理？** A: 后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**Q: 运行时报错 "非法指令" 或 "illegal instruction"** A: 可能是 CPU 缺少 `avx` 指令集支持，请在[paddle官网](#) 下载 `noavx` 版本覆盖安装

**Q: NVIDIA GPU预测时，报错显存不足：** A: 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过`export FLAGS_fraction_of_gpu_memory_to_use=0.3`来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 `allocate` 更多的显存。

**Q: 我想使用多线程预测，怎么做？** 如果需要多线程预测，可以每个线程启动一个Program实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的`thread_num`参数。

#### Q: 运行SDK报错 Authorization failed

**情况一：日志显示 `Http perform failed: null respond`** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受`HTTP_PROXY` 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示`failed to get/check device id(xxx)`或者`Device fingerprint mismatch(xxx)`** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

情况三：Atlas Python SDK日志提示 `ImportError: libavformat.so.58: cannot open shared object file: No such file or directory` 或者其他类似so找不到 可以在 `LD_LIBRARY_PATH` 环境变量加上 `libs` 和 `thirdpartylibs` 路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内 `libs` 和 `thirdpartylibs` 路径的方法如下(以华为Atlas300 SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas300 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## Linux集成文档-Atlas

### 简介

本文档介绍EasyEdge/EasyDL的Linux Atlas SDK的使用方法。

注意Atlas有两种产品形态，Atlas 200和Atlas 300，请参见此处的[文档说明](#)

- 网络类型支持：图像分类
- 硬件支持：
  - CPU: aarch64
  - Atlas 300 卡
- 操作系统支持：Atlas指定的Linux版本，Ubuntu 16.04 x86\_64 或 centos 7 x86\_64，请从Atlas文档中下载。

### Release Notes

时间	版本	说明
2020.3.23	0.1	初始版本，支持图像分类

### 性能数据

数据仅供参考，实际数值根据使用线程数、利用率等情况可能有所波动

模型类型	模型算法	芯片类型	SDK类型	实测硬件	单次预测耗时
EasyDL 图像分类	高性能	Atlas 300	Atlas 300	Atlas 800服务器	9ms
EasyDL 图像分类	高精度	Atlas 300	Atlas 300	Atlas 800服务器	12ms
EasyDL 物体检测	高性能	Atlas 300	Atlas 300	Atlas 800服务器	11ms
EasyDL 物体检测	高精度	Atlas 300	Atlas 300	Atlas 800服务器	31ms

### atlas 300 加速卡注意事项

一般服务器（HOST侧）安装多个300加速卡，每个300加速卡有4个芯片。一个芯片（DEVICE侧）可以认为是一个单独的系统，并且不共享储存系统。

每个芯片都有独立的device-id，可以通过命令查看：`sudo npu-smi info`

由于模型需要在芯片上运行。因此运行SDK前，需要手动将模型复制到每个单独芯片的储存系统上。

### 测试atlas 300的官方demo

#### 环境准备

请参见此处的[文档说明](#)，搭建环境，测试HelloDavinci demo通过后，再测试本demo

#### 修改300加速卡SSH密码（可选）

请在咨询华为技术人员后，修改Device登录密码



```
ssh HwHiAiUser@192.168.1.199
**登录后会强制修改密码**
ssh HwHiAiUser@192.168.1.198
```

## 快速开始

SDK在以下环境中测试通过

- ubuntu 16.04, Atlas 800 服务器指定版本;

Atlas DDK 的ddk\_info信息：

```
{
  "VERSION": "1.3.8.B902",
  "NAME": "DDK",
  "TARGET": "ASIC"
}
```

## 1. 安装软件

```
sudo apt-get install sshpass build-essential
```

## 2. 测试Demo

编译运行：

下载后，模型资源文件默认已经打包在开发者下载的SDK包中，

Step 0：使用HwHiAiUser登录

Step 1：运行一次install-demo.sh脚本，会得到测试demo。

Step 2：请在官网获取序列号，填写在demo\_async.cpp及demo\_sync.cpp的开始处license\_key字段。



图片加载失败

step3：准备测试图片

覆盖image目录下的 1.jpg，更多图片可以用于demo中的批量测试模式

step4(可选)：修改test\_300.sh下的以下开发板登录信息

```
export DDK_PATH=$HOME/tools/che/ddk/ddk # ddk的安装路径

declare -a DIVICE_IPS=("192.168.1.199") # 300加速卡芯片的ip地址，device=0 对应192.168.1.199
DEVICE_PASSWORD="Huawei@SYS3" # 之前 修改300加速卡SSH密码
MAIN_CPP="demo_async.cpp" # demo_async.cpp" 异步接口，“demo_async.cpp” 同步接口

OpenCV_install_dir=/home/HwHiAiUser/opencv_x64/ # OpenCV 3.4版本，需要存在
${OpenCV_install_dir}/share/OpenCV/OpenCVConfig.cmake文件
```

step5: 运行demo，会自动编译OpenCV 3.4库，如果报错请自行编译，目录设置在 OpenCV\_install\_dir

```
cd demo
sh test_300.sh
```

图像分类demo运行效果：

```
[stat] [100001]image/1.jpg(4 images) time used: 41ms (at 1583765958531) total:705ms
[result][100001]image/1.jpg[281470472005664] is: n07747607 orange 0.973633 950;
```

n07747607 orange 分类名  
0.973633 分类概率  
950 分类名的序号

物体检测的demo运行效果：

```
[stat] time used : 101ms; all time used:478
images[3] result:
label:no2_ynen;prob:0.985352 loc:[(0.459961,0.839844), (0.5625,0.988281)]

no2_ynen 分类名 ， 也可以获取分类名的序号
0.985352 分类概率
loc:[(0.459961,0.839844), (0.5625,0.988281)]， 检测框的位置。(0.459961,0.839844表示左上角的点，(0.5625,0.988281)右下角的点；
如原始图片608， 左上角(0.459961*608,0.839844*608)， 右下角(0.5625*608,0.988281*608)
```

## SDK接口使用

使用该方式，将运行库嵌入到开发者的程序当中。

### 同步接口使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ;
auto predictor = global_controller()->CreateEdgePredictor(config);
int ret = predictor->init();
# 若返回非0，请查看输出日志排查错误原因。
auto img = cv::imread({图片路径});
// step 3: 预测图像
std::vector<EdgeResultData> result2;
predictor->infer(img, result2);
# 解析result2即可获取结果
```

### 异步接口使用流程

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 3: 创建Predictor ; 这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 4: 设置异步回调
predictor->set_result_handler(YOUR_HANDLER);

// step 5: 初始化
int ret = predictor->init();
**若返回非0, 请查看输出日志排查错误原因。 **

// step 6: 预测图像
auto img = cv::imread({图片路径});
color_format = kBGR;
float threshold = 0.1;

uint64_t seq_id;
predictor->infer_async(img, color_format, 0.1, nullptr, seq_id);
**YOUR_HANDLER里面有seq_id的回调结果**

```

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

- 接口

```
virtual int set_licence_key(const std::string& license) = 0;
```

### 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

### 日志及报错

#### 日志

日志需要开启Atlas 的 INFO级别，/etc/slog.conf中配置关闭zip格式。清空/var/dlog 目录，运行atlas 300 [官方示例代码](#)，可以在/var/dlog目录下看见host和device开头的2个日志文件，中间是明文的info级别的日志

日志共有3处：

- host 测的easyedge.log。当前运行目录下。
- device侧的easyedge.device.xxx.log。 device侧的日志，在芯片的同名目录下。
- /var/dlog host 与device开头的log文件， ddk运行日志，其中device侧有略微延时

#### 通用错误码

错误码	常量	解释
1000004	RESOURCE_LOAD_FAILED	缺少data/model/conf.json文件或者该文件以及被改动。下载包中的data/model下的所有文件都不要改动，尝试使用默认配置。或者按照报错复制到对应目录。
7000001	AUTH_FAILED	服务端校验序列号失败
7000002	AUTH_LICENSE_INVALID	校验序列号
7000003	AUTH_LICENSE_EXPIRED	序列号过期
5000001	NET_CURL_PERFORM_FAILED	服务端校验序列号的请求因为网络原因失败
6000001	GET_MACHINE_ID_FAILED	没有相关权限，请反馈

**Atlas SDK 错误码**

错误码值	常量	含义	报错示例信息	示例解释及解决方式
12000011	FILE_NOT_READABLE	资源文件不可读	data/model/params IS NOT READABLE	data/model/params，这个文件不可读。SDK下载包中的data/model下的所有文件都不要改动，尝试使用默认配置。或者按照报错复制到对应目录。
12000012	HIAI_ERRORLIST_FILE	status.h.list不是原始文件	data/model/status.h.list IS TOO SMALL	下载包中的data/model下的所有文件都不要改动，包括status.h.list
12000102	PREDICTOR_NOT_INITED	create后没有调用init()函数	please call init() first	调用infer函数前没有调用init()
12000103	PREDICTOR_NO_HANDLER	create后没有调用set_result_handler()函数	please call set_result_handler() first	调用infer_async函数前没有调用set_result_handler(),建议init前调用
12000104	PREDICTOR_ALREADY_INITED	init()不管是否成功，不能连续调用。	don't call init() more than once	如果失败，请再次新建一个Predictor
12000105	BATCH_SIZE	AtlasConfig里的batch_size设置与model_name不符合	model batch size is 1; your config batch size is 4	batch_size设置里4，model_name设置里params，不对应导致报错。model_name应该设置为params-batch4
12000106	INPUT_WIDTH	preprocess_args.json被改动	model input tensor width is 224; your config resize is 226	请勿修改preprocess_args.json
12000107	INPUT_HEIGHT	同上	同上	同上
12000201	BATCH_TOO_MANY_IMAGES	一次输入的图片大于batch_size	too_many_images input:2; batch_size is 1	调用infer函数，输入了2张图片，大于batch数。如果batch=1的话，每次infer只能传一张图。
12000202	IMAGE_FORMAT_CHANNELS	infer函数输入的color_format与cv::Mat里的channel数不匹配	EdgeColorFormat is not according to cv channels; format is 101; channels is 3; seq_id1	101表示kRGBA，cv::Mat里channel应该期望是4。如果是直接读的图片，填kBGR。
12200001	ENGINE_MATRIX_COMMON	Atlas DDK Matrix部分（非CreateGraph函数）接口报错。即返回值HIAI_StatusT不是HIAI_OK。具体解释见Atlas官方文档。	hiai::Graph::ParseConfigFile(graph.prototxt); status Code is 16855066; HIAI ERROR CODE is 101 HIAI_GRAPH_PROTO_FILE_PARSE_FAILED_CODE,	调用hiai::Graph::ParseConfigFile()返回16855066，对应的status.h.list中的错误码是101。保留日志，具体见Atlas官方文档。
12200002	ENGINE_AI_COMMON	Atlas DDK Device引擎部分 hiai::AIStatus 不为hiai::SUCCESS	_ai_model_manager->Process()	保留日志，具体见Atlas官方文档。
12200003	ENGINE_MATRIX_INIT	Atlas DDK CreateGraph() 初始化DDK报错。具体解释见Atlas官方文档。	hiai::Graph::CreateGraph(); data/model/graph.prototxt; status Code is 16855190; HIAI ERROR CODE is 225 HIAI_FILE_NOT_EXIST_CODE,	示例为缺少libatlas_device.so导致
12200004	EDGEATLAS_ENGINE_MATRIX_INIT_DEVICE	Atlas DDK CreateGraph() 初始化DDK报错，这个报错很可能是device侧出现问题	hiai::Graph::CreateGraph() ; data/model/graph.prototxt; status Code is 16855057; HIAI ERROR CODE is 92 HIAI_GRAPH_ENGINE_INIT_FAILED_CODE	需要具体排查DEVICE侧日志再次找具体报错，发现原因
12200005	ENGINE_ARGS_NULL	内部错误		请反馈
12300001	SYNC_INFER_TIMEOUT	调用infer同步接口时，内部会调用infer_async函数，这个函数超时	infer sync wait timeout more than 10ms	内部会调用infer_async函数超过10ms。1. 不要并发过高 2. 超时参数略微大些。

## 🔗 图像分类服务器端SDK集成文档-EdgeKitProxy

### 简介

本文档介绍EdgeKitProxy的使用方法。

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-05-17 | 1.0.0 | 第一版 ! |

### 快速开始

#### 二进制位置

位于SDK内bin目录中，文件名为edgekit\_serving，配套edgekit\_serving.yml为默认配置文件

### 注意事项

请参考各SDK文档中的注意事项

### 使用说明

### 服务启动

```
usage: edgekit_serving [<flags>]
```

#### Flags:

```
--help          显示帮助
-c, --cfg=./edgekit_serving.yml
                配置文件
-m, --model_dir=./RES    模型目录
-s, --serial_num=ABCD-EFGH-IJKL-MNOP
                序列号
--pool_min_size=1    预测池最小预测器个数
--pool_max_size=1    预测池最大预测器个数
--pool_full_interval_seconds=1
                    预测池满载多少秒进行扩容
--pool_idle_interval_seconds=1
                    预测池未满载多少秒进行缩容
--pool_available_device=1 ...
                    预测池可用设备列表
-d, --debug          开启debug模式
--log_to_std          日志输出至终端
--log_to_file          日志输出至文件
--log_file=easyedge.log  日志文件名
--log_max_size=10     日志最大大小 (MB)
--log_max_age=10      日志旧文件保留天数
--log_max_backups=100  日志旧文件保留个数
-h, --host=127.0.0.1   服务监听地址
-p, --port=24401       服务监听端口
--ws_max_handle_num=1  websocket接口最大处理请求个数
--ws_max_handle_timeout=30
                    websocket接口超时时间
```

### 配置文件说明

```
controller:
serialNum: AAAA AAAA AAAA AAAA # 序列号
modelDir: ../.././RES # 模型目录

predictorPool:
minSize: 1 # 预测池最小预测器个数
maxSize: 3 # 预测池最大预测器个数
fullIntervalSeconds: 1 # 预测池满载多少秒进行扩容
idleIntervalSeconds: 1 # 预测池未满载多少秒进行缩容
availableDevice: [-1] # 预测池可用设备列表

serving:
host: 0.0.0.0 # 服务监听地址
port: 24401 # 服务监听端口
enableHTTP: true # 对外开启HTTP服务
enableWS: false # 对外开启websocket服务
ws:
maxHandleNum: 1 # websocket接口最大处理请求个数
maxHandleTimeout: 30 # websocket接口超时时间

logging:
debug: true # 开启debug模式
logToStd: true # 日志输出至终端
logToFile: false # 日志输出至文件
logFile: easyedge.log # 日志文件名
maxSize: 10 # 日志最大大小 (MB)
maxAge: 10 # 日志旧文件保留天数
maxBackups: 100 # 日志旧文件保留个数
```

命令行参数会覆盖配置文件中同义配置

#### 服务调用

HTTP服务接口url: \${监听地址}/ HTTP服务接口url: \${监听地址}/ws

#### 请求参数

```

syntax = "proto3";

package easyedge.kit.proxy;

enum ImageType {
  Bin = 0; // 图片原始二进制内容，json格式下为base64编码后结果
  Mat = 1; // 图片Mat格式内容，json格式下为base64编码后结果
}

message HTTPRequest {
  bytes image = 1;
  ImageType image_type = 2;
  int32 height = 3;
  int32 width = 4;
  int32 channel = 5;
  float threshold = 6;
  int32 top_num = 7;
}

enum CommandType {
  GetInfo = 0;
  InferImage = 1;
}

enum InfoType {
  Hardware = 0;
}

message WebSocketRequest {
  string request_id = 1;
  CommandType command_type = 2;
  InfoType info_type = 3;
  bytes image = 4;
  ImageType image_type = 5;
  int32 height = 6;
  int32 width = 7;
  int32 channel = 8;
  int64 frame_id = 9;
  float threshold = 10;
  int32 top_num = 11;
}

```

## 返回参数

```

syntax = "proto3";

package easyedge.kit.proxy;

message BasicGPUInfo {
  string productName = 1;
  string memUsed = 2;
  string memTotal = 3;
  string gpuUtil = 4;
  string powerLimit = 5;
  string powerDraw = 6;
  string temperature = 7;
}

message DevStat {
  string name = 1;
  uint64 rx = 2;
  uint64 tx = 3;
}

message Chip {
  string name = 1;
  double powerUsed = 2;
  double powerLimit = 3;
  double temperature = 4;
}

```



```

double temperature = 4;
double chipUtil    = 5;
int64 memoryUsed  = 6;
int64 memoryTotal = 7;
}

message SMI {
  string name      = 1;
  string sdkVersion = 2;
  string driverVersion = 3;
  repeated Chip chips = 4;
}

message HInfo {
  string osName          = 1;
  string hostname        = 2;
  repeated string ipAddr = 3;
  repeated string macAddr = 4;
  uint64 bootTime       = 5;
  int32 cpuCores         = 6;
  double cpuMhz          = 7;
  string cpuModelName    = 8;
  double cpuUsage        = 9;
  map<string, double> cpuUsageDetail = 10;
  uint64 memTotal        = 11;
  uint64 memTotalUsed    = 12;
  double memUsage        = 13;
  map<string, double> memUsageDetail = 14;
  uint64 diskTotal       = 15;
  uint64 diskTotalUsed   = 16;
  double diskUsage       = 17;
  map<string, double> diskUsageDetail = 18;
  string userName        = 19;
  bool isInternetConnected = 20;
  string deviceId        = 21;
  int64 deviceTimestamp  = 22;
  map<string, DevStat> netUsageDetails = 23;
  repeated BasicGPUInfo gpuInfo      = 24;
  double gpuUtil                    = 25;
  uint64 gpuMemTotal                = 26;
  uint64 gpuMemTotalUsed            = 27;
  double gpuMemUsage                = 28;
  map<string, SMI> aiChiplInfo       = 29;
}

message LocationPoint {
  optional int32 x = 1;
  optional int32 y = 2;
}

message Location {
  optional int32 left    = 1;
  optional int32 top     = 2;
  optional int32 width   = 3;
  optional int32 height  = 4;
  repeated LocationPoint points = 5;
}

message Point {
  optional double x = 1;
  optional double y = 2;
}

message InferResultItem {
  optional int64 index = 1;
  optional double confidence = 2;
  optional double score = 3;
  optional string label = 4;
  optional string name = 5;
  optional int32 modelId = 6;
}

```

```

optional int32 modelKind = 6;

// 矩形检测
optional double x1 = 7;
optional double x2 = 8;
optional double y1 = 9;
optional double y2 = 10;
optional Location location = 11;

// 四边形检测
repeated Point points = 12;

// 追踪
optional int64 trackId = 13;
optional int64 frame = 14;
optional double fps = 15;

optional string mask = 16;
}

message HTTPResponse {
  int64 cost_ms = 1;
  int32 error_code = 2;
  int64 frame_id = 3;
  repeated InferResultItem results = 4;
}

message WebSocketInferResponse {
  string request_id = 1;
  int64 cost_ms = 2;
  int32 error_code = 3;
  int64 frame_id = 4;
  repeated InferResultItem results = 5;
  bytes annotated = 6; // 渲染后的图片原始二进制内容，json格式下为base64编码后结果，目前语义分割返回这个类型
}

message WebSocketHInfoResponse {
  string request_id = 1;
  int32 status = 2;
  string msg = 3;
  HInfo data = 4;
}

```

## 其他说明

### 单机负载均衡

通过配置文件或命令行参数配置了预测池相关配置后，若预测池最小与最大预测器个数不同，且扩缩容配置不为-1则开启单机负载均衡，服务启动时会创建最小数量的预测器，后续根据实际请求情况，若所有预测器均有负载的持续时间大于配置中的满载扩容时间，且预测器数量未到达最大个数时，会自动扩容，后续若请求并发数下降，预测器池中预测器不能跑满负载时，则会自动缩容，尽可能最大化利用单机资源

### 纯离线API集成说明

本文档主要说明定制化物体检测模型发布为服务器API（通过部署包实现）后如何使用。如还未训练模型，请前往[EasyDL](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### 部署包使用说明

#### 部署方法

EasyDL定制化物体检测模型的服务器API通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#)使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

使用方法：将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

## 授权说明

API部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

## 性能指标

物体检测模型可部署在CPU或GPU服务器上，单实例具体性能指标参见[算法性能及适配硬件 API参考](#)

## 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请部署包，部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/ObjectDetection](http://{IP}:{PORT}/{DEPLOY_NAME}/ObjectDetection) IP：服务部署所在机器的ip地址 PORT：服务部署后获取的端口 DEPLOY\_NAME：申请时填写的服务名称

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```

{
  "image": "<base64数据>"
}

```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
threshold	否	number	-	默认值为0.3，请在我的模型列表-模型效果查看推荐阈值

## 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	识别结果数组
+name	否	string	分类名称
+score	否	number	置信度
+location	否		
++left	否	number	检测到的目标主体区域到图片左边界的距离
++top	否	number	检测到的目标主体区域到图片上边界的距离
++width	否	number	检测到的目标主体区域的宽度
++height	否	number	检测到的目标主体区域的高度

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如缺少必要出入参时返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有任何疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
337000	Auth check failed	离线鉴权调用失败

### 模型更新/回滚操作说明

#### 模型更新

1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。

两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 nvidia-smi命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如easycl\_\$(DEPLOY\_NAME)\_v2

\$(DEPLOY\_NAME) :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_$(DEPLOY_NAME)_v2
cd easedl_$(DEPLOY_NAME)_v2
**将部署包上传至服务器该目录并解压**
tar zxvf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh

**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/$(DEPLOY_NAME) /home/baidu/work/$(DEPLOY_NAME)_V1
**记录当前模型的端口号**
docker ps -a |grep $(DEPLOY_NAME)

**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：$(DEPLOY_NAME)，前面已备份**
python2 install.py remove $(DEPLOY_NAME)
**安装当前部署包内新的EasyDL服务：$(DEPLOY_NAME)**
python2 install.py install $(DEPLOY_NAME)

** (可选操作) 更新证书**
python2 install.py lu

```

### 模型回滚

以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/$(DEPLOY_NAME) /home/baidu/work/$(DEPLOY_NAME)_V2
**使用V1版本**
cp -r /home/baidu/work/$(DEPLOY_NAME)_V1 /home/baidu/work/$(DEPLOY_NAME)

**停止当前模型容器**
docker ps -a |grep $(DEPLOY_NAME)
docker rm -f ${容器名}

**创建新的容器**
cd /home/baidu/work/$(DEPLOY_NAME) && bash start/start-1.sh

** (可选操作) 进入V1版本部署包所在目录执行license更新操作，假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easycl_$(DEPLOY_NAME)_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 端云协同服务说明

### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）

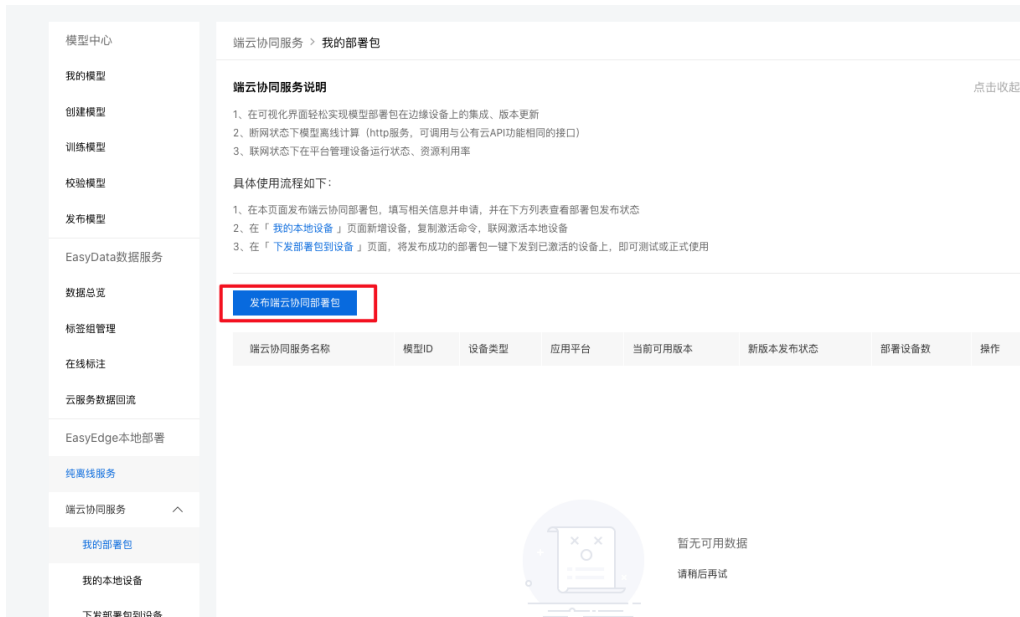
- 联网状态下在平台管理设备运行状态、资源利用率

目前本地服务器的应用平台支持Linux-AMD64(x86-64)，具体使用流程请参考下方文档。

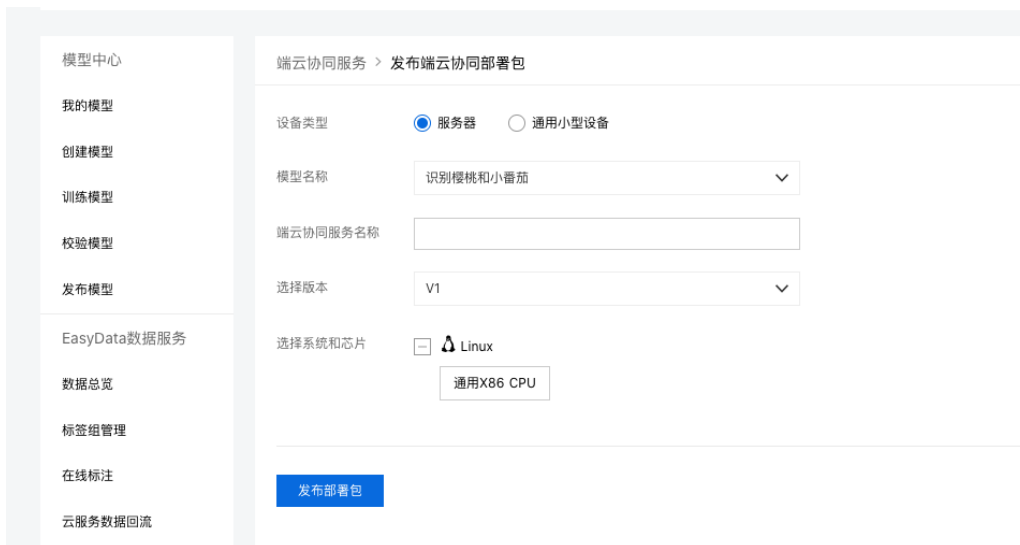
## 使用流程

### Step 1 发布端云协同部署包

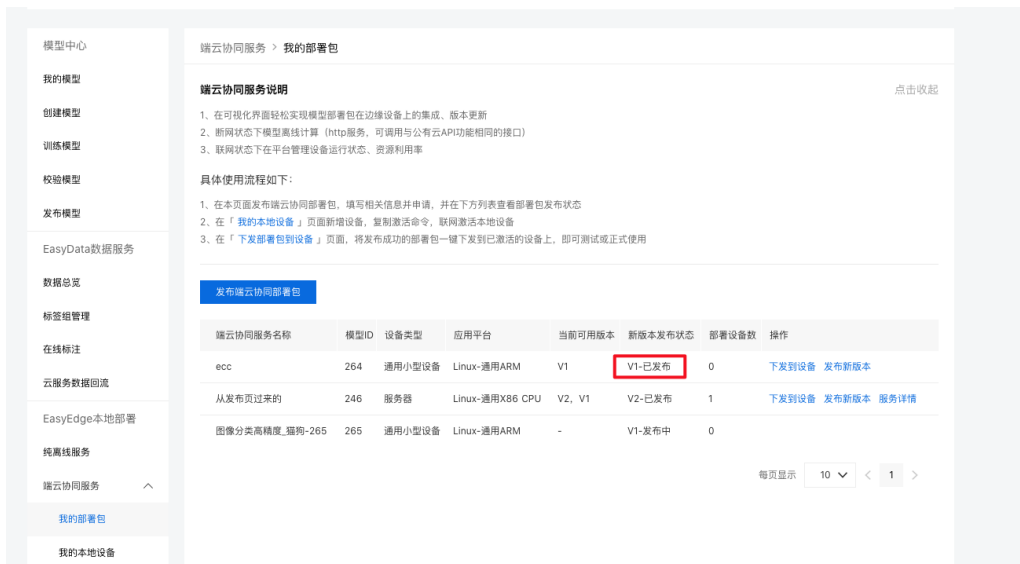
在[我的部署包](#)页面点击「发布端云协同部署包」



填写服务名称，选择模型版本并提交发布



在列表查看部署包发布状态

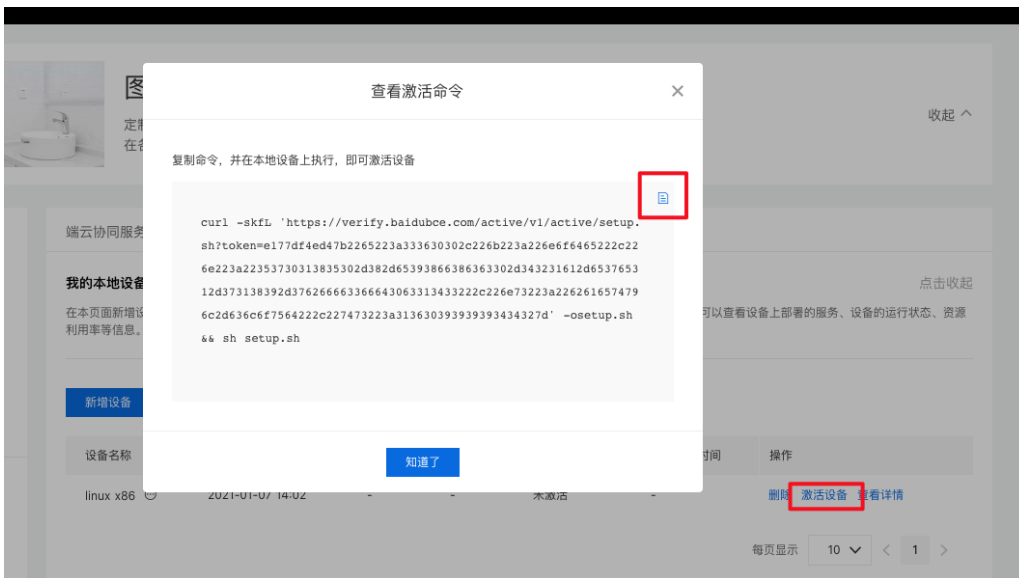


## Step 2 新增设备并激活

在[我的本地设备](#)页面新增设备



在列表中，点击设备对应的「[激活设备](#)」操作，复制激活命令并在本地设备上执行即可

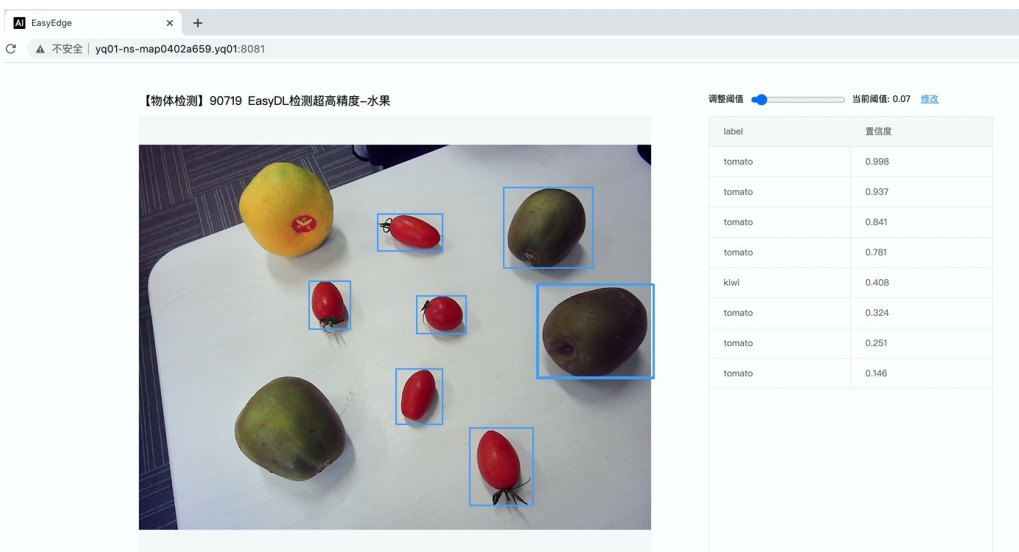


### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入<http://{设备ip}:{服务端口}>，默认8080，即可预览效果：



具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

[云端管理说明](#)



## 模型部署包管理

在**我的部署包**页面可以进行已发布的模型部署包的管理。

### 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。

端云协同服务 > 我的部署包

**端云协同服务说明** 点击收起

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「**我的本地设备**」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「**下发部署包到设备**」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

**发布端云协同部署包**

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
ecc	264	通用小型设备	Linux-通用ARM	V1	V1-已发布	0	<a href="#">下发到设备</a> <b>发布新版本</b>
...	246	服务器	Linux-通用X86 CPU	V2, V1	V2-已发布	1	<a href="#">下发到设备</a> <a href="#">发布新版本</a> <a href="#">服务详情</a>
...	265	通用小型设备	Linux-通用ARM	-	V1-发布中	0	

每页显示 10 < 1 >

**发布新版本** ×

将最新训练的模型版本发布为服务，发布成功后，即可从云端下发到设备

服务名称 ecc

模型ID 264

选择新版本 V1

**确认** **取消**

新版本发布成功后，即可在「下发部署包到设备」页面或当前服务的「服务详情」页面，将新版本下发到本地设备上。

端云协同服务 > 我的部署包

**端云协同服务说明** 点击收起

- 1、在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 2、断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 3、联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

- 1、在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
- 2、在「**我的本地设备**」页面新增设备，复制激活命令，联网激活本地设备
- 3、在「**下发部署包到设备**」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

**发布端云协同部署包**

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
ecc	264	通用小型设备	Linux-通用ARM	V1	V1-已发布	0	<a href="#">下发到设备</a> <a href="#">发布新版本</a>
...	246	服务器	Linux-通用X86 CPU	V2, V1	V2-已发布	1	<a href="#">下发到设备</a> <b>发布新版本</b> <a href="#">服务详情</a>
...	265	通用小型设备	Linux-通用ARM	-	V1-发布中	0	



### 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。

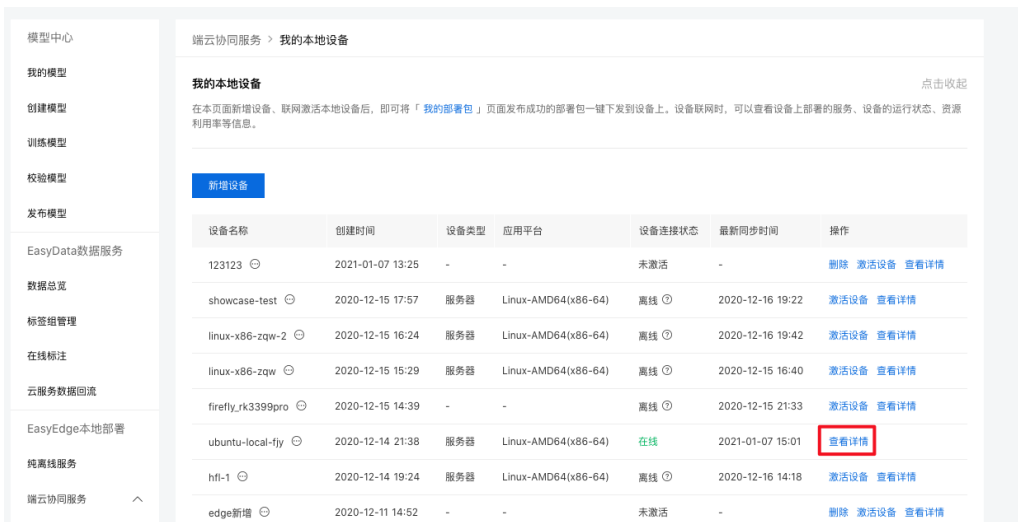


### 本地设备管理

在**我的本地设备**页面可以进行所有本地设备的管理。

### 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：



设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况

端云协同服务 > 我的本地设备 > ubuntu-local-fly

**设备详情**

设备名称 ubuntu-local-fly 连接状态 在线 实时刷新  OFF

设备类型 服务器 应用平台 Linux-AMD64(x86-64) 最新同步时间 2021-01-07 15:00

CPU使用率 31.1% 内存使用率 35.8%

**端云协同服务详情**

服务名称	模型ID	CPU占比	内存使用情况	内存占比	操作
██████████	246	0.01%	156.7MB	0.93%	<a href="#">查看服务配置</a>

## 通用小型设备部署

### 如何在通用小型设备部署

训练完毕后，可以选择将模型通过「SDK-纯离线服务」或「API-端云协同服务」部署，具体介绍如下：

#### 纯离线服务部署

纯离线服务目前仅支持通过SDK集成，可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布设备端SDK：

- 选择模型
- 选择部署方式「EasyEdge本地部署」-「通用小型设备」
- 选择版本
- 选择集成方式
- 点击发布

**发布模型**

选择模型 男女分类

部署方式 EasyEdge本地部署 通用小型设备

选择版本 V1

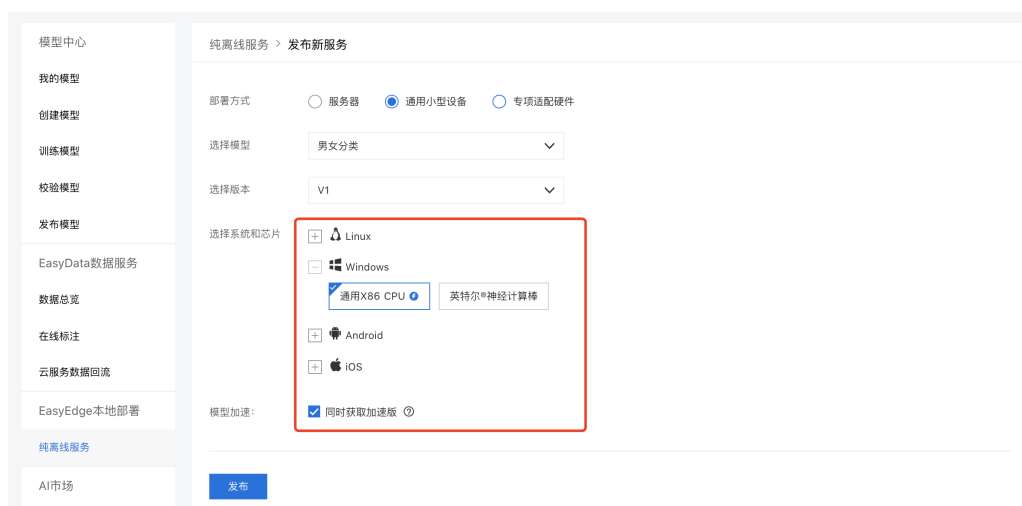
集成方式  SDK-纯离线服务

**发布**

**说明：**

1. 设备端SDK支持Android、iOS、Windows、Linux操作系统，具体的系统、硬件环境支持请参考[技术文档](#)，提供可直接体验的移动端app安装包，以及相应代码包、说明文档，供企业用户/开发者二次开发
2. 如SDK生成失败，或有任何其他问题，欢迎[提交工单](#)或加入QQ群(679517246)咨询了解

- 再根据实际使用设备选择系统与芯片
- 点击发布



也可以直接在「EasyEdge本地部署」-「纯离线服务」页面点击发布新服务，按上图所述进行申请发布

## 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

### 纯离线SDK说明

#### 纯离线SDK简介

本文档主要说明定制化模型发布后获得的SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 前往[官方论坛](#)交流，与其他开发者进行互动

#### SDK说明

SDK支持iOS、Android、Linux、Windows四种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
iOS	iOS 8.0 以上 (A仿生芯片版要求11.0以上)	ARMv7 ARM64 (Standard architectures) (暂不支持模拟器)
Android	通用ARM: Android 19以上 SNPE : Android 21以上 DDK : Android 21以上	通用ARM: 绝大部分的手机和平板、比较耗时 SNPE : 高通Soc, 仅支持Qualcomm Snapdragon 450 之后发布的soc。其中 660 之后的型号可能含有 Hexagon DSP模块, 具体列表见snpe 高通骁龙引擎 DDK : CPU支持华为麒麟970N、980的arm-v8a的soc, 支持的机型 mate10, mate10pro, P20, mate20等  支持armeabi-v7a arm-v8a CPU 架构, DDK仅支持 arm-v8a
Linux C++		CPU: AArch64 ARMv71 ASIC: Hisilicon NNIE1.1 on AArch64 (Hi3559AV100/Hi3559CV100等) ASIC: Hisilicon NNIE1.2 on ARMv71 (Hi3519AV100/Hi3559V200等)
Linux Python		Intel Movidius Myriad2/Myriad X
Linux Ubuntu 16.04		AArch64 HUAWEI Atlas 200
Windows	64位 Windows7 及以上	Intel CPU x86_64 Intel Movidius Myriad2/Myriad X (仅支持Win10)  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015

## 说明

针对iOS操作系统：虽然SDK库文件很大（ipa文件很大），但最终应用在用户设备中所占用的大小会缩小很多，如图像分类下载的ipa文件可能会100M以上，但实际集成在设备中只有20M左右。这与multi architectures、bitcode和AppStore的优化有关。

## 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)页面查看评测信息表

## 自适应芯片版SDK

发布SDK时可根据实际应用时的硬件/芯片配置选择最合适的SDK。如“华为NPU版”就是针对华为NPU芯片做了适配与加速的SDK。如实际应用时需要适配多种芯片，就可以发布“自适应芯片版”SDK，SDK被集成后会自动判断设备的芯片并运行相应的模型。

## 加速版SDK

发布SDK时，勾选「同时获取加速版」，就可以同时获得适配部分芯片（需选中且右侧带有加速标记）的基础版SDK和加速版SDK。



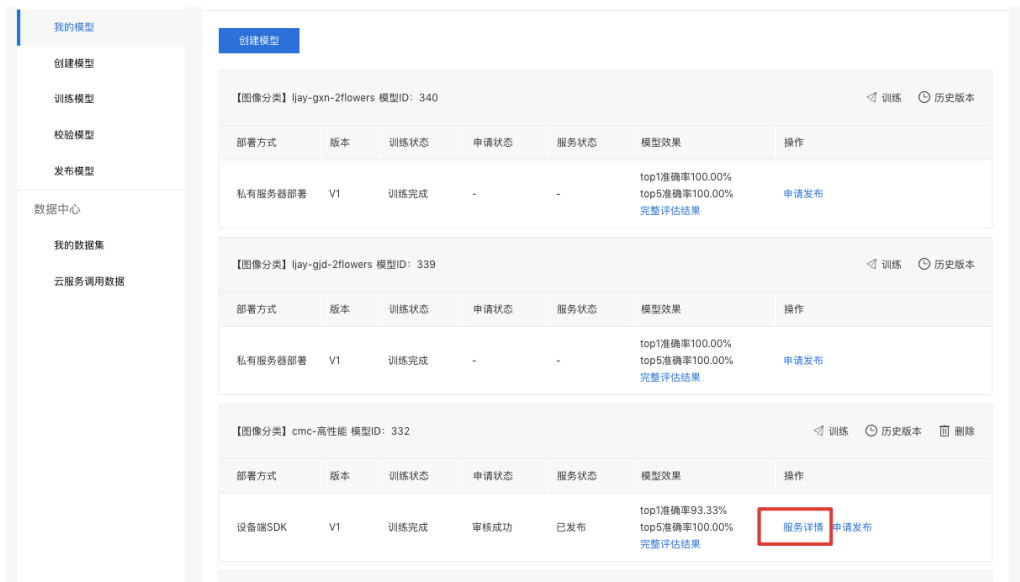
目前加速版SDK已支持Windows X86、Linux ARM、iOS ARM、Android ARM环境，加速后的SDK普遍在包大小、内存占用、识别速度等方面表现更优，详细对比请见[算法性能及适配硬件](#)。

加速版SDK和基础版的测试方式类似，只需在EasyDL控制台新增「加速版」测试序列号，即可获得3个月的测试期。

### 激活&使用SDK

SDK的激活与使用分以下四步：

#### ① 在【我的模型】-【服务详情】内下载SDK



### 设备端SDK下载

此处下载的SDK为未授权SDK，需要获取序列号激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发

操作系统	操作
iOS	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Android	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Linux	<a href="#">下载SDK</a> <a href="#">获取序列号</a>
Windows	<a href="#">下载SDK</a> <a href="#">获取序列号</a>

#### ② 在控制台获取序列号

按单台设备获得授权并使用SDK时：



Android或iOS操作系统的SDK可以选择按产品线激活（仅支持开发手机APP），序列号与包名（Package Name/Bundle ID）绑定：



### ③ 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

### ④ 正式使用

#### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在百度云控制台内[提交工单](#)反馈

#### 1、激活失败怎么办？

按设备激活时，激活失败可能由于以下几个原因造成：

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在百度云控制台内[提交工单](#)反馈

按产品线激活时，激活失败可能由于以下几个原因造成：

- ①可能是包名填写错误，请核对与序列号绑定的包名是否与实际包名一致
- ②序列号填写错误，请核实序列号后重新激活
- ③首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ④模型发布者和序列号所属账号非同一个账号，如果存在这种异常建议更换账号获取有效序列号
- ⑤序列号已过有效期，请申请延期后重试
- ⑥如有其他异常请在百度云控制台内[提交工单](#)反馈

## 2、怎样申请序列号使用延期

序列号激活后有效期为三个月，可以在[控制台](#)进行申请，申请流程：

### 1) 填写申请信息

The screenshot shows the 'EasyDL定制训练平台' (EasyDL Custom Training Platform) interface. The breadcrumb path is '产品服务 / EasyDL定制训练平台 - 离线SDK管理 / 申请延期'. The main content area is titled '申请延期' (Apply for Extension). It contains the following fields:

- \* 选择模型: 请选择 (dropdown menu)
- \* 选择版本: 请选择 (dropdown menu)
- \* 申请到期时间: 2019-03-31 (calendar icon)
- \* 申请延期理由: 请填写少于500字的申请延期理由 (text area)

A '提交审核' (Submit for Review) button is located at the bottom of the form.

- 2) 等待审核：审核周期通常需要1-3个工作日左右，期间会有工作人员电话回访，请填写有效的联系方式并保证手机畅通

## Android集成文档

### 简介

**1.1 Android SDK 硬件要求** Android 版本：支持 Android 5.0 (API 21) 及以上

硬件：支持 arm64-v8a 和 armeabi-v7a，暂不支持模拟器

通常您下载的SDK只支持固定的某一类芯片。

- **通用ARM**：支持大部分ARM 架构的手机、平板及开发板。**通常选择这个引擎进行推理。**
- **通用ARM GPU**：支持骁龙、麒麟、联发科等带GPU的手机、平板及开发板。
- **高端芯片AI加速模块**：
  - **高通骁龙引擎SNPE**：高通骁龙高端SOC，利用自带的DSP加速。其中 660 之后的型号可能含有 Hexagon DSP模块，具体列表见snpe高通骁龙引擎官网。
  - **华为NPU引擎DDK**：华为麒麟980的arm-v8a的soc。具体手机机型为mate10，mate10pro，P20，mate20，荣耀v20等。
  - **华为达芬奇NPU引擎DAVINCI**：华为NPU的后续版本，华为麒麟810，820，990，985的arm-v8a的soc。具体手机机型为华为mate30，p40，nova6，荣耀v30等。

**通用ARM**有额外的加速版，但是有一定的精度损失。

因GPU硬件限制，通用ARM GPU物体检测模型输入尺寸较大时会运行失败，可以在训练的时候将输入尺寸设为300\*300。



高端芯片AI加速模块，一般情况下推理速度较快。

运行内存不能过小，一般大于demo的assets目录大小的3倍。

### 1.2 功能支持 | 引擎 | 图像分类 | 物体检测 | 图像分割 | 文字识别

只支持EasyEdge | 姿态估计 | | :: | :: | :: | :: | :: | :: | | 通用ARM | √ | √ | √ | √ | √ | | 通用ARM GPU | √ | √ | √ | √ | | 高通骁龙引擎SNPE | √ | √  
| | 华为NPU引擎DDK | √ | √ | | | 华为达芬奇NPU引擎DAVINCI | √ | √ | √ | |

### 1.3 Release Notes

时间	版本	说明
2023.08.31	0.10.12	新增支持实例数鉴权；SNPE引擎升级；迭代优化
2023.06.29	0.10.11	迭代优化
2023.05.17	0.10.10	横屏兼容；迭代优化
2023.03.16	0.10.9	达芬奇NPU支持更多模型及语义分割模型；各芯片支持更多语义分割模型；精简版代码补充；迭代优化
2022.12.29	0.10.8	ARM / ARM-GPU 引擎升级；迭代优化
2022.10.27	0.10.7	达芬奇NPU新增适配麒麟985；迭代优化
2022.09.15	0.10.6	SNPE引擎升级；迭代优化
2022.07.28	0.10.5	迭代优化
2022.06.30	0.10.4	支持Android11；支持EasyEdge语义分割模型；迭代优化
2022.05.18	0.10.3	ARM / ARM-GPU 引擎升级；支持更多加速版模型发布；迭代优化
2022.03.25	0.10.2	ARM / ARM-GPU 引擎升级；支持更多检测模型；迭代优化
2021.12.22	0.10.1	DDK不再支持Kirin 970；迭代优化
2021.10.20	0.10.0	更新鉴权；更新达芬奇NPU、SNPE、通用ARM及ARM-GPU引擎；新增达芬奇NPU对检测模型的支持；支持更多姿态估计模型
2021.07.29	0.9.17	迭代优化
2021.06.29	0.9.16	迭代优化
2021.05.13	0.9.15	更新鉴权，更新通用arm及通用arm gpu引擎
2021.04.02	0.9.14	修正bug
2021.03.09	0.9.13	更新android arm的预处理加速
2020.12.18	0.9.12	通用ARM引擎升级；新增ARM GPU引擎
2020.10.29	0.9.10	迭代优化
2020.9.01	0.9.9	迭代优化
2020.8.11	0.9.8	更新ddk 达芬奇引擎
2020.7.14	0.9.7	支持arm版ocr模型，模型加载优化
2020.6.23	0.9.6	支持arm版fasterrcnn模型
2020.5.14	0.9.5	新增华为新的达芬奇架构npu的部分图像分类模型
2020.4.17	0.9.4	新增arm通用引擎量化模型支持
2020.1.17	0.9.3	新增arm通用引擎图像分割模型支持
2019.12.26	0.9.2	新增华为kirin麒麟芯片的物体检测支持
2019.12.04	0.9.1	使用paddleLite作为arm预测引擎
2019.08.30	0.9.0	支持EasyDL专业版
2019.08.30	0.8.2	支持华为麒麟980的物体检测模型
2019.08.29	0.8.1	修复相机在开发版调用奔溃的问题
2019.06.20	0.8.0	高通手机引擎优化
2019.05.24	0.7.0	升级引擎
2019.05.14	0.6.0	优化demo程序
2019.04.12	0.5.0	新增华为麒麟980支持
2019.03.29	0.4.0	引擎优化，支持sd卡模型读取
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

## 快速开始

### 2.1 安装软件及硬件准备

扫描模型下载SDK处的网页上的二维码，无需任何依赖，直接体验

如果需要源码方式测试：

打开AndroidStudio，点击 "Import Project..."。在一台较新的手机上测试。

详细步骤如下：

1. 准备一台较新的手机，如果不是通用arm版本，请参见本文的“硬件要求”，确认是否符合SDK的要求
2. 安装较新版本的AndroidStudio，[下载地址](#)
3. 新建一个HelloWorld项目，Android Studio会自动下载依赖，在这台较新的手机上测试通过这个helloworld项目。注意不支持模拟器。
4. 解压下载的SDK。
5. 打开AndroidStudio，点击 "Import Project..."。即：File->New-> "Import Project..."，选择解压后的目录。
6. 此时点击运行按钮（同第3步），手机上会有新app安装完毕，运行效果和二维码扫描的一样。
7. 手机上UI界面显示后，如果点击UI界面上的“开始使用”按钮，可能会报序列号错误。请参见下文修改

## 2.2 使用序列号激活

如果使用的是EasyEdge的开源模型，无需序列号，可以跳过本段直接测试。

建议申请包名为"com.baidu.ai.easyaimobile.demo"的序列号用于测试。

本文假设已经获取到序列号，并且这个序列号已经绑定包名。

SDK默认使用离线激活方式，即首次联网激活，后续离线使用。SDK同时支持按实例数鉴权方式，即周期性联网激活，离线后会释放所占设备实例。按实例数鉴权的启用参考本节2.2.3说明

**2.2.1 填写序列号** 打开Android Studio的项目，修改MainActivity类的开头SERIAL\_NUM字段。 MainActivity 位于 app\src\main\java\com\baidu\ai\edge\demo\MainActivity.java文件内。

```
// 请替换为您的序列号
private static final String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX"; //这里填您的序列号
```

### 2.2.2 修改包名

如果申请的包名为"com.baidu.ai.easyaimobile.demo"，这个是demo的包名，可以不用修改

打开app/build.gradle文件，修改"com.baidu.ai.easyaimobile.demo"为申请的包名

```
defaultConfig {
    applicationId "com.baidu.ai.easyaimobile.demo" // 修改为比如"com.xxx.xxx"
}
```

修改序列号和包名后，可以运行测试，效果同扫描二维码的一致

**2.2.3 按实例数鉴权** 设置好序列号和包名后，调用配置类的以下方法启用并配置心跳间隔时间：

```
XXXConfig config = new XXXConfig();
// 启用按实例数鉴权，配置心跳间隔，单位：秒
config.setInstanceAuthMode(10000);
```

配置类的详细说明参考后续章节【调用流程】

## 2.3 测试精简版

对于通用ARM、高通骁龙引擎SNPE、华为NPU引擎DDK和达芬奇NPU引擎Davinci的常见功能，项目内自带精简版，可以忽略开发板不兼容的摄像头。

此外，由于实时摄像开启，会导致接口的耗时变大，此时也可以使用精简版测试。

目前以下硬件环境有精简版测试：

- 通用ARM：图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation)，姿态估计 (Pose)
- 通用ARM GPU：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)，姿态估计 (Pose)
- 高通骁龙引擎SNPE：图像分类 (Classify)，物体检测 (Detection)
- 华为NPU引擎DDK：图像分类 (Classify)，物体检测 (Detection)
- 华为达芬奇NPU引擎Davinci：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)

具体代码分别在infertest、snpetest、ddktest和davincitest目录下。

修改方法为（以通用ARM为例）：更改app/main/AndroidManifest.xml中的启动Activity。

```
<activity android:name=".infertest.MainActivity" <!-- 原始的是".MainActivity" -->
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

开启后会自动选择图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation) 或姿态估计 (Pose) 测试。

Demo APP 检测模型运行示例

精简版检测模型运行示例



```

Hello World!
ARM Detection
Start running: 0
Predict 0: (size:100, firstRe
confidence:0.6314938, bo
181)}}
Finish running
Task finished
  
```

### 识别结果

置信度  0.30

序号	名称	置信度
1	person	0.63
2	person	0.47
3	car	0.42
4	horse	0.40
5	dog	0.34
6	truck	0.34

#### 使用说明

##### 3.1 代码目录结构

集成时需要“复制到自己的项目里”的目录或者文件：

1. app/libs

## 2. app/src/main/assets/xxxx-xxxx 如app/src/main/assets/infer

```

+app 简单的设置，模拟用户的项目
|---+libs 实际使用时需要复制到自己的项目里
    |---arm64-v8a v8a的so
    |---armeabi-v7a v7a的so
    |---easyedge-sdk.jar jar库文件
|---+src/main
    |---+assets
        |---demo demo项目的配置，实际集成不需要
        |---infer 也可能是其它命名，infer表示通用arm。实际使用时可以复制到自己的项目里
|---+java/com.baidu.ai.edge/demo
    |---+infertest 通用Arm精简版测试，里面有SDK的集成逻辑
        |--- MainActivity 通用Arm精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里
        面的序列号
            |--- TestInferArmClassifyTask 通用Arm精简版分类
            |--- TestInferArmDetectionTask 通用Arm精简版检测
            |--- TestInferArmOcrTask 通用Arm精简版OCR
            |--- TestInferArmPoseTask 通用Arm精简版姿态
            |--- TestInferArmSegmentTask 通用Arm精简版分割
            |--- TestInferArmGpuClassifyTask 通用ArmGpu精简版分类
            |--- TestInferArmGpuDetectionTask 通用ArmGpu精简版检测
            |--- TestInferArmGpuPoseTask 通用ArmGpu精简版姿态
            |--- TestInferArmGpuSegmentTask 通用ArmGpu精简版分割
    |---+snpetest SNPE精简版测试
        |--- MainActivity SNPE精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
        的序列号
            |--- TestSnpeDspClassifyTask SNPE DSP精简版分类
            |--- TestSnpeDspDetectionTask SNPE DSP精简版检测
            |--- TestSnpeGpuClassifyTask SNPE Gpu精简版分类
            |--- TestSnpeGpuDetectionTask SNPE Gpu精简版检测
    |---+ddktest DDK精简版测试
        |--- MainActivity DDK精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面的
        序列号
            |--- TestDDKClassifyTask DDK精简版分类
            |--- TestDDKDetectionTask DDK精简版检测
    |---+davincitest Davinci精简版测试
        |--- MainActivity Davinci精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
        的序列号
            |--- TestDavinciClassifyTask Davinci精简版分类
            |--- TestDavinciDetectionTask Davinci精简版检测
            |--- TestDavinciSegmentTask Davinci精简版分割
        |--- CameraActivity 摄像头扫描示例，里面有SDK的集成逻辑
        |--- MainActivity 启动Activity，使用时需要修改里面的序列号
|--- build.gradle 这里修改包名
+camera_ui UI模块，集成时可以忽略

```

## 3.2 调用流程 以通用ARM的检测模型功能为例，

代码可以参考TestInferDetectionTask

1. 准备配置类，如InferConfig，输入：通常为一个assets目录下的文件夹，如infer。
2. 初始化Manager，比如InferManager。输入：第1步的配置类和序列号
3. 推理图片，可以多次调用 3.1 准备图片，作为Bitmap输入 3.2 调用对应的推理方法，比如detect 3.3 解析结果，结果通常是一个List，调用结果类的Get方法，通常能获取想要的结果
4. 直到长时间不再使用我们的SDK，调用Manger的destroy方法释放资源。

## 3.3 具体接口说明 下文的示例部分以通用ARM的检测模型功能为例

即接口为InferConfig， InferManager， InferManager.detect。

其它引擎和模型调用方法类似。

下文假设已有序列号及对应的包名

### 3.3.1. 准备配置类

- INFER：通用ARM，`InferConfig`
- ARM GPU：ArmGpuConfig
- SNPE：高通骁龙DSP，`SnpeConfig`
- SNPE GPU：高通骁龙GPU，`SnpeGpuConfig`
- DDK：华为NPU，`DDKConfig`
- DDKDAVINCI：华为达芬奇NPU，`DDKDaVinciConfig`

```
InferConfig mInferConfig = new InferConfig(getAssets(),
    "infer");
// assets 目录下的infer，infer表示通用arm
```

输入：assets下的配置  
输出：具体的配置类

### 3.3.2. 初始化Manager类

- INFER：通用ARM，`InferManager`
- ARM GPU：通用ARM GPU，`InferManager`
- SNPE：高通骁龙DSP，`SnpeManager`
- SNPE GPU：高通骁龙GPU，`SnpeManager`
- DDK：华为NPU，`DDKManager`
- DDKDAVINCI：华为达芬奇NPU，`DavinciManager`

```
String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX";

// InferManager 为例:
new InferManager(this, config, SERIAL_NUM); // config为上一步的InferConfig
```

#### 注意要点

1. 同一个时刻只能有唯一有效的InferManager。旧的InferManager必须调用destory后，才能新建一个new InferManager()。
2. InferManager的任何方法，都不能在UI线程中调用。
3. new InferManager() 及InferManager成员方法由于线程同步数据可见性问题，都必须在**一个线程中执行**。如使用android自带的ThreadHandler类。

输入：1.配置类；2.序列号  
输出：Manager类

### 3.3.3. 推理图片

- 接口可以多次调用，但是必须在一个线程里，不能并发
- confidence, 置信度[0-1]，小于confidence的结果不返回。填confidence=0，返回所有结果
- confidence可以不填，默认用模型推荐的。

准备图片，作为Bitmap输入，

- 输入为Bitmap，其中Bitmap的options为默认。如果强制指定的话，必须使用`Bitmap.Config.ARGB_8888`

调用对应的推理方法及结果解析 见下文的各个模型方法

### 3.3.4 分类Classify

```
public interface ClassifyInterface {
    List<ClassificationResultModel> classify(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 ClassifyInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 ClassificationResultModel  
 异常：一般首次出现。可以打印出异常错误码。

ClassificationResultModel  
 - label：分类标签，定义在label\_list.txt中  
 - confidence：置信度，0-1  
 - labelIndex：标签对应的序号

### 3.3.5 检测Detect

对于EasyDL口罩检测模型请注意输入图片中人脸大小建议保持在88到9696像素，可根据场景远近程度缩放图片后传入

```
public interface DetectInterface {
    List<DetectionResultModel> detect(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 DetectInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 DetectionResultModel List  
 异常：一般首次出现。可以打印出异常错误码。

DetectionResultModel  
 - label：标签，定义在label\_list.txt中  
 - confidence：置信度  
 - bounds：Rect，左上角和右下角坐标

### 3.3.6 图像分割Segmentation

```
public interface SegmentInterface {
    List<SegmentationResultModel> segment(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 SegmentInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 SegmentationResultModel  
 异常：一般首次出现。可以打印出异常错误码。

SegmentationResultModel  
 - label：标签，定义在label\_list.txt中  
 - confidence：置信度  
 - labelIndex：标签对应的序号  
 - box: Rect对象表示的对象框  
 - mask：byte[]表示的原图大小的0，1掩码，绘制1的像素即可得到当前对象区域

mask 字段说明，如何绘制掩码也可参考demo工程

```
1 0 1
image 1 1 0  =>  mask(byte[]) 101 110 011
0 1 1
```

### 3.3.7 文字识别OCR

暂时只支持通用ARM引擎，不支持其它引擎，暂时只支持EasyEdge的开源OCR模型。

```
public interface OcrInterface {
    List<OcrResultModel> ocr(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 OcrInterface
```



输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 OcrResultModel List，每个OcrResultModel对应结果里的一个四边形。  
 异常：一般首次出现。可以打印出异常错误码。

OcrResultModel

- label：识别出的文字
- confidence：置信度
- List<Point>：4个点构成四边形

### 3.3.8 姿态估计Pose

暂时只支持通用ARM引擎，不支持其它引擎

```
public interface PoseInterface {
    List<PoseResultModel> pose(Bitmap bitmap) throws BaseException;
    // 如InferManger 继承 PoseInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 PoseResultModel List  
 异常：一般首次出现。可以打印出异常错误码。

PoseResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- Pair<Point, Point>：2个点构成一条线

### 3.3.9 释放

释放后这个对象不能再使用，如果需要使用可以重新new一个出来。

```
public void destory() throws BaseException
```

### 3.3.10 整体示例

以通用ARM的图像分类预测流程为例：

```
try {
    // step 1: 准备配置类
    InferConfig config = new InferConfig(context.getAssets(), "infer");

    // step 2: 准备预测 Manager
    InferManager manager = new InferManager(context, config, "");

    // step 3: 准备待预测的图像，必须为 Bitmap.Config.ARGB_8888 格式，一般为默认格式
    Bitmap image = getFromSomeWhere();

    // step 4: 预测图像
    List<ClassificationResultModel> results = manager.classify(image, 0.3f);

    // step 5: 解析结果
    for (ClassificationResultModel resultModel : results) {
        Log.i(TAG, "labelIndex=" + resultModel.getLabelIndex()
            + ", labelName=" + resultModel.getLabel()
            + ", confidence=" + resultModel.getConfidence());
    }

    // step 6: 释放资源。预测完毕请及时释放资源
    manager.destory();
} catch (Exception e) {
    Log.e(TAG, e.getMessage());
}
```

### 3.3.11 高通骁龙引擎的额外配置

```
"autocheck_qcom": true, // 如果改成false, sdk跳过检查手机是否是高通的Soc, 非高通的Soc会奔溃直接导致app闪退
```

```
"snpe_runtimes_order": [],
// 不填写为自动, 按照 {DSP, GPU, GPU_FLOAT16, CPU}次序尝试初始化, 也可以手动指定如[2,1,3,0], 具体数字的定义见下段
```

```
public interface SnpeRuntimeInterface {
    int CPU = 0;
    int GPU = 1;
    int DSP = 2;
    int GPU_FLOAT16 = 3;
}
```

```
// SnpeManager 中, 使用public static ArrayList<Integer> getAvailableRuntimes(Context context) 方法可以获取高通SOC支持的运行方式
```

## 集成指南

1. 复制库文件libs
2. 添加Manifest权限
3. 复制模型文件
4. 添加调用代码(见上一步具体接口说明)

### 4.1 复制库文件libs A. 如果项目里没有自己的jar文件和so文件:

复制app/libs 至自己项目的app/libs目录。  
参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

### B. 如果项目里有自己的jar文件, 但没有so文件

easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a放到app/src/main/jniLibs目录下

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
}
```

### C. 如果项目里有自己的jar文件和so文件

easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a取交集和自己的so放在一起，交集的意思是比如自己的项目里有x86目录，必须删除x86。

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a' // abiFilter取交集，即只能少不能多
        }
    }
}
```

jar文件库如果没有设置成功的，编译的时候可以发现报错。

so库如果没有编译进去的话，也可以通过解压apk文件确认。运行的时候会有类似jni方法找不到的报错。

#### 4.2 Manifest配置

参考app/src/main/AndroidManifest.xml文件，添加：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- Android 11 支持 -->
<uses-permission
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />

<!-- 高版本 Android 支持 -->
<application
    android:requestLegacyExternalStorage="true"
    android:usesCleartextTraffic="true">
</application>
```

#### 4.3 混淆规则（可选） 请不要混淆SDK里的jar文件。

```
-keep class com.baidu.ai.edge.core.*.*{ *; }
```

#### 4.4 Android 11支持 除Manifest中必要配置外，请参考BaseActivity获取所有文件访问权限，否则可能影响SDK正常使用。

SDK 默认使用 easyedge-sdk.jar，未启用 AndroidX，若您的项目使用 AndroidX，并在集成中提示 android.support 相关错误，请参考 app/build.gradle 使用 etc/easyedge-sdk-androidx.jar 以支持 AndroidX：

```
// app/build.gradle

dependencies {
    implementation project(':camera_ui')
    implementation files('libs/easyedge-sdk-androidx.jar') // 修改 jar 包依赖
}
```

**错误码** | 错误码 | 错误描述 | 详细描述及解决方法 | |---|---|---| | 1001 | assets 目录下用户指定的配置文件不存在或不正确 | SDK使用assets目录下一系列文件作为配置文件。如果文件缺失或内容不正确，则有此报错 | | 1002 | json格式的配置文件解析出错 | 如缺少某些字段。正常情况下，配置文件请不要修改 | | 1003 | 应用缺少权限 | 请根据提示动态申请缺少的权限 | | 19xx | Sdk内部错误 | 请与百度人员联系 | | 2001 | XxxMANAGER 只允许一个实例 | 如已有XxxMANAGER对象，请调用destory方法 | | 2002 | XxxMANAGER 已经调用过destory方法 | 在一个已经调用destory方法的DETECT\_MANAGER对象上，不允许再调用任何方法 | | 2003 | 传入的assets下模型文件路径非法 | 比如缺少模型文件，XxxConfig.getModelFileAssetPath() 返回为null | | 2012 | JNI内存错误 | heap的内存不够 | | 2103 | license过期 | license失效或者系统时间有异常 | | 2601/2602 | assets 目录下模型文件打开/读取失败 | 请根据报错信息检查模型文件是否存在 | | 27xx | Sdk内部错误 | 请与百度人员联系 | | 28xx | 引擎内部错误 | 请与百度人员联系 | | 29xx | Sdk内部错误 | 请与百度人员联系 | | 3000 | so加载错误 | 请确认所有so文件存在于apk中 | | 3001 | 模型加载错误 | 请确认模型放置于能被加载到的合法路径中，并确保配置文件正确 | | 3002 | 模型卸载错误 | 请与百度人员联系 | | 3003 | 调用模型错误 | 在模型未加载正确或者so库未加载正确的情况下调用了分类接口 | | 31xx | SDK激活失败 | 请与百度人员联系 | | 4011 | SDK类型与设备硬件不匹配 | 比如适配DSP的SDK运行在麒麟芯片上会出现此报错，请在部署包支持的硬件上使用SDK | | 50xx | SDK调用异常 | 请与百度

人员联系 |

**报错日志收集** 通常 Logcat 可以看见日志及崩溃信息，若设备无法获取日志信息，可使用 Demo 中的 xCrash 工具：

```
// 1. 引入 app/build.gradle 的 xCrash 依赖
android {
    ...
    dependencies {
        implementation 'com.iqiyi.xcrash:xcrash-android-lib:2.4.5' // 可以保存崩溃信息，默认未引入
        ...
    }
}
// 2. 启用日志收集。日志将保存在 /sdcard/<包名>/xCrash
// app/src/main/java/com.baidu.ai.edge/demo/MyApplication.java
protected void attachBaseContext(Context context) {
    // 日志保存位置
    String basePath = Environment.getExternalStorageDirectory().toString() + "/" + context.getPackageName();
    // 启用
    XCrash.InitParameters params = new XCrash.InitParameters();
    params.setAppVersion(BaseManager.VERSION);
    params.setLogDir(basePath + "/xCrash");
    XCrash.init(this, params);
}
```

## 🔗 iOS集成文档

### 简介

本文档描述 EasyEdge/EasyDL iOS 离线预测SDK相关功能；

目前支持EasyEdge的功能包括：

- 图像分类
- 物体检测
- 人脸检测
- 姿态估计
- 百度OCR模型

目前支持EasyDL的功能包括：

- 图像分类
- 物体检测
- 图像分割

### 系统支持

系统：

- 通用arm版本：iOS 9.0 以上
- A仿生芯片版：iOS 15.0 及以上

硬件：arm64 (Standard architectures)（暂不支持模拟器）

内存：图像分割模型需要手机内存3GB以上，并尽量减少其他程序内存占用

### 离线SDK包说明

根据用户的选择，下载的离线SDK，可能包括以下类型：

- EasyEdge
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。

- EasyDL 通用版/全功能AI开发平台BML（原EasyDL专业版）
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。
  - 自适应芯片版：同时整合了以上两种版本，自动在iOS 15以下中使用通用ARM版，在iOS 15以上系统中使用A仿生芯片版，自适应系统，但SDK体积相对较大。
- AI市场试用版SDK

### SDK大小说明

SDK库的二进制与\_TEXT增量约3M。

资源文件大小根据模型不同可能有所差异。

物体检测(高性能)的DemoApp在iPhone 6, iOS 11.4下占用空间实测小于40M。

虽然SDK库文件很大（体现为SDK包文件很大，ipa文件很大），但最终应用在用户设备中所占用的大小会缩小很多。这与multi architectures、bitcode和AppStore的优化有关。

**获取序列号** 生成SDK后，点击获取序列号进入控制台获取。EasyEdge[控制台](#)、EasyDL[控制台](#)、BML[控制台](#)。

试用版SDK在SDK的RES文件夹中的SN.txt中包含试用序列号。

更换序列号、更换设备时，首次使用需要联网激活。激活成功之后，有效期内可离线使用。

### Release Notes

时间	版本	说明
2023.08.31	0.7.13	新增按实例数鉴权；迭代优化
2023.06.29	0.7.12	迭代优化
2023.05.17	0.7.11	CoreML引擎升级，支持更多语义分割模型；兼容横屏；迭代优化
2023.03.16	0.7.10	支持更多语义分割模型；迭代优化
2022.12.29	0.7.9	ARM引擎升级；迭代优化
2022.10.27	0.7.8	支持更多检测模型；迭代优化
2022.09.15	0.7.7	支持更多检测模型；迭代优化
2022.07.28	0.7.6	迭代优化
2022.06.29	0.7.5	支持EasyEdge语义分割模型；CoreML引擎升级，新增EasyEdge检测模型支持；迭代优化
2022.05.18	0.7.4	ARM引擎升级；支持EasyDL物体检测超高精度模型；支持更多加速版模型发布；迭代优化
2022.03.25	0.7.3	ARM引擎升级；支持更多检测模型
2021.12.22	0.7.2	支持EasyEdge更多姿态估计模型；迭代优化
2021.10.20	0.7.1	ARM引擎升级
2021.07.29	0.7.0	迭代优化
2021.04.06	0.6.1	ARM引擎升级
2021.03.09	0.6.0	支持EasyEdge人脸检测及姿态估计模型
2020.12.18	0.5.7	ARM引擎升级
2020.09.17	0.5.6	CoreML引擎升级，支持AI市场试用版SDK
2020.08.11	0.5.5	CoreML支持EasyDL专业版模型，支持EasyEdge OCR模型
2020.06.23	0.5.4	ARM引擎升级
2020.04.16	0.5.3	ARM引擎升级；支持压缩加速版模型
2020.03.13	0.5.2	ARM引擎升级；支持图像分割模型
2020.01.16	0.5.1	ARM引擎升级；增加推荐阈值支持
2019.12.04	0.5.0	ARM引擎升级；增加coreml3的支持
2019.10.24	0.4.5	支持EasyDL专业版；ARM引擎升级
2019.08.30	0.4.4	支持EasyDL经典版图像分类高性能、高精度
2019.06.20	0.4.3	引擎优化
2019.04.12	0.4.1	支持EasyDL经典版物体检测高精度、高性能模型
2019.03.29	0.4.0	引擎优化，支持CoreML；
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

### 快速开始 文件结构说明

```

.EasyEdge-iOS-SDK
├── EasyDLDemo # Demo工程文件
├── LIB # 依赖库
├── RES
│   ├── easyedge # 模型资源文件夹
│   │   ├── model
│   │   ├── params
│   │   ├── label_list.txt
│   │   ├── infer_cfg.json
│   │   └── conf.json
└── DOC # 文档

```

### 测试Demo

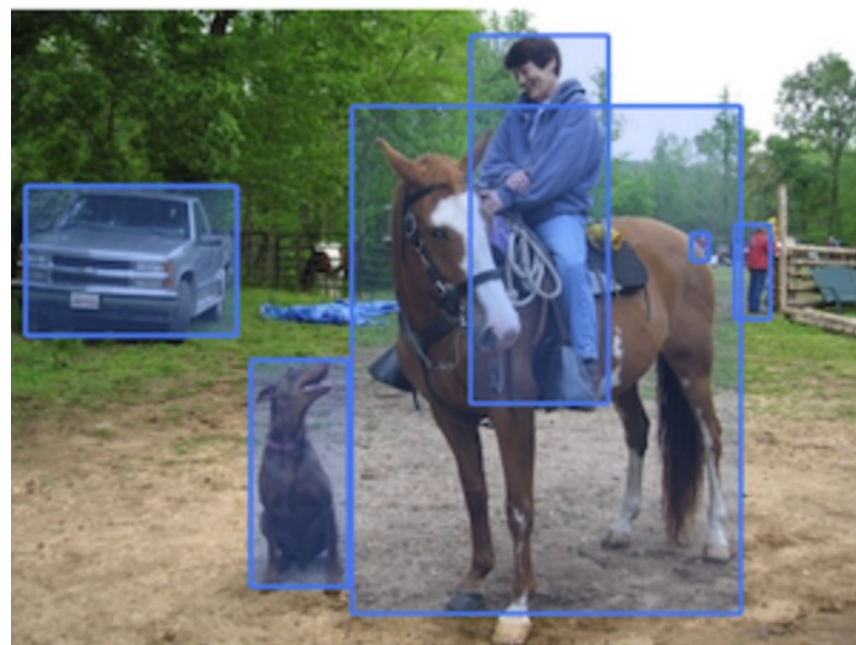
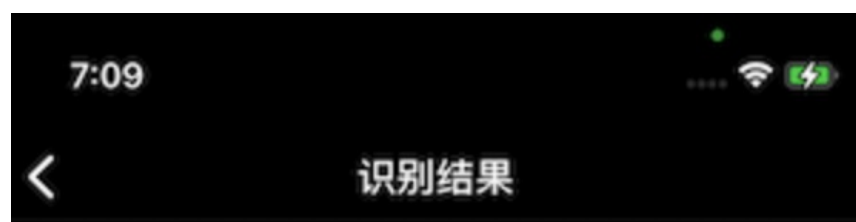
按如下步骤可直接运行 SDK 体验 Demo：

步骤一：用 Xcode 打开 EasyDLDemo/EasyDLDemo.xcodeproj

步骤二：配置开发者自己的签名

步骤三：连接手机运行，不支持模拟器

检测模型运行示例：



阈值：0.30



序号	名称	置信度
1	person	0.632
2	person	0.468
3	car	0.423
4	horse	0.400
5	dog	0.342

重新识别

SDK使用说明 集成指南 步骤一：依赖库集成 步骤二：import <EasyDL/EasyDL.h> , import <Vision/Vision.h>

### 依赖库集成

1. 复制 LIB 目录至项目合适的位置
2. 配置 Build Settings 中 Search paths: 以 SDK 中 LIB 目录路径为例

- Framework Search Paths : \${PROJECT\_DIR}/../LIB/lib
- Header Search Paths : \${PROJECT\_DIR}/../LIB/include
- Library Search Paths : \${PROJECT\_DIR}/../LIB/lib

集成过程如出现错误，请参考 Demo 工程对依赖库的引用

### 使用流程

1. 生成模型，下载SDK 开发者在官网下载的SDK已经自动为开发者配置了模型文件和相关配置，开发者直接运行即可。
2. 使用序列号激活 2.1. 离线激活（默认鉴权方式） 首次联网激活，后续离线使用

将前面申请的序列号填入：

```
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
```

根据序列号类型，序列号与BundleID绑定或与BundleID+设备绑定。  
请确保设备时间正确。

- 2.2. 按实例数激活 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间

填入序列号，配置按实例数鉴权并设置心跳间隔：

```
// 设置序列号
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
// 配置实例数鉴权及心跳间隔，单位：秒
[EasyDL setInstanceAuthMode:10000];
```

### 3. 初始化模型

```
EasyDLModel *_model = [[EasyDLModel alloc] initWithResourceDirectory:@"easymodel" withError:&err];
```

请注意相关资源必须以 folder reference 方式加入Xcode工程。也即默认的easymodel文件夹在Xcode文件列表里显示为蓝色。

### 4. 调用检测接口

```
UIImage *img = .....;
NSArray *result = [model detectUIImage:img withFilterScore:0 andError:&err];

/**
 * 检测图像
 * @param image 带检测图像
 * @param score 只返回得分高于score的结果(0 ~ 1)
 * @return 成功返回识别结果，NSArray的元素为对应模型的结果类型；失败返回nil，并在err中说明错误原因
 */
- (NSArray *)detectUIImage:(UIImage *)image
  withFilterScore:(CGFloat)score
    andError:(NSError **)err;
```

返回的数组类型如下，具体可参考 EasyDLResultData.h 中的定义：



模型类型	类型
图像-图像分类	EasyDLClassfiData
图像-物体检测/人脸检测	EasyDLObjectDetectionData
图像-实例分割/语义分割	EasyDLObjSegmentationData
图像-姿态估计	EasyDLPoseData
图像-文字识别	EasyDLOcrData

### 错误说明

SDK的方法会返回NSError错，直接返回的NSError的错误码定义在EEasyDLErrorCode中。NSError附带message（有时候会附带NSUnderlyingError），开发者可根据code和message进行错误判断和处理。

### FAQ

#### 1. 如何多线程并发预测？

SDK内部已经能充分利用多核的计算能力。不建议使用并发来预测。

如果开发者想并发使用，请务必注意EasyDLModel所有的方法都不是线程安全的。请初始化多个实例进行并发使用，如

```

- (void)testMultiThread {
    UIImage *img = [UIImage imageNamed:@"1.jpeg"];
    NSError *err;
    EasyDLModel * model1 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];
    EasyDLModel * model2 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];

    dispatch_queue_t queue1 = dispatch_queue_create("testQueue", DISPATCH_QUEUE_CONCURRENT);
    dispatch_queue_t queue2 = dispatch_queue_create("testQueue2", DISPATCH_QUEUE_CONCURRENT);

    dispatch_async(queue1, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model1 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"1: %@", res[0];
        }
    });

    dispatch_async(queue2, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model2 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"2: %@", res[0];
        }
    });
}

```

#### 2. 编译时出现 Undefined symbols for architecture arm64: ...

- 出现 `cx11, vtable` 字样：请引入 `libc++.tbd`
- 出现 `cv::Mat` 字样：请引入 `opencv2.framework`
- 出现 `CoreML, VNRequest` 字样：请引入 `CoreML.framework` 并务必 `#import <CoreML/CoreML.h>`

#### 3. 运行时报错 Image not found: xxx ...

请Embed具体报错的库。4.编译时报错：Invalid bitcode version 这个可能是开发者使用的xcodes低于12导致，可以升级至12版本。

### Windows集成文档

#### 简介

本文档介绍物体检测通用小型设备Windows SDK的使用方法。

- 硬件支持：
  - Intel CPU 普通版 \* x86\_64

- CPU 加速版 - Intel Xeon with AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE \* - AMD Core Processors with AVX2
- Intel Movidius Myriad2/Myriad X (仅支持Win10)
- 操作系统支持
  - 普通版：64位 Windows 7 及以上，64位Windows Server2012及以上
  - 加速版：64位 Windows 10，64位Windows Server 2019及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015-2019
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

\*intel 官方合作，拥有更好的适配与性能表现

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | 优化模型算法 | | 2022-09-15 | 1.7.0 | 新增支持表格预测 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | CPU基础版推理引擎优化升级；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | CPU加速版推理引擎优化升级 | | 2021-08-19 | 1.3.2 | 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版 | |

## 快速开始

### 1. 安装依赖

必须安装：

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

Visual C++ Redistributable Packages for Visual Studio 2015-2019

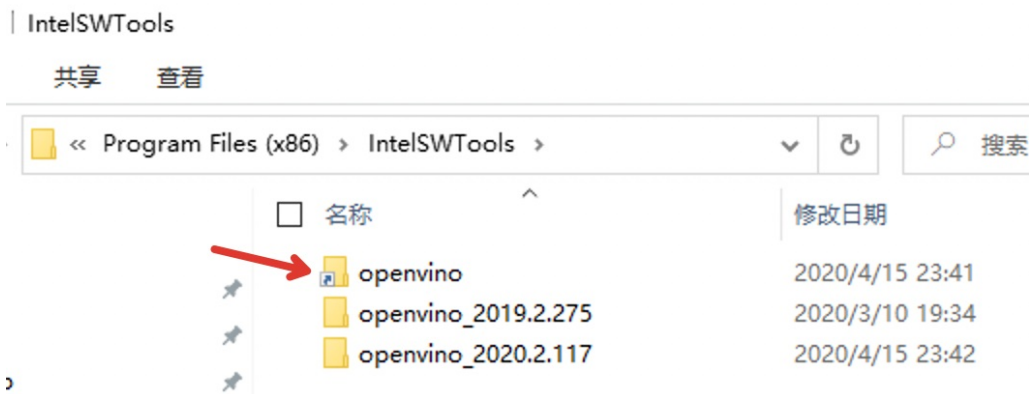
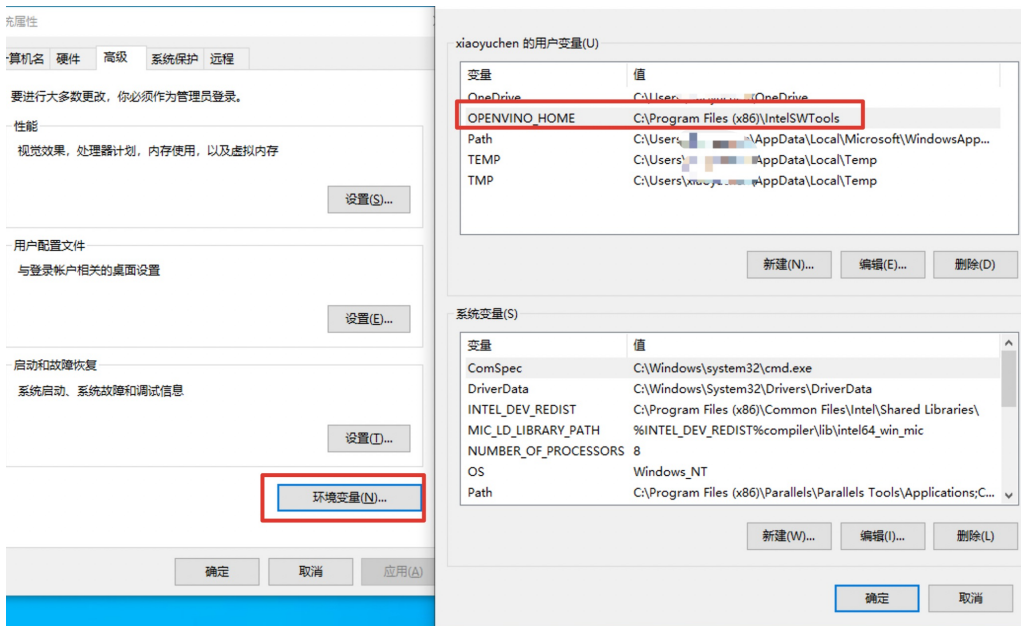
<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-160>

可选安装：

**Openvino (仅使用Python Intel Movidius必须)**

- 使用 OpenVINO™ toolkit 安装, 请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装, 请参考 [Openvino Inference Engine文档](#) 编译安装 2020.3.1LTS (必须) 版本。

安装完成后, 请设置环境变量OPENVINO\_HOME为您设置的安装地址, 默认是C:\Program Files (x86)\IntelSWTools, 并确保文件夹下的openvino的快捷方式指到了2020.3.1LTS版本。

**注意事项**

1. 安装目录不能包含中文
2. Windows Server 请自行开启, 选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“, 点击安装, 安装之后重启即可。

**2. 运行离线SDK**

解压下载好的SDK, 打开EasyEdge.exe, 输入Serial Num, 选择鉴权模式, 点击“启动服务“, 等待数秒即可启动成功, 本地服务默认运行在

`http://127.0.0.1:24401/`

其他任何语言只需通过HTTP调用即可。

如启动失败, 可参考如下步骤排查:



## 2.1 离线鉴权（默认鉴权模式） 首次联网激活，后续离线使用



## 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

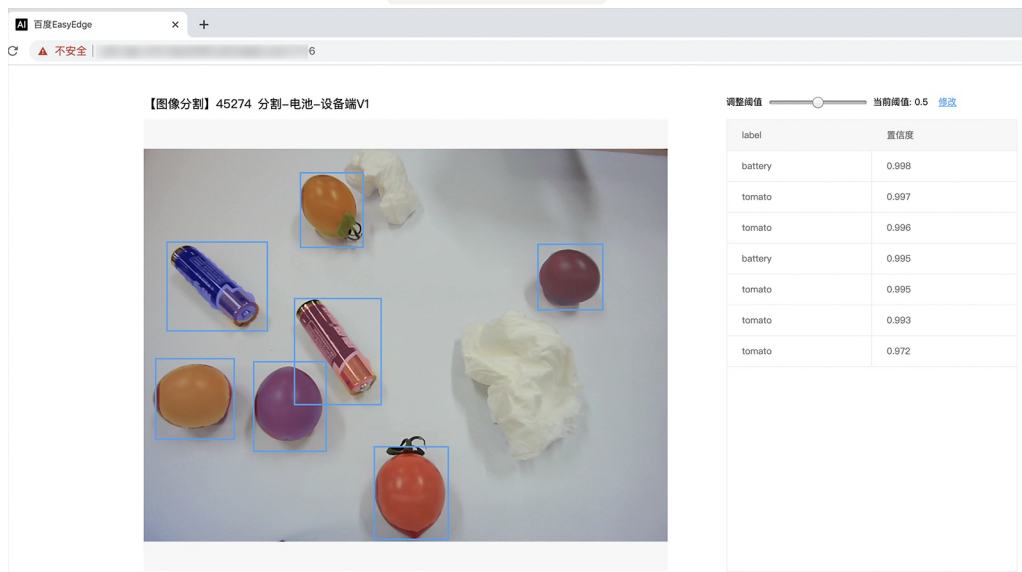
```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

## 2.3 序列号激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。



#### 使用说明

调用说明 使用示例如下：

```
python

c#

C++

java
```

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img).json()

print(result)
```

结果 获取的结果存储在response字符串中。请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|----|-----| | confidence | float | 0~1 | 检测的置信度 | | label | string | | 检测的类别 | | index | number | | 检测的类别 | | x1, y1 | float | 0~1 | 矩形的左上角坐标 (相对长宽的比例值) | | x2, y2 | float | 0~1 | 矩形的右下角坐标 (相对长宽的比例值) |

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

#### 集成指南

##### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

##### 基于c++ dll集成

##### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

##### 集成方法

参考src目录中的CMakeLists.txt进行集成

##### 基于c# dll集成

##### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

##### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

#### FAQ

1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：`.NET Framework 4.5` Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是Python Intel Movidius版，需额外确保Opencv安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `C:\Users\${用户名}\.baidu\easyedge` 目录，再重新激活。

8. 勾选“开机自动启动”后，程序闪退

一般是写注册表失败。

可以确认下HKEY\_CURRENT\_USER下Software\Microsoft\Windows\CurrentVersion\Run能否写入（如果不能写入，可能被杀毒软件等工具管制）。也可以尝试基于bin目录下的easyedge\_serving.exe命令行形式的二进制，自行配置开机自启动。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## Linux集成文档-C++

### 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：- 图像分类 - 物体检测 - 图像分割
- 硬件支持：
  - CPU: aarch64 armv7hf
  - GPU: ARM Mali G系列
  - ASIC: Hisilicon NNIE1.1 on aarch64 (Hi3559AV100/Hi3559CV100等)
  - ASIC: Hisilicon NNIE1.2 on armv7l (Hi3519AV100/Hi3559V200等)
  - Intel Movidius Myriad2 / Myriad X on x86\_64
  - Intel Movidius Myriad2 / Myriad X on armv7l
  - Intel Movidius Myriad2 / Myriad X on aarch64
  - Intel iGPU on x86\_64
  - 比特大陆 Bitmain SE5 (BM1684)



- 瑞芯微 RK3399Pro / RV1109 / RV1126 / RK3568 / RK3588
- 华为 Atlas200
- 晶晨 A311D
- 寒武纪 MLU220 on aarch64
- 英特尔 iGPU
- 操作系统支持：
  - Linux (Ubuntu, Centos, Debian等)
  - 海思HiLinux
  - 树莓派Raspbian/Debian
  - 瑞芯微Firefly

#### 性能数据参考 [算法性能及适配硬件](#)

**Release Notes** | 时间 | 版本 | 说明 | | --- | --- | --- | | 2023.08.31 | 1.8.3 | Atlas系列Socs支持语义分割模型，Atlas Cann版本升级至6.0.1 | | 2023.06.29 | 1.8.2 | 比特大陆版本升级至V23.03.01 | | 2023.05.17 | 1.8.1 | 新增支持intel iGPU + CPU异构模式 | | 2023.03.16 | 1.8.0 | 新增支持瑞芯微RK3588 | | 2022.10.27 | 1.7.1 | 新增语义分割模型http请求示例 | | 2022.09.15 | 1.7.0 | 新增瑞芯微 RK3568 支持, RK3399Pro、RV1126升级到RKNN1.7.1 | | 2022.07.28 | 1.6.0 | 引擎升级；新增英特尔 iGPU 支持 | | 2022.04.25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022.03.25 | 1.4.0 | EasyDL新增上线支持晶晨A311D NPU预测引擎；Arm CPU、Arm GPU引擎升级；atlas 200在EasyDL模型增加多个量化加速版本； | | 2021.12.22 | 1.3.5 | RK3399Pro, RV1109/RV1126 SDK扩展模型压缩加速能力，更新端上推理库版本;边缘控制台IEC功能升级，适配更多通用小型设备，NNIE在EasyDL增加量化加速版本；Atlas200升级到Cann5.0.3 | | 2021.06.29 | 1.3.1 | 视频流解析支持调整分辨率；预测引擎升级；设备端sdk新增支持瑞芯微RV1109、RV1126 | | 2021.05.13 | 1.3.0 | 新增视频流接入支持；EasyDL模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.0 | http server服务支持图片通过base64格式调用 | | 2021.01.27 | 1.1.0 | EasyDL经典版分类高性能模型升级；部分SDK不再需要单独安装OpenCV；新增RKNPU预测引擎支持；新增高通骁龙GPU预测引擎支持 | | 2020.12.18 | 1.0.0 | 1.0版本发布！安全加固升级、性能优化、引擎升级、接口优化等多项更新 | | 2020.10.29 | 0.5.7 | 优化多线程预测细节 | | 2020.09.17 | 0.5.6 | 支持linux aarch64架构的硬件接入intel神经计算棒预测；支持比特大陆计算盒SE50 BM1684 | | 2020.08.11 | 0.5.5 | 支持linux armv7hf架构硬件(如树莓派)接入intel神经计算棒预测 | | 2020.06.23 | 0.5.4 | arm引擎升级 | | 2020.05.15 | 0.5.3 | 支持EasyDL 专业版新增模型；支持树莓派(armv7hf, aarch64) | | 2020.04.16 | 0.5.2 | Jetson系列SDK支持多线程infer | | 2020.02.23 | 0.5.0 | 新增支持人脸口罩模型；Jetson SDK支持批量图片推理; ARM支持图像分割 | | 2020.01.16 | 0.4.7 | 上线海思NNIE1.2，支持EasyEdge以及EasyDL；ARM引擎升级；增加推荐阈值支持 | | 2019.12.26 | 0.4.6 | 海思NNIE支持EasyDL专业版 | | 2019.11.02 | 0.4.5 | 移除curl依赖；支持自动编译OpenCV；支持EasyDL 专业版 Yolov3；支持EasyDL经典版高精度物体检测模型升级 | | 2019.10.25 | 0.4.4 | ARM引擎升级，性能提升30%；支持EasyDL专业版模型 | | 2019.09.23 | 0.4.3 | 增加海思NNIE加速芯片支持 | | 2019.08.30 | 0.4.2 | ARM引擎升级；支持分类高性能与高精度模型 | | 2019.07.25 | 0.4.1 | 引擎升级，性能提升 | | 2019.06.11 | 0.3.3 | paddle引擎升级；性能提升 | | 2019.05.16 | 0.3.2 | 新增armv7l支持 | | 2019.04.25 | 0.3.1 | 优化硬件支持 | | 2019.03.29 | 0.3.0 | ARM64 支持；效果提升 | | 2019.02.20 | 0.2.1 | paddle引擎支持；效果提升 | | 2018.11.30 | 0.1.0 | 第一版！ |

**【1.0 接口升级】** 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例。**【关于SDK包与RES模型文件夹配套使用的说明】** 我们强烈建议用户使用部署tar包中配套的SDK和RES一起使用。更新模型时，如果SDK版本号有更新，请务必同时更新SDK，旧版本的SDK可能无法正确适配新发布出来的RES。

#### 快速开始

SDK在以下环境中测试通过

- aarch64(arm64), Ubuntu 16.04, gcc 5.3 (RK3399)
- Hi3559AV100, aarch64, Ubuntu 16.04, gcc 5.3
- Hi3519AV100, armv7l, HiLinux 4.9.37, (Hi3519AV100R001C02SPC020)
- armv7hf, Raspbian, (Raspberry 3b)
- aarch64, Raspbian, (Raspberry 4b)
- armv7hf, Raspbian, (Raspberry 3b+)



- armv7hf, Ubuntu 16.04, (RK3288)
- Bitmain se50 BM1684, Debian 9
- Rockchip rk3399pro, Ubuntu 18.04
- Rockchip rv1126, Debian 10
- Rockchip rk3568, Ubuntu 20.04
- Rockchip rk3588, Ubuntu 20.04
- Atlas200(华为官网指定的Ubuntu 18.04版本)
- Amlogic A311D, Ubuntu 20.04
- MLU220, aarch64, Ubuntu 18.04

## 安装依赖

### 依赖包括

- cmake 3+
- gcc 5.4 以上(需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.5 (可选)

**依赖说明：**树莓派 树莓派Raspberry默认为armv7hf系统，使用SDK包中名称中包含 armv7hf\_ARM\_的tar包。如果是aarch64系统，使用SDK包中名称中包含 aarch64\_ARM\_的tar包。

在安装前可通过以下命令查看是32位还是64位：

```
getconf LONG_BIT
32
```

**依赖说明：**比特大陆SE计算盒 需要安装SophonSDK V23.05.01及以上版本，SDK的默认安装位置为 /opt/sophon/，如SDK安装在自定义地址，需在CMakeList.txt中指定SDK安装地址：

```
**这里修改并填入所使用的SophonSDK路径**
set(EDGE_BMSDK_ROOT "{这里填写sdk路径}")
```

可通过命令 `bm-smi` 查看内部SDK和驱动的版本号（SophonSDK V23.05.01对应的内部SDK和驱动为0.4.6）。对于使用旧版BM1684 SDK或者低版本SophonSDK的用户，可参考[SophonSDK安装包](#)中的《LIBSOPHON 使用手册》先卸载旧版BM1684 SDK，安装、升级SophonSDK。

**依赖说明：**海思开发板 海思开发板需要根据海思SDK文档配置开运行环境和编译环境，SDK和opencv都需要在该编译环境中编译。NNIE1.2用 arm-himix200-linux交叉编译好的opencv，下载链接：<https://pan.baidu.com/s/13QW0ReeWx4ZwgYg4Iretyw> 密码:yq0s。下载后修改SDK CMakeList.txt

**依赖说明：**RK3399Pro 所有用例基于 Npu driver版本1.7.1的RK3399pro开发板测试通过，SDK采用预编译模式，请务必确保板上驱动版本为1.7.1 查看RK3399Pro板上driver版本方法：`dpkg -l | grep 3399pro`

**依赖说明：**RV1109/RV1126 所有用例基于Rknn\_server版本1.7.3的RV1126开发板测试通过，SDK采用预编译模式，请务必确保板上驱动版本为1.7.3 查看RV1109/RV1126板上Rknn\_server版本方法：`strings /usr/bin/rknn_server | grep build`

**依赖说明：**RK3568 所有用例基于Rknn\_server版本1.2.0的RK3568开发板测试通过，查看RK3568板上Rknn\_server版本方法：`strings /usr/bin/rknn_server | grep build`

**依赖说明：**RK3588 RK3588开发板需要确保环境正确安装了RKNPU驱动，平台用例基于v0.8.0版本的RKNPU驱动测试通过，查看RK3588NPU驱动版本的方法：`sudo cat /sys/kernel/debug/rknpu/version`

**依赖说明：**晶晨A311D 所有用例基于晶晨A311D开发板测试通过，需要驱动版本为 6.4.4.3（下载驱动请联系开发版厂商）查看晶晨A311D开发板驱动版本方法：`dmesg | grep Galcore`

**依赖说明：**英特尔iGPU 用户在使用英特尔iGPU SDK前，需要根据英特尔[官方文档](#)提前安装好英特尔集成显卡驱动以及相关基础软件环境，安装完成后通过 `clinfo` 指令确认OpenCL能够正常识别到集成显卡信息，正确识别集显情况下clinfo指令输出参考如下：

```
root@baidu-ql1ar9908r-10001:~# cd /opt/
Number of platforms           1
Platform Name                 Intel(R) OpenCL HD Graphics
Platform Vendor               Intel(R) Corporation
Platform Version               OpenCL 3.0
Platform Profile               FULL_PROFILE
Platform Extensions            cl_khr_byte_addressable_store cl_khr_device_uuid cl_khr_fp16 cl_khr_global_int32_base_atomics cl_khr_global_int32_extended_atomics cl_khr_icd cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics cl_khr_command_queue_families cl_khr_intel_required_subgroup_size cl_khr_intel_subgroups_short cl_khr_spir cl_khr_accelerator cl_khr_driver_diag_queries cl_khr_priority_hints cl_khr_throttle_hints cl_khr_create_command_queue cl_khr_intel_subgroups_char cl_khr_intel_subgroups_long cl_khr_il_program cl_khr_intel_mem_force_host_memory cl_khr_subgroup_extended_types cl_khr_subgroup_non_uniform_vote cl_khr_subgroup_ballot cl_khr_subgroup_non_uniform_arithmetic cl_khr_subgroup_shuffle cl_khr_subgroup_shuffle_relative cl_khr_subgroup_clustered_reduce cl_khr_device_attribute_query cl_khr_suggested_local_work_size cl_khr_split_work_group_barrier cl_khr_fp64 cl_khr_subgroups cl_khr_spirv_device_side_ova_motion_estimation cl_khr_spirv_media_block_io cl_khr_spirv_subgroups cl_khr_spirv_no_integer_arp_decoration cl_khr_uniform_shared_memory cl_khr_mipmap_image cl_khr_mipmap_image_writes cl_khr_planar_yuv cl_khr_packed_yuv cl_khr_motion_estimation cl_khr_device_side_ova_motion_estimation cl_khr_subgroup_motion_estimation cl_khr_intel64_base_atomics cl_khr_intel64_extended_atomics cl_khr_image2d_from_buffer cl_khr_depth_images cl_khr_3d_image_writes cl_khr_media_block_io cl_khr_v_api_media_sharing cl_khr_sharing_format_query cl_khr_pci_bus_info
Platform Host timer resolution      1ns
Platform Extensions function suffix      INTEL
Platform Name                       Intel(R) OpenCL HD Graphics
Number of devices                     1
Device Name                           Intel(R) UHD Graphics 630 [0x9bc8]
Device Vendor                           Intel(R) Corporation
Device Vendor ID                         0x9bc8
Device Version                           OpenCL 3.0 NEO
Driver Version                           22.53.25242.13
Device OpenCL C Version                  OpenCL C 1.2
Device Type                               GPU
Device Profile                           FULL_PROFILE
Device Available                         Yes
Compiler Available                       Yes
Linker Available                         Yes
Max compute units                        24
Max clock frequency                       1150MHz
Device Partition                          (core)
```

### 使用序列号激活 请在官网获取序列号

SDK内bin目录下提供预编译二进制文件，可直接运行**(二进制运行详细说明参考下一小节)**，用于图片推理和模型http服务，在二进制参数的serial\_num(或者serial\_key)处填入序列号可自动完成联网激活（请确保硬件首次激活时能够连接公网，如果确实不具备联网条件，需要使用纯离线模式激活，请下载使用百度智能边缘控制台纳管SDK）

```
**SDK内提供的一些二进制文件，填入序列号可完成自动激活，以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}
```

如果是基于源码集成，设置序列号方法如下

```
global_controller()->set_licence_key("")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量或者源码设置）实例数鉴权环境变量设置方法

```
export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30
```

### 实例数鉴权源码设置方法

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)
```

基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

```
对于硬件使用为：Intel Movidius Myriad2 / Myriad X / IGPU on Linux x86_64 / armv7hf / aarch64，在编译或运行demo程序前执行以下命令：
source ${cpp_kit位置路径}/thirdparty/opencv/bin/setupvars.sh
或者执行
source ${cpp_kit位置路径}/thirdparty/opencv/setupvars.sh (opencv-2022.1+)
如果SDK内不包含setupvars.sh脚本，请忽略该提示
```

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号首次激活需要使用，激活后可不用填序列号)

也能运行二进制)

```
**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=../lib ./easyedge_image_inference ../../RES /xxx/cat.jpeg "1111-1111-1111-1111"
```

demo运行效果：



```
> ./easyedge_image_inference ../../RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=../lib ./edgekit_serving --cfg=./edgekit_serving.yml

**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=../lib ./easyedge_serving ../../RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

```
HTTP(or Webservice) is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试，网页右侧会展示模型推理结果

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

同时，可以调用HTTP接口来访问服务。

**请求http服务** 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

```
python
```

```
c#
```

```
C++
```

```
java

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=img.json())

print(result)
```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

### 集成文档

使用该方式，将运行库嵌入到开发者的程序当中。编译demo项目 SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```
cd src
mkdir build && cd build
cmake .. && make
./easyedge_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easyedge_image_inference {模型RES文件夹} {测试图片路径}
```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

对于硬件使用为Intel Movidius Myriad2 / Myriad X 的，如果宿主机找不到神经计算棒Intel® Neural Compute Stick，需要执行以下命令添加USB Rules：

```
cp ${cpp_kit位置路径}/thirdparty/openvino/deployment_tools/inference_engine/external/97-myriad-usbboot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig
```

### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置运行参数
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

对于口罩检测模型，将 `EdgePredictorConfig config` 修改为 `PaddleMultiStageConfig config` 即可。

口罩检测模型请注意输入图片中人脸大小建议保持在 88到9696像素之间，可根据场景远近程度缩放图片后再传入SDK。

**SDK参数配置** SDK的参数通过 `EdgePredictorConfig::set_config` 和 `global_controller()->set_config` 配置。 `set_config` 的所有key在 `easyedge_xxxx_config.h` 中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过 `EdgePredictorConfig::set_config` 设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过 `global_controller()->set_config` 设置

以序列号为为例，KEY的说明如下：

```

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

使用方法如下：

```

EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");

```

具体支持的运行参数可以参考开发工具包中的头文件的详细说明。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR, HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};
```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域, 0代表非目标区域
```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

## 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

- 接口

class `VideoDecoding` :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct `VideoConfig`

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};         // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;           // frame存储为视频文件的路径
    bool save_all{false};            // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

**注意：**1.如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。2.使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

## 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```

EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");

```

## 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

## http服务

1. 开启http服务 http服务的启动参考`demo_serving.cpp`文件。



```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

## 2. http接口详细说明 http 请求方式一：无额外编码 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```

import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果

```

Python请求示例 (视频测试, 注意: 区别于图片预测, 需指定Content-Type; 否则会调用图片推理接口)

```

import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data).json()

```

http 请求方法二：json格式，图片传base64格式字符串 HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 图像分类网络：body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 物体检测和实例分割网络：Body请求示例：

```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)

```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    },
  ]
}
```

#### 其他配置

##### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



## FAQ

### 1. 如何处理一些 undefined reference / error while loading shared libraries?

如：`./easyedge_demo: error while loading shared libraries: libeasyedge.so.1: cannot open shared object file: No such file or directory` 这是因为二进制运行时ld无法找到依赖的库。如果是正确cmake && make 的程序，会自动处理好链接，一般不会出现此类问题。

遇到该问题时，请找到具体的库的位置，设置LD\_LIBRARY\_PATH。

示例一：`libverify.so.1: cannot open shared object file: No such file or directory` 链接找不到libveirfy.so文件，一般可通过 `export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib` 解决(实际冒号后面添加的路径以libverify.so文件所在的路径为准)

示例二：`libopencv_videoio.so.4.5: cannot open shared object file: No such file or directory` 链接找不到libopencv\_videoio.so文件，一般可通过 `export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/thirdparty/opencv/lib` 解决(实际冒号后面添加的路径以libopencv\_videoio.so所在路径为准)

### 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

### 3. 如何将我的模型运行为一个http服务？

目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

### 4. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

### 5. 运行SDK报错 Authorization failed

情况一：日志显示 `Http perform failed: null respond` 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示`failed to get/check device id(xxx)`或者`Device fingerprint mismatch(xxx)` 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

### 6. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

7. 运行NNIE引擎报错 `std::bad_alloc` 检查开发板可用内存，一些比较大的网络占用内存较多，推荐内存500M以上

8. 运行二进制时，提示 `libverify.so cannot open shared object file`

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 编译时报错：`file format not recognized` 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中，再解压缩、编译

🔗 Linux集成文档-Python

## 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法，适用于 EasyDL 通用版和BML。

- 网络类型支持：图像分类，物体检测
- 硬件支持：
  - Intel Movidius Myriad2 / Myriad X / IGPU
  - 瑞芯微 RK3399Pro
- 语言支持：*Intel Movidius Myriad2 / Myriad X / IGPU: Python 3.5, 3.6, 3.7* 瑞芯微 RK3399Pro: Python 3.6

## Release Notes

时间	版本	说明
2022.10.27	1.3.5	新增Arm7 CPU、Arm8 CPU、Jetson、华为昇腾Atlas开发板对应Python SDK，支持图像分类、物体检测、人脸检测、实例分割；新增 Intel IGPU 支持
2022.05.18	1.3.0	新增RK3399Pro NPU对应Python SDK，支持图像分类、物体检测
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持加速棒
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.07.19	1.1.7	提供模型更新工具
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】 序列号配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。

**依赖说明：** Intel Movidius 加速棒 使用Intel Movidius加速棒 SDK、 Intel IGPU 预测时，必须安装 OpenVINO 预测引擎，两种方式：

- 使用 OpenVINO™ toolkit 安装，请参考 [OpenVINO toolkit 文档](#)安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装，请参考 [Openvino Inference Engine文档](#)编译安装 2020.3.1 (必须) 版本。

安装完毕，运行之前，请按照OpenVino的文档 设置环境变量

```
source /opt/intel/openvino/bin/setupvars.sh
```

**依赖说明：** RK3399Pro 所有用例基于 Npu driver版本1.7.3的RK3399pro开发板测试通过 查看RK3399Pro板上driver版本方法：运行sdk内提供demo项目，日志里会提供API和Driver版本信息

```
2022-12-20 14:26:07,765 VERBOSE [EasyEdge] [rockchip_edge_predictor.cpp:87] 547887054864 Create predictor , 5029536
D RKNNAPI: =====
D RKNNAPI: RKNN VERSION:
D RKNNAPI: API: 1.7.3 (0cfd4a1 build: 2022-08-15 17:10:10)
D RKNNAPI: DRV: 1.7.3 (c4ea832 build: 2022-08-13 09:13:08)
D RKNNAPI: =====
```

升级399Pro driver版本参考瑞芯微github：[https://github.com/airockchip/RK3399Pro\\_npu](https://github.com/airockchip/RK3399Pro_npu) 2. 安装 easyedge python wheel 包 安装说明：Intel Movidius 加速棒 / Intel IGPU

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。

**安装说明：RK3399Pro**

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_RK3399Pro-{版本号}-cp36-cp36m-linux_aarch64.whl
```

具体名称以 SDK 包中的 whl 为准，特别注意这里要同时安装两个whl包 **安装说明：ArmV7 CPU**

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_armv7l.whl
pip3 install -U EasyEdge_Devkit_ARM-{版本号}-cp36-cp36m-linux_armv7l.whl
```

**安装说明：ArmV8 CPU (Aarch64 CPU)**

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_ARM-{版本号}-cp36-cp36m-linux_aarch64.whl
```

**安装说明：Jetson SDK**

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
pip3 install -U EasyEdge_Devkit_JetPack{版本号}-{版本号}-cp36-cp36m-linux_aarch64.whl
```

**安装说明：华为昇腾Atlas开发板**

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_aarch64.whl
EasyEdge_Devkit_Atlas200-{版本号}-cp36-cp36m-linux_aarch64.whl
```

### 3. 使用序列号激活



#### 获取序列号

此发布、下载的SDK为未授权SDK，需要前往控制台获取序列号激活后才能正常使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标test	134318-v1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精调无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精调无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
		基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>	

修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的设置），需要调用函数指定实例数鉴权模式，并且实例数鉴权模式下，支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，参考

```
pred.set_instance_auth_mode()
pred.set_instance_update_interval(200)
```

#### 4. 测试 Demo

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



#### 使用说明

#### 使用流程

```
import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.MOVIDIUS, engine=edge.Engine.OPENVINO)
pred.infer_image((numpy.ndarray的图片))
pred.close()
```

#### 初始化

- 接口

```
def init(self,
          model_dir,
          device=Device.LOCAL,
          engine=Engine.PADDLE_FLUID,
          config_file='conf.json',
          preprocess_file='preprocess_args.json',
          model_file='model',
          params_file='params',
          graph_file='graph.ncsmodel',
          label_file='label_list.txt',
          device_id=0
        ):
    """
    Args:
        device: Device.CPU
        engine: Engine.PADDLE_FLUID
        model_dir: str
            model dir
        preprocess_file: str
        model_file: str
        params_file: str
        graph_file: str
        label_file: str
        device_id: int

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success

    """
```

#### 预测图像



- 接口

```
def infer_image(self, img,
                threshold=0.3,
                channel_order='HWC',
                color_format='BGR',
                data_type='numpy'):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

**升级模型** 适用于经典版升级模型，执行bash update\_model.sh，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

## FAQ

### Q: 运行SDK报错 Authorization failed

**情况一：日志显示 Http perform failed: null respond** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx)** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

**情况三：ArmV7、ArmV8 CPU、Jetson、Atlas Python SDK日志提示ImportError: libavformat.so.58: cannot open shared object file: No such file or**

directory 或者其他类似so找不到 可以在LD\_LIBRARY\_PATH环境变量加上libs和thirdpartylibs路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内libs和thirdpartylibs路径的方法如下(以Atlas SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas200 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## Linux集成文档-Atlas

### 简介

本文档介绍EasyEdge/EasyDL的Linux Atlas SDK的使用方法。

注意Atlas有两种产品形态，Atlas 200和Atlas 300，请参见此处的[文档说明](#)

- 网络类型支持：图像分类
- 硬件支持：
  - CPU: aarch64
  - Atlas 200 卡
- 操作系统支持：Atlas指定的Linux版本，Ubuntu 16.04 aarch64，请从Atlas文档中下载。

### 性能数据

数据仅供参考，实际数值根据使用线程数、利用率等情况可能有所波动

模型类型	模型算法	芯片类型	SDK类型	实测硬件	单次预测耗时
EasyDL 图像分类	高性能	Atlas 200	Atlas 200	Atlas 200DK	9ms
EasyDL 图像分类	高精度	Atlas 200	Atlas 200	Atlas 200DK	12ms
EasyDL 物体检测	高性能	Atlas 200	Atlas 200	Atlas 200DK	11ms
EasyDL 物体检测	高精度	Atlas 200	Atlas 200	Atlas 200DK	31ms

### Release Notes

时间	版本	说明
2020.6.15	0.2	支持物体检测
2020.3.10	0.1	初始版本，支持图像分类

### 测试atlas 200的官方demo

请参见此处的[文档说明](#)，搭建开发环境，测试atlas 200的mindstudio demo通过后，再测试

### 快速开始

SDK在以下环境中测试通过

- ubuntu 16.04, aarch64-linux-gnu-g++ 5.4，编译器
- ubuntu 16.04，开发板

Atlas DDK 的ddk\_info信息：

```
{
  "VERSION": "1.3.T34.B891",
  "NAME": "DDK",
  "TARGET": "Atlas DK"
}
```

## 2. 测试Demo

**编译运行：** 下载后，模型资源文件默认已经打包在开发者下载的SDK包中，

Step 1：运行一次unpack.sh脚本，会得到测试demo。

Step 2：请在官网获取序列号，填写在demo\_async.cpp及demo\_sync.cpp的开始处license\_key字段。



step3：准备测试图片

覆盖image目录下的 1.jpg，更多图片可以用于demo中的批量测试模式

step4：修改test\_200.sh下的以下开发板登录信息

```
export DDK_PATH=$HOME/tools/che/ddk/ddk # ddk的安装路径
SSH_USER=HwHiAiUser@192.168.3.25 # 200 开发板的ssh登录信息
PORT=8822 # 200 开发板的ssh登录端口
```

step: 运行demo，会自动编译OpenCV 3.4库

```
cd demo
sh test_200.sh
```

图像分类的demo运行效果：

```
[stat] [100001]image/1.jpg(4 images) time used: 41ms (at 1583765958531) total:705ms
[result][100001]image/1.jpg[281470472005664] is: n07747607 orange 0.973633 950;

n07747607 orange 分类名
0.973633 分类概率
950 分类名的序号
```

物体检测的demo运行效果：

```
[stat] time used: 101ms; all time used:478
images[3] result:
label:no2_ynen:prob:0.985352 loc:[(0.459961,0.839844), (0.5625,0.988281)]

no2_ynen 分类名，也可以获取分类名的序号
0.985352 分类概率
loc:[(0.459961,0.839844), (0.5625,0.988281)]，检测框的位置。(0.459961,0.839844)表示左上角的点，(0.5625,0.988281)右下角的点；
如原始图片608，左上角(0.459961*608,0.839844*608)，右下角(0.5625*608,0.988281*608)
```

### SDK接口使用

使用该方式，将运行库嵌入到开发者的程序当中。

### 同步接口使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ;
auto predictor = global_controller()->CreateEdgePredictor(config);
int ret = predictor->init();
# 若返回非0, 请查看输出日志排查错误原因。
auto img = cv::imread({图片路径});
// step 3: 预测图像
std::vector<EdgeResultData> result2;
predictor->infer(img, result2);
# 解析result2即可获取结果

```

## 异步接口使用流程

```

// step 0: 设置序列号
global_controller()->set_licence_key("set your license here");

// step 1: 配置模型资源目录
AtlasConfig config;
config.model_dir = {模型文件目录};

// step 3: 创建Predictor ; 这这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 4: 设置异步回调
predictor->set_result_handler(YOUR_HANDLER);

// step 5: 初始化
int ret = predictor->init();
**若返回非0, 请查看输出日志排查错误原因。 **

// step 6: 预测图像
auto img = cv::imread({图片路径});
color_format = kBGR;
float threshold = 0.1;

uint64_t seq_id;
predictor->infer_async(img, color_format, 0.1, nullptr, seq_id);
**YOUR_HANDLER里面有seq_id的回调结果**

```

## 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

- 接口

```
virtual int set_licence_key(const std::string& license) = 0;
```

## 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

## FAQ

1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3`

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

## 2. EasyDL 离线SDK与云服务效果不一致，如何处理？

目前离线SDK与云服务的处理有些许差异，具体如下：

- 图像分类模型：离线SDK与云服务使用通用(非快速训练、非AutoDL Transfer)的效果类似
- 物体检测模型：离线SDK的高精度模型与云服务的精度较低，服务性能更佳的效果类似

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

## 端云协同服务说明

### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 联网状态下在平台管理设备运行状态、资源利用率

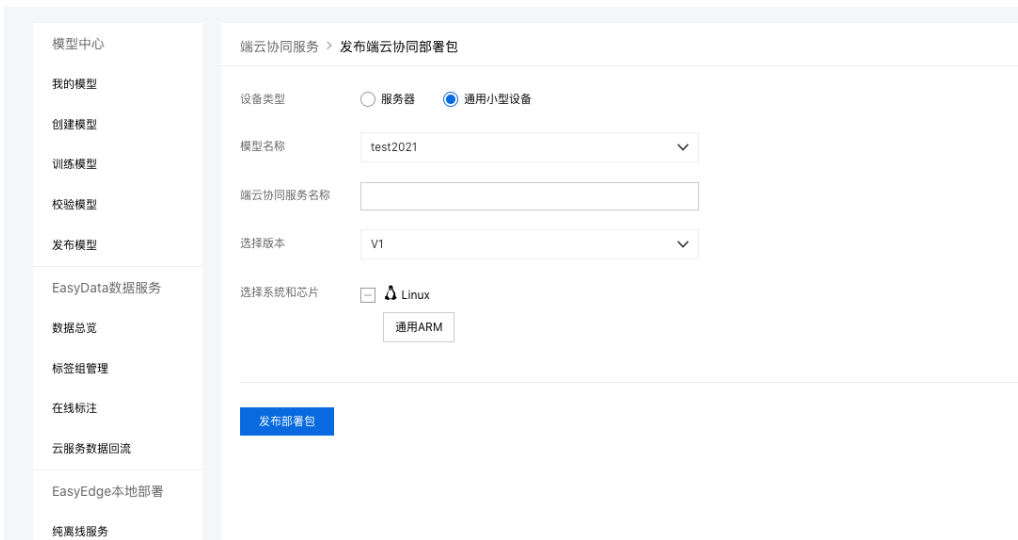
目前通用小型设备的应用平台支持Linux-ARM，具体使用流程请参考下方文档。

### 使用流程

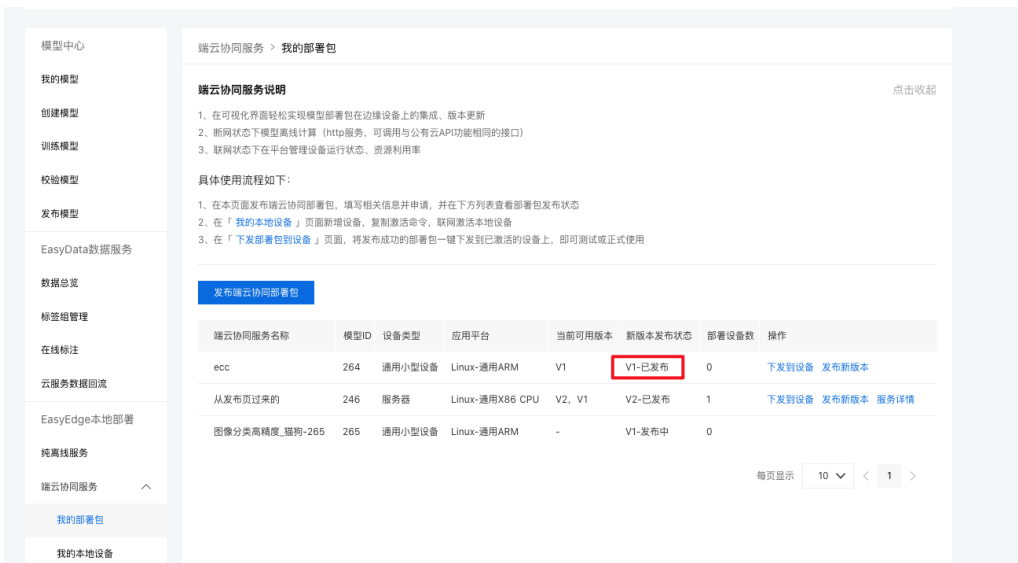
#### Step 1 发布端云协同部署包

在[我的部署包](#)页面点击「发布端云协同部署包」

填写服务名称，选择模型版本并提交发布



在列表查看部署包发布状态



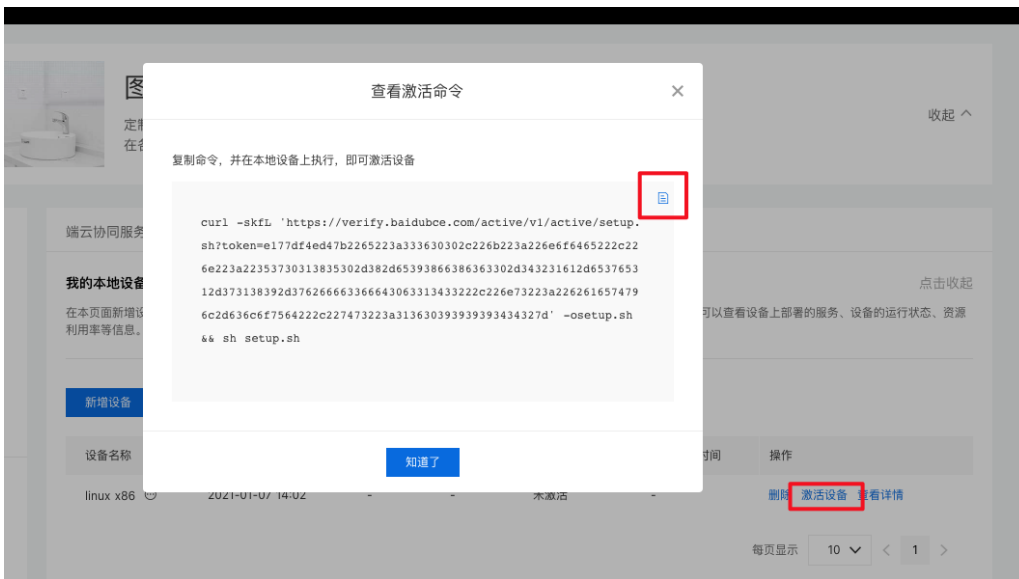
### Step 2 新增设备并激活

在**我的本地设备**页面新增设备





在列表中，点击设备对应的「激活设备」操作，复制激活命令并在本地设备上执行即可



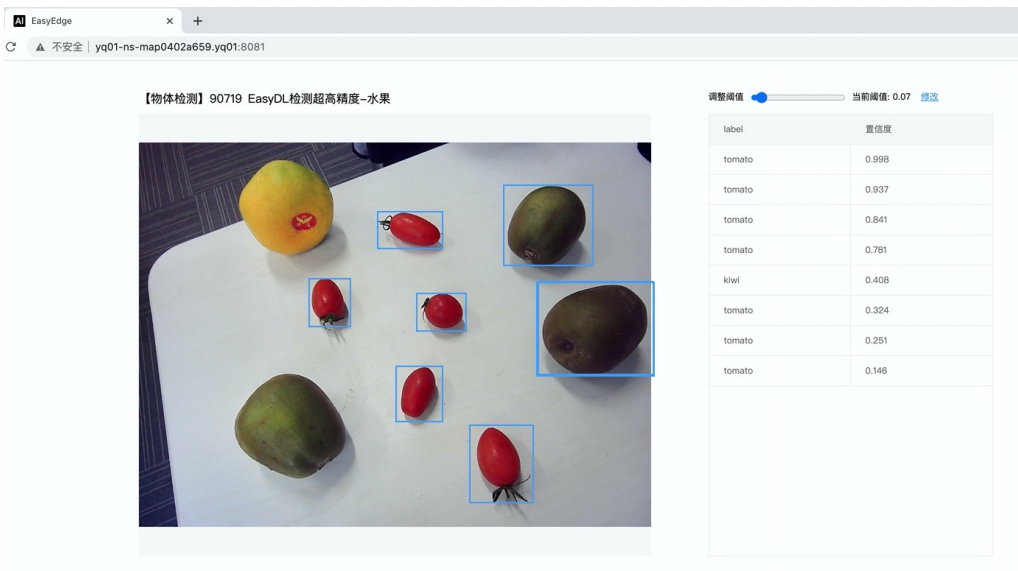
### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入 `http://{设备ip}:{服务端点, 默认8080}`，即可预览效果：





具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

### 云端管理说明

### 模型部署包管理

在[我的部署包](#)页面可以进行已发布的模型部署包的管理。

### 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。



新版本发布成功后，即可在「下发部署包到设备」页面或当前服务的「服务详情」页面，将新版本下发到本地设备上。



## 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。

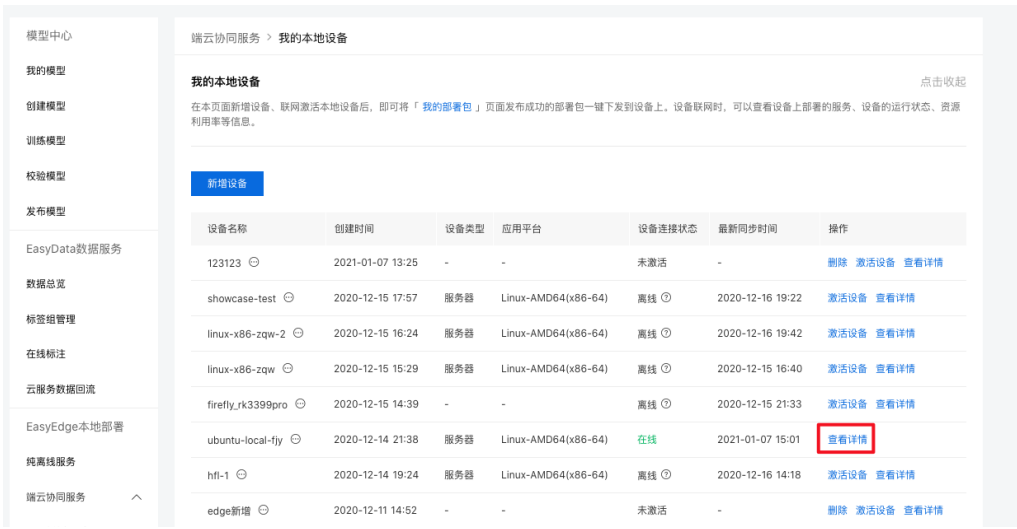


## 本地设备管理

在[我的本地设备](#)页面可以进行所有本地设备的管理。

## 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：



设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况



### ☞ 软硬一体方案部署

### ☞ 如何获取物体检测软硬一体产品

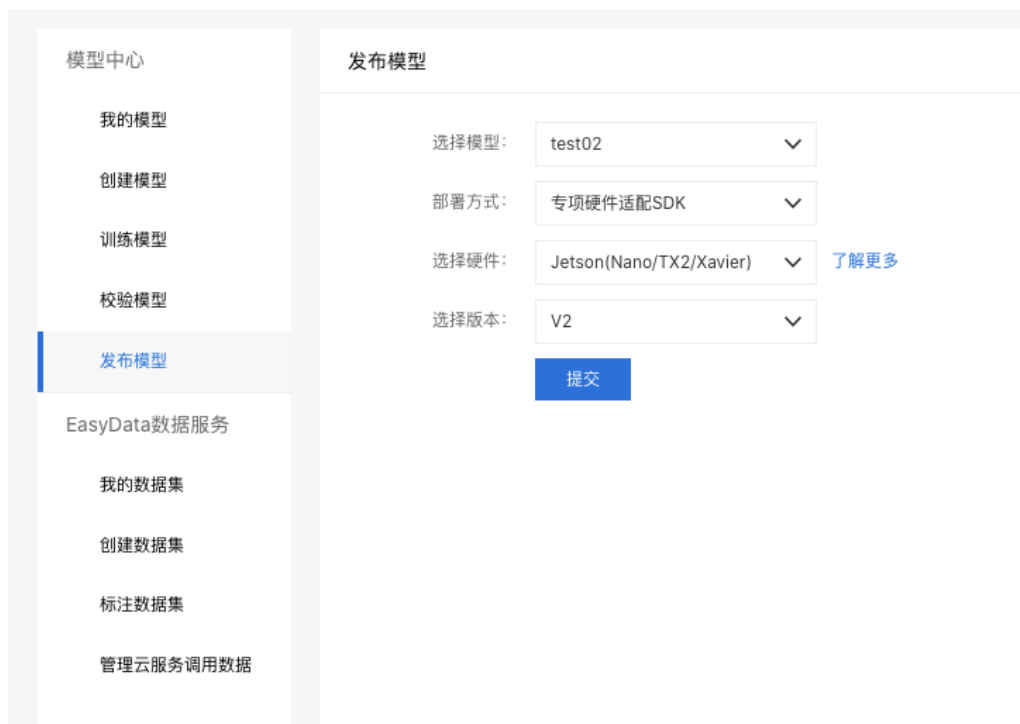
为进一步提升前端智能计算的用户体验，EasyDL推出了多款软硬一体方案。将高性能硬件与EasyDL图像分类/物体检测模型深度适配，可应用于工业分拣、视频监控等多种设备端离线计算场景，让离线AI落地更轻松。[了解不同方案](#)

方案获取流程如下：

Step 1：在EasyDL训练专项适配所选硬件的图像分类/物体检测模型，迭代模型至效果满足业务要求



Step 2：发布模型时选择对应硬件



Step 3 : 在AI市场购买方案获得硬件和用于激活专用SDK的专用序列号，参考文档集成后，即可实现离线AI预测



如有其他硬件方案需求，请在百度智能云控制台内[提交工单](#)反馈。

#### 🔗 物体检测EdgeBoard(FZ)专用SDK集成文档

##### 简介

本文档介绍 EasyEdge/EasyDL在EdgeBoard®边缘计算盒/Lite计算卡上的专用软件的使用流程。

EdgeBoard系列硬件可直接应用于AI项目研发与部署，具有高性能、易携带、通用性强、开发简单等四大优点。

详细硬件参数请在[AI市场](#)浏览。

EdgeBoard产品使用手册：<https://ai.baidu.com/ai-doc/HWCE/Yk3b86gvp>

##### 软核版本

CPP-SDK版本	对应软核
1.3.2、1.3.4、1.3.5	1.8.1
1.3.0、1.3.1、1.3.2、1.3.4	1.8
0.5.7-1.2.1	1.5
0.5.2+	1.4

SDK升级需配合EdgeBoard硬件软核升级，建议升级软核为SDK对应版本，否则可能出现结果错误或者其他异常。

可以通过 `dmesg | grep "DRIVER Version"` 命令获取EdgeBoard当前的软核版本

**Release Notes 注意\***：升级完成相应的软核之后需要重启机器生效。

sdk对应的软核说明：**如果客户使用的软核是mobile版本的，需要使用1.4的SDK；如果不是mobile 版本，可以选择1.5+（目前最高版本更新至1.8.1）版本的SDK使用。**

1.5+版本的软核以及sdk更新情况如下表所示：

时间	版本	说明	EdgeBoard非mobile对应的软核以及特性	
2021.1 2.20	1.3.5	升预测引擎为PaddleLite 1.8.1,推理库支持了Ubuntu18.04文件系统	<a href="https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk">https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk</a> (含有EB升级Ubuntu18.04系统的步骤)	
2021.1 0.15	1.3.2、 1.3.4、1.3.4	推理库支持了Ubuntu18.04文件系统	<a href="https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk">https://ai.baidu.com/ai-doc/HWCE/Fkuqounlk</a>	
2021.0 6.29	1.3.1	视频流解析支持分辨率调整；预测引擎升级；	<a href="https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x">https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x</a>	
2021.0 5.14	1.3.0	新增视频流接入支持；展示已发布模型性能评估报告	<a href="https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x">https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x</a>	
2021.0 5.14	1.2.1	功能无更新	<a href="https://ai.baidu.com/ai-doc/HWCE/okqiwkm32">https://ai.baidu.com/ai-doc/HWCE/okqiwkm32</a>	
2020.1 0.29	0.5.7	预测引擎切换为PaddleLite 1.5		-
2019.1 2.27	0.4.5	引擎升级，支持zu5/zu3，支持EasyDL 高精度检测模型		-
2019.0 7.25	0.4.0	EdgeBoard SDK Release!		-

mobile软核以及sdk更新情况如下表所示：| 时间 | 版本 | 说明 | EdgeBoard mobile对应的软核以及特性 | | --- | --- | --- | --- | --- |  
|2021.05.14|1.2.1|功能无更新|<https://ai.baidu.com/ai-doc/HWCE/okqiwkm32>|<https://ai.baidu.com/ai-doc/HWCE/Lkqiwlziw>|

## 快速开始

开发者从EasyEdge/EasyDL下载的软件部署包中，包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行EdgeBoard计算盒。

部署包中包含多版本SDK：

- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.8*`：适用于EdgeBoard 1.5+软核
- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.4*`：适用于EdgeBoard 1.4软核

SDK文件结构

```

baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.5_*
├── README.txt
├── bin
│   ├── easyedge_image_inference
│   ├── easyedge_serving
│   └── easyedge_video_inference
├── include
│   └── easyedge
├── lib
│   ├── libeasyedge.so -> libeasyedge.so.1
│   ├── libeasyedge.so.1 -> libeasyedge.so.1.3.1
│   ├── libeasyedge.so.1.3.1
│   ├── libeasyedge_static.a
│   ├── libeasyedge_videoio.so -> libeasyedge_videoio.so.1
│   ├── libeasyedge_videoio.so.1 -> libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio_static.a
│   ├── libpaddle_full_api_shared.so -> libpaddle_full_api_shared.so.1.8.0
│   ├── libpaddle_full_api_shared.so.1.8.0
│   ├── libverify.so -> libverify.so.1
│   ├── libverify.so.1 -> libverify.so.1.0.0
│   └── libverify.so.1.0.0
├── now_sre.log
├── src
│   ├── CMakeLists.txt
│   ├── cmake
│   ├── common
│   ├── demo_image_inference
│   ├── demo_serving
│   └── demo_video_inference
└── thirdparty
    └── opencv

```

1.1.0+的SDK自带OpenCV，src编译的时候会引用thirdparty/opencv路径下的头文件和库文件。

## Demo使用流程

用户在AI市场购买计算盒之后，请参考以下步骤进行集成和试用。

### 1. 将计算盒连接电源

指示灯亮起，等待约1分钟。

- 参考[EdgeBoard使用文档](#)配置网口或串口连接。登录EdgeBoard计算盒。
- 加载驱动（开机加载一次即可）。

```
insmod /home/root/workspace/driver/{zu9|zu5|zu3}/fpgadv.ko
```

根据购买的版本，选择合适的驱动。若未加载驱动，可能报错：

```
Failed to to fpga device: -1
```

- 设置系统时间（系统时间必须正确）

```
date --set "2019-5-18 20:48:00"
```

### 2. (可选) 启动HTTP服务

部署包中附带了HTTP服务功能，开发者可以进入SDK根目录，运行easyedge\_serving程序启动HTTP服务。

```

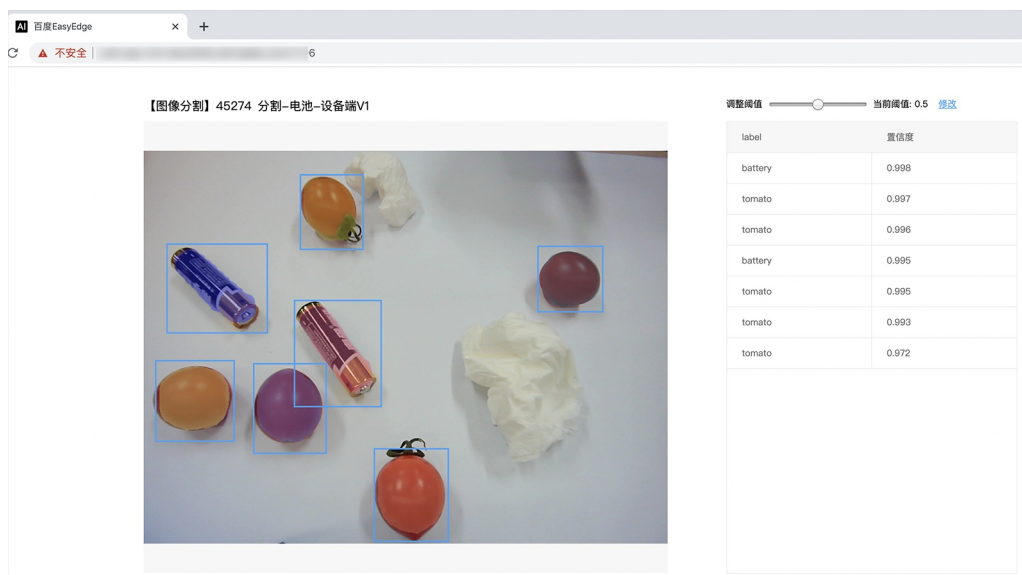
**./easyedge_serving {RES目录} "" {绑定的host, 默认0.0.0.0} {绑定的端口, 默认24401}**
cd ${SDK_ROOT}
export LD_LIBRARY_PATH=./lib
./demo/easyedge_serving ../../RES ""

```

日志显示

```
2019-07-18 13:27:05,941 INFO [EasyEdge] [http_server.cpp:136] 547974369280 Serving at 0.0.0.0:24401
```

则启动成功。此时可直接在浏览器中输入 `http://{EdgeBoard计算盒ip地址}:24401/`，在h5中测试模型效果。



同时，可以调用HTTP接口来访问盒子。具体参考下文接口说明。

EdgeBoard HTTP Server 目前使用的是单线程处理请求。

### 3. 编译运行Demo

编译：

```
cd src
mkdir build && cd build
cmake .. && make
```

运行

```
./easyedge_image_inference {RES资源文件夹路径} {测试图片路径}
```

便可看到识别结果。

使用说明

使用流程

激活成功之后，有效期内可离线使用。

1. 配置PaddleFluidConfig
2. 新建Predictor :global\_controller()->CreateEdgePredictor(config);
3. 初始化 predictor->init()
4. 传入图片开始识别predictor->infer(img, ...);

目前EdgeBoard暂不支持并行多模型计算。

接口说明

预测图片

```

/**
 * @brief 同步预测接口
 * inference synchronous
 * Supported by most chip and engine
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @param threshold
 * @return
 */
virtual int infer(
    cv::Mat &image, std::vector<EdgeResultData> &result, float threshold = 0.1
) = 0;

```

## 识别结果说明

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // object detection field
    float x1, y1, x2, y2; // (x1, y1): 左上角 , (x2, y2): 右下角 ; 均为0~1的长宽比例值。
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考demo文件中使用opencv绘制矩形的逻辑。

## HTTP 私有服务请求说明

### http 请求参数

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤， 0~1	0.1

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

### Python

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

### Cpp label=C#



```
    FileStream fs = new FileStream("./img.jpg", FileMode.Open);
    BinaryReader br = new BinaryReader(fs);
    byte[] img = br.ReadBytes((int)fs.Length);
    br.Close();
    fs.Close();
    string url = "http://127.0.0.1:8402?threshold=0.1";
    HttpRequest request = (HttpRequest)HttpRequest.Create(url);
    request.Method = "POST";
    Stream stream = request.GetRequestStream();
    stream.Write(img, 0, img.Length);
    stream.Close();

    HttpResponse response = request.GetResponse();
    StreamReader sr = new StreamReader(response.GetResponseStream());
    Console.WriteLine(sr.ReadToEnd());
    sr.Close();
    response.Close();
```

Cpp label=C++ 需要安装curl

```

#include <sys/stat.h>
#include <curl/curl.h>
#include <iostream>
#include <string>
#define S_ISREG(m) (((m) & 0170000) == (0100000))
#define S_ISDIR(m) (((m) & 0170000) == (0040000))

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"
", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"
", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s
", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

### Java请求示例

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

- 接口

class `VideoDecoding` :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};          // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;            // frame存储为视频文件的路径
    bool save_all{false};             // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

- 3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 错误说明

SDK所有主动报出的错误，均覆盖在EdgeStatus枚举中。同时SDK会有详细的错误日志，开发者可以打开Debug日志查看额外说明：

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

#### FAQ

##### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

可以通过安装`libcurl3 libcurl-openssl1.0-dev`来解决。如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库`easyedge_static.a`，自己指定需要的Library的版本。

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} paddle-mobile)
```

##### 2. error while loading shared libraries: libeasyedge.so.0.4.0: cannot open shared object file: No such file or directory

类似错误包括`libpaddle-mobile.so`找不到。

直接运行SDK自带的二进制可能会有这个问题，设置`LD_LIBRARY_PATH`为SDK部署包中的lib目录即可。开发者自行使用CMake编译的二进制可以有效管理.so的依赖。

##### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

##### 4. 预测过程中报内存不足“Killed”

此问题仅出现在ZU5，因为FZ5A带vcu，给他预留的内存过大导致，如果用不到VCU可以把这部分改小。修改`/run/media/mmcblk1p1/uEnv.txt`：

```
ethaddr=00:0a:35:00:00:09
uenvcmd=fatload mmc 1 0x3000000 image.ub && bootm 0x3000000

bootargs=earlycon console=ttyPS0,115200 clk_ignore_unused cpuidle.off=1 root=/dev/mmcblk1p2 rw rootwait cma=128M
```

注意中间空行要保留。

##### 5. 预测结果异常

如果购买的计算盒较早，驱动文件较旧，而SDK比较新（或SDK比较旧，但是计算盒较新），可能出现结果异常，如结果均为空或者nan。请参

考“软核版本”小节更新软核和驱动版本。

## 6. 编译过程报错file format not recognized

```
libeasymedge.so: file format not recognized; treating as linker script
```

下载的SDK zip包需要放到板子内部后，再解压、编译。

7. 提示 driver\_version(1.4.0) not match paddle\_lite\_version(1.5.1) 需更新驱动，否则可能导致结果异常。参考“软核版本”小节。

## 🔗 物体检测EdgeBoard(VMX)专用SDK集成文档

### 简介

本文档旨在介绍 EasyDL在EdgeBoard USB加速卡VMX（以下简称VMX加速卡或加速卡）上的专用软件的使用流程。EdgeBoard系列硬件适用于项目开发及部署，具有高性能、易携带、通用性强、开发简单等四大优点。您可在[AI市场](#)了解EdgeBoard相关系列产品，同时可以在[软硬一体方案](#)了解性能数据。

注意：本型号主要面向产品集成和企业项目，未同时售卖散热片和外壳，部分情况下芯片温度较高，**开发过程中，请勿用手触摸，谨防烫伤**

### 硬件介绍

VMX加速卡，采用Intel® Movidius™ 视觉 MyriadX处理器芯片，通过 USB3.0 通讯type-c接口方式，配合外围电路即可将该模组嵌入到第三方智能化产品中，采用标准 USB通讯协议，对接简单，开发速度快，具有强大的深度学习计算功能。可通过OpenVINO™和OpenCV软件库工具链移植算法，兼容百度PaddlePaddle支持Paddle2onnx和PaddleHub并集成EasyDL，使产品应用范围广，性能更稳定，增强用户体验。

VMX加速卡适用于深度学习加速，能够解决复杂的人工智能软硬件设计挑战，它可以集成基于视觉的加速器和推理引擎来实现深度边缘学习的解决方案。（3D/2D人脸识别、人头检测、人脸属性分析（性别、年龄）、人脸特征比对、手势及姿态识别、物体检测及分类、算法移植等功能。）

### 硬件配置与说明

核心板模块: Intel® Movidius™ MyriadX，内置内存LP-DDR4 4GBit。

#### • 硬件指标

#### CPU

- o Intel® Movidius Myriad X MA2485 Vision Processing Unit
- o Total performance of over 4 trillion operations per second (TOPS)
- o Over 1 TOPS performance on neural network inference w/ NCE accelerator
- o 16 Programmable 128-bit VLIW Vector Processors
- o 16 Configurable MIPI Lanes w/ enhanced Vision Accelerators
- o 2.5 MB of Homogenous On-Chip Memory w/ 4Gbit LPDDR4

#### Size

- o 38mm x 38mm

**Interface** o USB TYPE C (USB3.0) 辅助接口精简设计

**Boot** o USB 启动模式 - 内置 switch 缺省模式设置

**Power** o 平均功耗0.5W~2.2W

**Security** o 支持 eFuse 加密

### 运行说明

VMX加速卡包含独立的AI运算芯片，采用 USB Type-C通讯方式，通讯协议简单可靠，可连接不同芯片架构主机，包括 X86、ARM SOC等。加速卡运行需要通过TypeC接口连接宿主机执行，宿主机目前支持的软硬件环境包括：

- Linux: x86-64, armv7hf
- Windows: x86-64, Windows 10

使用过程中，请尽量避免直接接触板卡元器件；或者使用防静电锡纸包裹板卡。

### 快速开始 Linux

开发者从EasyDL训练模型之后，下载的软件部署包中，包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行。

## Release Notes

### Python SDK

时间	版本	说明
2020.12.18	1.2.0	性能优化；接口优化升级；推理引擎升级
2020.09.17	1.1.19	支持更多模型与平台

Python SDK适用于Linux x86-64和Windows平台。

2020-12-18: 【接口升级】 Python SDK序列号配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

### C++ SDK

时间	版本	说明
2021.06.29	1.3.1	视频流解析支持分辨率调整
2021.05.14	1.3.0	新增视频流接入支持；展示已发布模型性能评估报告
2020.12.18	1.0.0	性能优化；接口优化升级；推理引擎升级
2020.09.17	0.5.6	新增C++ SDK，支持Linux armv7hf（树莓派）架构的硬件接入VMX预测

C++ SDK适用于Linux x86-64和Linux armv7hf平台。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

**将加速卡连接宿主机** 请使用质量合规的usb线连接。连接之后，检查设备是否被操作系统识别：Linux 通过 `lsusb -v` 命令检查是否有 Myriad设备：

```
> sudo lsusb -v | grep -C 5 Myriad
bMaxPacketSize0    64
idVendor           0x03e7
idProduct          0x2485
bcdDevice          0.01
iManufacturer      1 Movidius Ltd.
iProduct           2 Movidius MyriadX
iSerial            3 03e72485
```

Windows 可以在设备管理器中查询。

如果使用 VirtualBox 之类的虚拟机，请在虚拟机加入 03e7:24 和 03e7:f63b 两个 usb 设备。

## 获取并安装依赖

### 1) 安装依赖

宿主机与sdk为以下情况：1  Windows x86-64：请参考 [OpenVINO toolkit 文档](#) 安装 2022.1.1LTS 版本 2  Linux x86-64：请参考 [OpenVINO toolkit 文档](#) 安装 2022.1.1LTS 版本, 安装时可忽略Configure the Model Optimizer及后续部分。

注：1. ebvpu+arm 组合产品正在下架过渡期，后续不再维护 armv7 版本的 sdk；2. 当前问题可以升级 openvino 版本解决，但无法保证 armv7 环境下的稳定性；

安装完毕，运行之前，请按照OpenVino的文档 设置环境变量

```
source /opt/intel/opencv/bin/setupvars.sh
```

2) 从EasyDL 控制台获取SDK 在任意位置解压缩。

获取序列号 从[AI市场订单详情](#)或者[EasyDL控制台](#)获取序列号。

更换序列号、更换设备时，首次使用需要联网激活。激活成功之后，有效期内可离线使用。

请确保激活设备时使用的 操作系统账号与后续使用时运行的账号一致，否则会造成验证失败

## Python SDK

### 1. 安装wheel包

```
pip3 install -U BaiduAI_EasyEdge_SDK-[版本号]-cp37-cp37m-linux_x86_64.whl
```

注意，请根据python的版本选择对应的whl文件，其中,1.2.0是SDK版本号，cp37表示是python3.7版本

--

注意，pip安装时请添加-U参数

### 2. 将步骤2中获得的序列号 填入demo.py

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的设置），需要调用函数指定实例数鉴权模式，并且实例数鉴权模式下，支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，参考

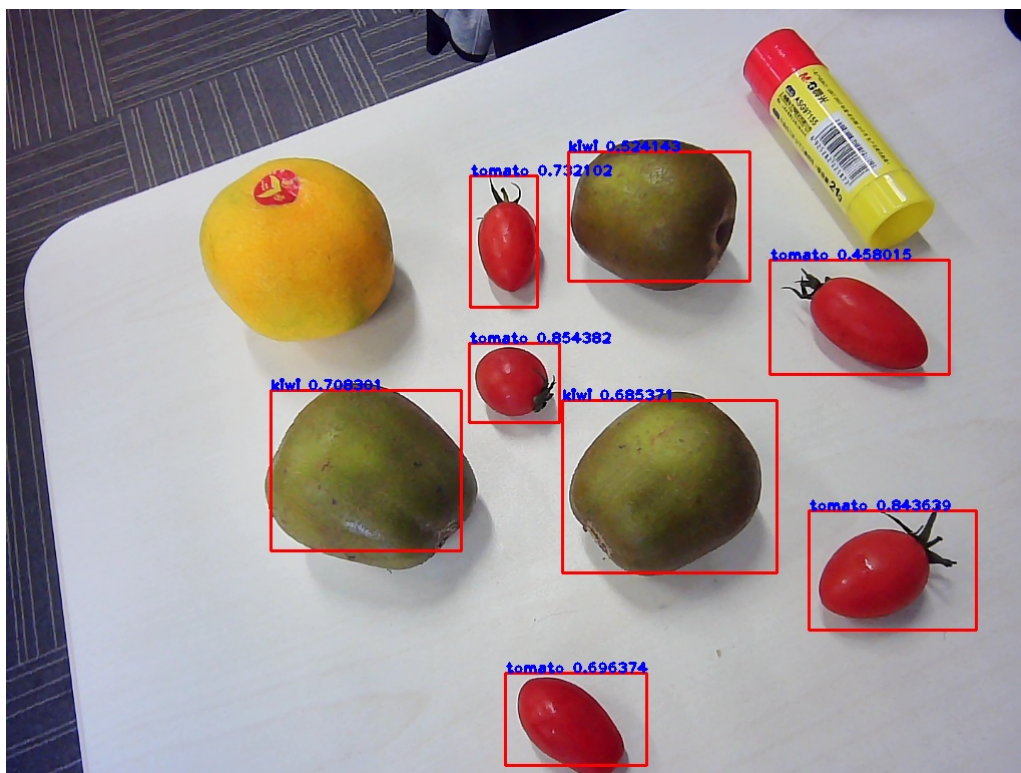
```
pred.set_instance_auth_mode()
pred.set_instance_update_interval(200)
```

### 3. 测试demo.py

```
python3 demo.py {模型资源文件夹RES路径} {待识别的图片路径}
```

生成的样例结果图片如下：





## 使用流程

```
import BaiduAI.EasyEdge as edge
```

```
pred = edge.Program()
```

```
pred.set_auth_license_key("这里填写序列号")
```

```
pred.init(model_dir={RES文件夹路径}, device=edge.Device.MOVIDIUS, engine=edge.Engine.OPENVINO)
```

```
pred.infer_image((numpy.ndarray的图片))
```

```
pred.close()
```

接口的详细说明请主要参考 SDK 中的接口注释

## 接口说明

### Program

- 初始化

```

def init(self,
    model_dir,
    device=Device.CPU,
    engine=Engine.NCSDK,
    config_file='conf.json',
    preprocess_file='preprocess_args.json',
    model_file='model',
    params_file='params',
    graph_file='graph.ncsmodel',
    label_file='label_list.txt',
    device_id=0,
    **kwargs
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device
        engine: BaiduAI.EasyEdge.Engine
        preprocess_file: str
        model_file: str
        params_file: str
        graph_file: str ncs的模型文件 或 PaddleV2的模型文件
        label_file: str
        device_id: int 设备ID
        thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success

    """

```

- 预测单张图像

```

def infer_image(self, img, threshold=None,
    channel_order='HWC',
    color_format='BGR',
    data_type='numpy'
):
    """
    Args:
        img: np.ndarray or bytes
        channel_order(string):
            channel order: HWC or CHW
        color_format(string):
            color format order: RGB or BGR
        threshold(float):
            only return result with confidence larger than threshold
        data_type(string): 仅在图像分割时有意义。 'numpy' or 'string'
            'numpy': 返回已解析的mask
            'string': 返回未解析的mask游程编码

    Returns:
        list

    """

```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中, data\_type为numpy时, 返回图像掩码的二维数组

```
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

## C++ SDK

### 使用说明

模型资源文件默认已经打包在开发者下载的SDK包中。请先将SDK包整体拷贝到具体运行的宿主机设备中，再解压缩编译；

在编译或运行demo程序前执行以下命令：

```
source ${cpp_kit位置路径}/thirdparty/opencv/bin/setupvars.sh
```

如果opencvino预测引擎找不到设备需要执行以下命令：

```
sudo cp ${cpp_kit位置路径}/thirdparty/opencvino/deployment_tools/inference_engine/external/97-myrriad-usbboot.rules
/etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig ````
```

### 使用流程

```
// step 1: 配置运行参数
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num"); // 设置序列号
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread({图片路径});
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame，需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频，需在video_config中开启配置
}
}
```

**运行参数配置** 运行参数的配置通过结构体EdgePredictorConfig完成，其定义如下所示：

```
struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};
```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时部分参数也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

具体支持的运行参数可以参考开发工具包中的头文件。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```

cv::Mat mask为图像掩码的二维数组
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域, 0代表非目标区域

```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码, 解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding, 此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};      // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};            // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};     // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;       // frame存储为视频文件的路径
    bool save_all{false};       // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定`CONTROLLER_KEY_AUTH_MODE`为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

- 1. 开启http服务 http服务的启动参考`demo_serving.cpp`文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

- 2. 请求http服务

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试。



URL中的get参数：

参数	说明	默认值
threshold	阈值过滤， 0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Java请求示例

- http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

其他配置

- 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



### Linux FAQ

1. EasyDL 离线 SDK 与云服务效果不一致，如何处理？我们会逐渐消除这部分差异，如果开发者发现差异较大，可通过[工单](#)、[论坛](#)联系我们协助处理。

2. 硬件出现问题或者出现故障怎么办？软件使用有问题怎么处理？

- 如果持续在静电较多的环境中使用，建议使用防静电锡纸包裹板卡
- 如果硬件无法启动等故障，您可以通过商品页联系供应商处理；其它硬件问题，您可以邮件 [edgeboard-vmx.com](mailto:edgeboard-vmx.com)，我们将在0-2日内处理您的问题。为加快处理进度，您在邮件中，尽量描述清楚问题或者需求细节，避免来回沟通。
- 软件使用问题，请尽量通过[工单](#)、[论坛](#)联系我们协助处理。

3. 运行时报错：NC\_ERROR

```
Can not init Myriad device: NC_ERROR
```

一般是硬件没有插上，请确保lsusb能够找到该硬件。或者等待几秒后再试。

### 快速开始 Windows

1. 安装依赖

将操作系统升级到Windows 10

安装.NET Framework4.5

```
https://www.microsoft.com/zh-CN/download/details.aspx?id=42642
```

Visual C++ Redistributable Packages for Visual Studio 2013

```
https://www.microsoft.com/zh-cn/download/details.aspx?id=40784
```

Visual C++ Redistributable Packages for Visual Studio 2015

```
https://www.microsoft.com/zh-cn/download/details.aspx?id=48145
```

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num



点击“启动服务”，等待数秒即可启动成功，本地服务

默认运行在

```
http://127.0.0.1:24401/
```

其他任何语言只需通过HTTP调用即可。

### 接口调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img).json()
```

C## 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

**返回参数** | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----| | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 | | x1, y1 | float | 0~1 | 物体检测，矩形的左上角坐标（相对长宽的比例值） | | x2, y2 | float | 0~1 | 物体检测，矩形的右下角坐标（相对长宽的比例值） |

关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

Windows FAQ

1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下：  
 .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

2. 服务调用时返回为空，怎么处理？调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <http://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <http://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted? Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 其他问题 如果无法解决，可到论坛发帖：<http://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## 🔗 物体检测Jetson专用SDK集成文档

### 简介

本文档介绍EasyEdge/EasyDL的Jetson SDK的使用方法。Jetson SDK支持的硬件包括Jetson nano，Jetson TX2，Jetson AGX Xavier和Jetson Xavier NX。您可在[AI市场](#)了解Jetson相关系列产品，同时可以在[软硬一体方案](#)了解部署方案。

### 模型支持：

- EasyDL图像：图像分类高精度，图像分类高性能，物体检测高精度，物体检测均衡，物体检测高性能，目标跟踪单标签模型。
- BML：
  - 公开数据集预训练模型：SSD-MobileNetV1，YOLOv3-DarkNet，YOLOv3-MobileNetV1，ResNet50，ResNet101，SE-ResNeXt50，SE-ResNeXt101，MobileNetV2，EfficientNetB0\_small，EfficientNetB4，MobileNetV3\_large\_x1\_0，ResNet18\_vd，SE\_ResNet18\_vd，Xception71。
  - 百度超大规模数据集预训练模型：YOLOv3-DarkNet，MobileNetV3\_large\_x1\_0，ResNet50\_vd，ResNet101\_vd。
- EasyEdge：EasyEdge支持的模型较多，详见[查看模型网络适配硬件](#)。若模型不在此列表，可以尝试使用自定义网络生成端计算组件。

**软件版本支持** 使用EasyDL的Jetson系列SDK需要安装指定版本的JetPack和相关组件。所支持的JetPack版本会随着SDK版本的升级和新版本JetPack的推出而不断的更新。在使用SDK前请务必保证软件版本满足此处声明版本。目前所支持的JetPack版本包括：

- JetPack5.0.2
- JetPack5.0.1
- JetPack4.6
- JetPack4.5
- JetPack4.4 (deprecated，该版本SDK会在未来某个版本移除，请切换至新版本JetPack)
- JetPack4.2.2 (已移除，请切换至新版本JetPack)

安装JetPack时请务必安装对应的组件：

- 使用SDK Manager安装JetPack需要勾选TensorRT、OpenCV、CUDA、cuDNN等选项。
- 使用SD Card Image方式（仅对Jetson Nano和Jetson Xavier NX有效）则无需关心组件问题，默认会全部安装。

**Release Notes** | 时间 | 版本 | 说明 | | --- | --- | --- | | 2022.12.29 | 1.7.2 | 新增支持JetPack5.0.2；缓存机制优化；模型性能优化 | | 2022.07.28 | 1.6.0 | 新增支持JetPack5.0.1，新增目标追踪接入实时流的demo | | 2022.05.18 | 1.5.0 | 部分模型切换格式，max\_batch\_size含义变更，由输入图片数不大于该值变更为等于该值；移除适用于JetPack4.2.2的SDK；示例代码demo\_stream\_inference重构；示例代码移除frame\_buffer，新增更安全高效的safe\_queue | | 2021.12.22 | 1.3.5 | 新增支持JetPack4.6；支持在EasyEdge平台语义分割模型生成开发套件；修复缓存问题；支持自定义缓存路径 | | 2021.10.20 | 1.3.4 | 新增支持JetPack4.5；大幅提升EasyDL有损压缩加速模型的推理速度 | | 2021.06.29 | 1.3.1 | 视频流支持分辨率调整；支持将预测后的视频推流，新增推流demo | | 2021.05.13 | 1.3.0 | 新增视频流接入支持；EasyDL模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.1 | EasyEdge新增一系列模型的支持；性能优化 | | 2021.01.27 | 1.1.0 | EasyDL经典版高性能分类模型升级；

EasyDL经典版检测模型新增均衡选项；

EasyEdge平台新增Jetson系列端计算组件的生成；

问题修复 | | 2020.12.18 | 1.0.0 | 接口升级和一些性能优化 | | 2020.08.11 | 0.5.5 | 部分模型预测速度提升 | | 2020.06.23 | 0.5.4 | 支持JetPack4.4DP，支持EasyDL专业版更多模型 | | 2020.05.15 | 0.5.3 | 专项硬件适配SDK支持Jetson系列 |

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE 含义变更。变更前：预测输入图片数不大于该值均可。变更后：预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理，在输入图片数小于该值时依然可正常运行，但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值，尽可能保持最小。 【版本移除】 适用于JetPack4.4版本的SDK被标记为deprecated，SDK会在未来某个版本移除，建议切换至最新版本JetPack。适用于JetPack4.2.2版本的SDK被移除。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

2021-10-20: 【版本移除】适用于JetPack4.2.2版本的SDK被标记为deprecated，该版本代码已停止更新，SDK会在未来某个版本移除，请切换至新版本JetPack

**快速开始 安装依赖** 本SDK适用于JetPack4.5、JetPack4.6、JetPack5.0系列版本，请务必安装其中之一版本，并使用对应版本的SDK。注意在安装JetPack时，需同时安装CUDA、cuDNN、OpenCV、TensorRT等组件。

如已安装JetPack需要查询相关版本信息，请参考下文中的开发板信息查询与设置。

### 使用序列号激活

首先请在官网获取序列号。



将获取到的序列号填写到demo文件中或以参数形式传入。



默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量设置），需要设置额外的环境变量，指定CONTROLLER\_KEY\_AUTH\_MODE为2，`global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`，实例数鉴权模式下还支持指定license证书更新时间，单位是秒，要求设置为大于20的整数，否则会采用默认的license更新时间，修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

**编译并运行Demo** 模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

编译运行：

```
cd src
mkdir build && cd build
cmake ..
make -j$(nproc)
**make install 为可选，也可将lib所在路径添加为环境变量**
sudo make install
sudo ldconfig
./demo_batch_inference/easyedge_batch_inference {模型RES文件夹} {测试图片路径或仅包含图片的文件夹路径} {序列号}
```

demo运行示例：

```
baidu@nano:~/ljay/easydl/sdk/demo/build$ ./demo_batch_inference/easyedge_batch_inference ../../../../RES/
/ljay/images/mix008.jpeg
2020-08-06 20:56:30,665 INFO [EasyEdge] 548125646864 Compiling model for fast inference, this may take a while (Acceleration)
2020-08-06 20:57:58,427 INFO [EasyEdge] 548125646864 Optimized model saved to:
/home/baidu/.baidu/easyedge/jetson/mcache/24110044320/m_cache, Don't remove it
Results of image /ljay/images/mix008.jpeg:
2, kiwi, p:0.997594 loc: 0.352087, 0.56119, 0.625748, 0.868399
2, kiwi, p:0.993221 loc: 0.45789, 0.0730294, 0.73641, 0.399429
2, kiwi, p:0.992884 loc: 0.156876, 0.0598725, 0.3802, 0.394706
1, tomato, p:0.992125 loc: 0.523592, 0.389156, 0.657738, 0.548069
1, tomato, p:0.991821 loc: 0.665461, 0.419503, 0.805282, 0.573558
1, tomato, p:0.989883 loc: 0.297427, 0.439999, 0.432197, 0.59325
1, tomato, p:0.981654 loc: 0.383444, 0.248203, 0.506606, 0.400926
1, tomato, p:0.971682 loc: 0.183775, 0.556587, 0.286996, 0.711361
1, tomato, p:0.968722 loc: 0.379391, 0.0386965, 0.51672, 0.209681
Done
```

检测结果展示：



### 测试Demo HTTP 服务

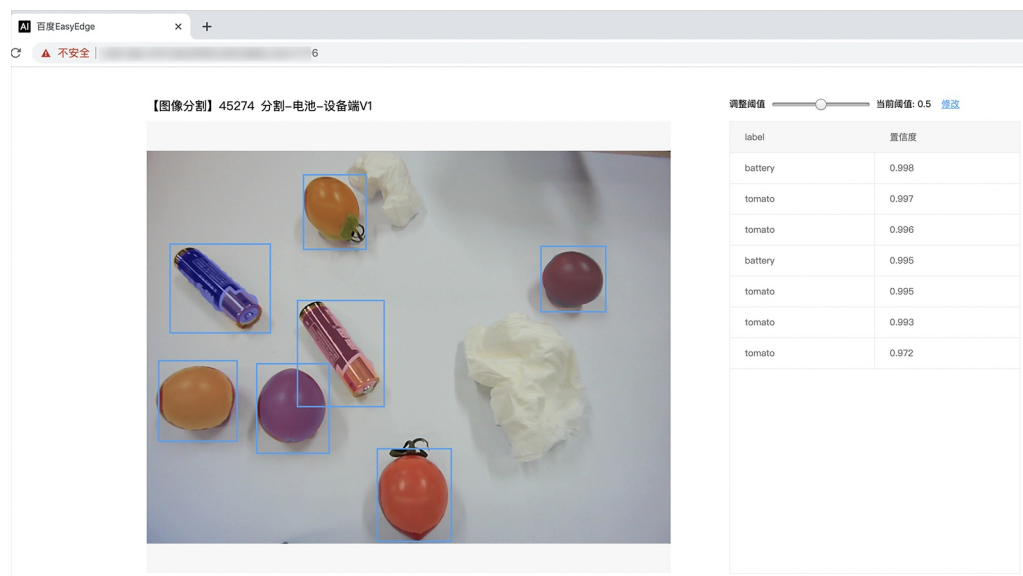
编译demo完成之后，会同时生成一个http服务，运行

```
**. /easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

### 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型运行参数
EdgePredictorConfig config;
config.model_dir = model_dir;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, serial_num);
config.set_config(params::PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE, 1); // 优化的模型可以支持的batch_size
config.set_config(params::PREDICTOR_KEY_GTURBO_FP16, false); // 置true开启fp16模式推理会更快, 精度会略微降低, 但取决于硬件是否支持fp16, 不是所有模型都支持fp16, 参阅文档
config.set_config(params::PREDICTOR_KEY_GTURBO_COMPILE_LEVEL, 1); // 编译模型的策略, 如果当前设置的max_batch_size与历史编译存储的不同, 则重新编译模型

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

### 初始化接口

```

auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

```

若返回非0, 请查看输出日志排查错误原因。

### 预测接口

```
/**
 * @brief
 * 单图预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& results
) = 0;

/**
 * @brief
 * 批量图片预测接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `max_batch_size`，其含义见下方参数配置接口的介绍。

**参数配置接口** 参数配置通过结构体EdgePredictorConfig完成。



```

struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};

```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

针对Jetson开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）

```

```

* 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
* 值类型: int
* 默认值: 1
*/
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产出的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的 GPU 卡号，对于 Jetson，此值无需更改。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制，当此值为 0 时，表示忽略当前设置的 `max_batch_size` 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 `max_batch_size` 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference。

**PREDICTOR\_KEY\_GTURBO\_FP16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持 fp16 的模式包括：EasyDL 图像分类高精度模型。

## 预测视频接口

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

class `VideoDecoding` :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct `VideoConfig`

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};         // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;           // frame存储为视频文件的路径
    bool save_all{false};            // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于`/dev/video0`的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

#### 返回格式

预测成功后，从 `EdgeResultData`中可以获得对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测或图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 图片宽度 = 检测框的左上角的横坐标 y1 图片高度 = 检测框的左上角的纵坐标 x2 图片宽度 = 检测框的右下角的横坐标 y2 图片高度 = 检测框的右下角的纵坐标

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里, 图片的解码运行在cpu之上, 可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量, 根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

### 2. 请求http服务

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片来进行测试。

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供, 则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

python

c#

C++

java

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                      data=img.json())

print(result)
```

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考接口使用-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

**多线程预测** Jetson 系列 SDK 支持多线程预测, 创建一个 predictor, 并通过 PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY 控制所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。需要注意的是多线程的启用会随着线程数的增加而降低单次 infer 的推理速度, 建议优先使用 batch inference 或权衡考虑使用。

#### 已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时, 部分结果错误

A: EasyDL图像分类高精度模型在有些显卡上可能存在此问题, 可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

#### 2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object

A: 如果遇到此问题, 请确认没有频繁调用 init 接口, 通常调用 infer 接口即可满足需求。

### 3. 开启 fp16 后, 预测结果错误

A: 不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括: EasyDL图像分类高精度模型。目前不支持的将会在后面的版本陆续支持。

### 4. 部分模型不支持序列化

A: 针对JetPack4.4、4.5版本, 部分模型无法使用序列化, 如已知的BML的MobileNetV1-SSD和物体检测高性能模型。需要每次加载模型的时候编译模型, 过程会比较慢。此问题将在后续JetPack版本中修复。目前JetPack4.6版本SDK已修复该问题。

**开发板信息查询与设置 查询L4T或JetPack版本** 查询JetPack版本信息, 可以通过下面这条命令先查询L4T的版本。

```
**在终端输入如下命令并回车**
$ head -n 1 /etc/nv_tegra_release
**就会输出类似如下结果**
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看, 板子当前的L4T版本为R32.4.3, 对应JetPack4.4。注意, L4T的版本不是JetPack的版本, 一般可以从L4T的版本唯一对应到JetPack的版本, 下面列出了最近几个版本的对应关系:

```
L4T R32.6.1 --> JetPack4.6
L4T R32.5.1 --> JetPack4.5.1
L4T R32.5 --> JetPack4.5
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下, 执行AI推理的速度是不一样的, 如果对速度需求很高, 可以把功率开到最大, 但记得加上小风扇散热~

```
**1. 运行下面这条命令可以查询开发板当前的运行功率模式**
$ sudo nvpmode -q verbose
**$ NV Power Mode: MAXN**
**$ 0**
**如果输出为MAXN代表是最大功率模式**

**2. 若需要把功率调到最大, 运行下面这条命令**
$ sudo nvpmode -m 0

**如果你进入了桌面系统, 也可以在桌面右上角有个按钮可以切换模式**

**3. 查询资源利用率**
$ sudo tegrastats
```

### FAQ 1. EasyDL SDK与云服务效果不一致, 如何处理?

后续我们会消除这部分差异, 如果开发者发现差异较大, 可联系我们协助处理。

### 2. 运行SDK报错 Authorization failed 日志显示 Http perform failed: null respond 在新的硬件上首次运行, 必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

### 3. 使用libcurl请求http服务时, 速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题, 添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

### 4. 运行demo时报找不到libeasyedge\_extension.so

需要export libeasyedge\_extension.so所在的路径, 如路径为/home/work/baidu/cpp/lib, 则需执行:

```
export LD_LIBRARY_PATH=/home/work/baidu/cpp/lib:${LD_LIBRARY_PATH}
```

或者在编译完后执行如下命令将lib文件安装到系统路径：

```
sudo make install
```

如不能安装，也可手动复制lib下的文件到/usr/local/lib下。

## 5. 运行demo时报如下之一错误

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Compiling model for fast inference, this may take a while (Acceleration)
Killed

**或**

2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Build graph failed
```

请适当降低PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE和PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY的值后尝试。

**6. 运行有损压缩加速的模型，运算精度较标准模型偏低** 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

## 🔗 物体检测辨影专用SDK集成文档

### 简介

本文档介绍EasyEdge/EasyDL的辨影软硬一体方案SDK的使用方法。支持的硬件包括辨影Air、辨影Pro。您可以在[软硬一体方案](#)了解部署方案。

### 模型支持：

- EasyDL图像：图像分类高精度，图像分类高性能，物体检测高精度，物体检测均衡，物体检测高性能
- BML：
  - 公开数据集预训练模型：SSD-MobileNetV1，YOLOv3-DarkNet，YOLOv3-MobileNetV1，ResNet50，ResNet101，SE-ResNeXt50，SE-ResNeXt101，MobileNetV2，EfficientNetB0\_small，EfficientNetB4，MobileNetV3\_large\_x1\_0，ResNet18\_vd，SE\_ResNet18\_vd，Xception71。
  - 百度超大规模数据集预训练模型：YOLOv3-DarkNet，MobileNetV3\_large\_x1\_0，ResNet50\_vd，ResNet101\_vd。
- EasyEdge：EasyEdge支持的模型较多，详见[查看模型网络适配硬件](#)。若模型不在此列表，可以尝试使用自定义网络生成端计算组件。

Release Notes | 时间 | 版本 | 说明 | | --- | --- | --- | | 2022.08.01 | 1.3.5 | 新增支持辨影软硬一体方案部署 |

**辨影软件接入使用SDK** 辨影Air/Pro自带软件预置了大量飞桨开源模型，支持EasyDL/BML模型SDK一键导入使用，详细的辨影使用说明见购买后获得的使用说明书

- 辨影推理主界面





- 辨影设置界面。在应用中可选预置模型能力，也可选择EasyDL/BML导入的模型SDK



快速开始 接下来的文档内容将会描述辨影SDK的集成开发教程，仅需要使用辨影自带软件的用户无需关注

使用序列号激活

首先请在[EasyDL智能云官网](#)获取序列号。

EasyDL图像

算力资源管理

公有云部署

EasyEdge本地部署

- 服务器纯离线服务
- 设备端纯离线服务
- 专项硬件纯离线...

专项硬件适配服务

按单台设备激活 按多台设备激活

使用说明: (展开查看授权方式、激活方法等)

提示: 如您已完成企业资质认证, 将享受该账号下任意部署包免费试用2个月, 更多企业认证权益详见 [企业权益专属礼包](#)

购买永久授权 新增测试序列号 管理序列号 查看转让序列号 批量离线激活

设备名	激活状态	序列号
自定义设备	已过期	8BA2-39AA-73F
自定义设备	未激活	2013-2CD8-45E

将获取到的序列号填写到demo文件中或以参数形式传入。



默认情况下(联网激活或者离线激活的场景), 按照上述说明正确设置序列号即可, 如果是实例数鉴权模式 (请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式, 仅实例数鉴权需要进行下面的变量设置), 需要设置额外的环境变量, 指定CONTROLLER\_KEY\_AUTH\_MODE为2, `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)`, 实例数鉴权模式下还支持指定license证书更新时间, 单位是秒, 要求设置为大于20的整数, 否则会采用默认的license更新时间, 修改实例数鉴权license更新时间的方法参考 `global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)`

编译并运行Demo 模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

编译运行:

```
cd src
mkdir build && cd build
cmake ..
make
**make install 为可选, 也可将lib所在路径添加为环境变量**
sudo make install
sudo ldconfig
./demo_batch_inference/easyedge_batch_inference {模型RES文件夹} {测试图片路径或仅包含图片的文件夹路径} {序列号}
```

demo运行示例:

```
baidu@nano:~/ljay/easydl/sdk/demo/build$ ./demo_batch_inference/easyedge_batch_inference ../../../../RES/
/ljay/images/mix008.jpeg
2020-08-06 20:56:30,665 INFO [EasyEdge] 548125646864 Compiling model for fast inference, this may take a while (Acceleration)
2020-08-06 20:57:58,427 INFO [EasyEdge] 548125646864 Optimized model saved to:
/home/baidu/.baidu/easyedge/jetson/mcache/24110044320/m_cache, Don't remove it
Results of image /ljay/images/mix008.jpeg:
2, kiwi, p:0.997594 loc: 0.352087, 0.56119, 0.625748, 0.868399
2, kiwi, p:0.993221 loc: 0.45789, 0.0730294, 0.73641, 0.399429
2, kiwi, p:0.992884 loc: 0.156876, 0.0598725, 0.3802, 0.394706
1, tomato, p:0.992125 loc: 0.523592, 0.389156, 0.657738, 0.548069
1, tomato, p:0.991821 loc: 0.665461, 0.419503, 0.805282, 0.573558
1, tomato, p:0.989883 loc: 0.297427, 0.439999, 0.432197, 0.59325
1, tomato, p:0.981654 loc: 0.383444, 0.248203, 0.506606, 0.400926
1, tomato, p:0.971682 loc: 0.183775, 0.556587, 0.286996, 0.711361
1, tomato, p:0.968722 loc: 0.379391, 0.0386965, 0.51672, 0.209681
Done
```

检测结果展示:



## 测试Demo HTTP 服务

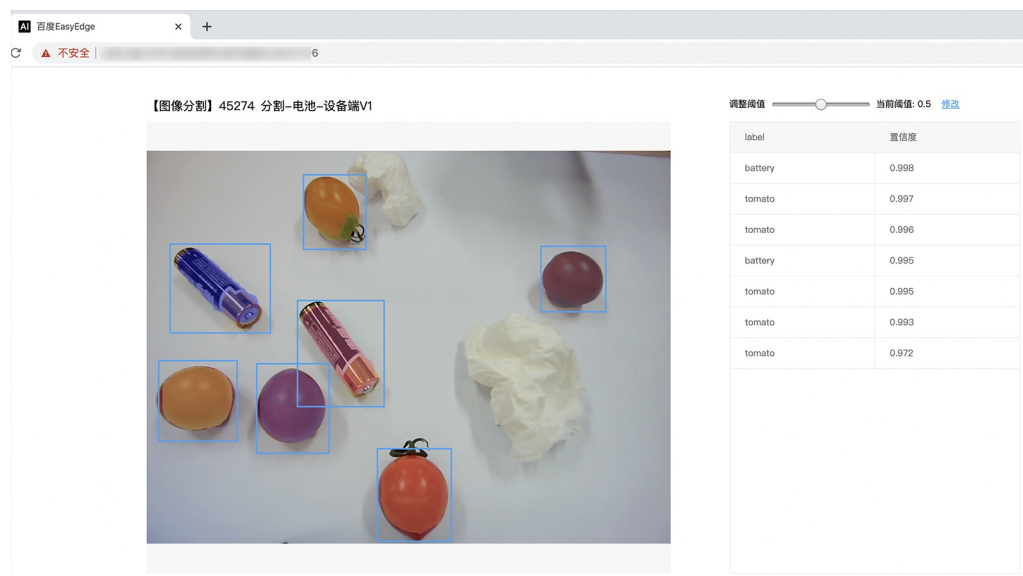
编译demo完成之后，会同时生成一个http服务，运行

```
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型运行参数
EdgePredictorConfig config;
config.model_dir = model_dir;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, serial_num);
config.set_config(params::PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE, 1); // 优化的模型可以支持的最大batch_size，实际单次推理的图片数不能大于此值
config.set_config(params::PREDICTOR_KEY_GTURBO_FP16, false); // 置true开启fp16模式推理会更快，精度会略微降低，但取决于硬件是否支持fp16，不是所有模型都支持fp16，参阅文档
config.set_config(params::PREDICTOR_KEY_GTURBO_COMPILE_LEVEL, 1); // 编译模型的策略，如果当前设置的max_batch_size与历史编译存储的不同，则重新编译模型

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame，需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频，需在video_config中开启配置
}

```

### 初始化接口

```

auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

```

若返回非0，请查看输出日志排查错误原因。

### 预测接口

```
/**
 * @brief
 * 单图预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& results
) = 0;

/**
 * @brief
 * 批量图片预测接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `max_batch_size`，其含义见下方参数配置接口的介绍。

**参数配置接口** 参数配置通过结构体EdgePredictorConfig完成。

```

struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};

```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

针对Jetson开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产出的max_batch_size不相等时，则重新编译模型（推荐）

```

```

* 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
* 值类型: int
* 默认值: 1
*/
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产出的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数不可大于此值，但可以是不大于此值的任意图片数。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 **max\_batch\_size** 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 **compile\_level** 来控制，当此值为 0 时，表示忽略当前设置的 **max\_batch\_size** 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 **max\_batch\_size** 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference。

**PREDICTOR\_KEY\_GTURBO\_FP16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持fp16的模式包括：EasyDL图像分类高精度模型。

## 预测视频接口

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 VideoDecoding，此类提供了获取视频帧数据的便利函数。通过 VideoConfig 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK infer 接口的参数进行预测。

class VideoDecoding :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct VideoConfig



```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};         // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;           // frame存储为视频文件的路径
    bool save_all{false};            // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

#### 返回格式

预测成功后，从 `EdgeResultData`中可以获得对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测或图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 图片宽度 = 检测框的左上角的横坐标 y1 图片高度 = 检测框的左上角的纵坐标 x2 图片宽度 = 检测框的右下角的横坐标 y2 图片高度 = 检测框的右下角的纵坐标

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里, 图片的解码运行在cpu之上, 可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量, 根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

### 2. 请求http服务

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片来进行测试。

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供, 则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Java请求示例参考[这里](#)

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考接口使用-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

**多线程预测** 辨影系列 SDK 支持多线程预测, 创建一个 predictor, 并通过 PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY 控制所支持的最大并发量, 只需要 init 一次, 多线程调用 infer 接口。需要注意的是多线程的启用会随着线程数的增加而降低单次 infer 的推理速度, 建议优先使用 batch inference 或权衡考虑使用。

#### 已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时, 部分结果错误

A: EasyDL图像分类高精度模型在有些显卡上可能存在此问题, 可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

#### 2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object

A: 如果遇到此问题, 请确认没有频繁调用 init 接口, 通常调用 infer 接口即可满足需求。

#### 3. 开启 fp16 后, 预测结果错误

A: 不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括: EasyDL图像分类高精度模型。目前不支持的将会在后面的版本陆续支持。

#### 4. 部分模型不支持序列化

A: 针对JetPack4.4版本, 部分模型无法使用序列化, 如已知的BML的MobileNetV1-SSD和物体检测高性能模型。需要每次加载模型的时候编译模

型，过程会比较慢。此问题将在后续JetPack版本中修复。

**开发板信息查询与设置 查询L4T或JetPack版本** 查询JetPack版本信息，可以通过下面这条命令先查询L4T的版本。

```
**在终端输入如下命令并回车**
$ head -n 1 /etc/nv_tegra_release
**就会输出类似如下结果**
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看，板子当前的L4T版本为R32.4.3，对应JetPack4.4。注意，L4T的版本不是JetPack的版本，一般可以从L4T的版本唯一对应到JetPack的版本，下面列出了最近几个版本的对应关系：

```
L4T R32.6.1 --> JetPack4.6
L4T R32.5.1 --> JetPack4.5.1
L4T R32.5 --> JetPack4.5
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下，执行AI推理的速度是不一样的，如果对速度需求很高，可以把功率开到最大，但记得加上小风扇散热~

```
**1. 运行下面这条命令可以查询开发板当前的运行功率模式**
$ sudo nvpmode -q verbose
**$ NV Power Mode: MAXN**
**$ 0**
**如果输出为MAXN代表是最大功率模式**

**2. 若需要把功率调到最大，运行下面这条命令**
$ sudo nvpmode -m 0

**如果你进入了桌面系统，也可以在桌面右上角有个按钮可以切换模式**

**3. 查询资源利用率**
$ sudo tegrastats
```

#### FAQ 1. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**2. 运行SDK报错 Authorization failed 日志显示 Http perform failed: null respond** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

#### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

#### 4. 运行demo时报找不到libeasyedge\_extension.so

需要export libeasyedge\_extension.so所在的路径，如路径为/home/work/baidu/cpp/lib，则需执行：

```
export LD_LIBRARY_PATH=/home/work/baidu/cpp/lib:${LD_LIBRARY_PATH}
```

或者在编译完后执行如下命令将lib文件安装到系统路径：

```
sudo make install
```

如不能安装，也可手动复制lib下的文件到/usr/local/lib下。

## 5. 运行demo时报如下之一错误

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Compiling model for fast inference, this may take a while (Acceleration)
Killed

**或**

2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Build graph failed
```

请适当降低PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE和PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY的值后尝试。

**6. 运行有损压缩加速的模型，运算精度较标准模型偏低** 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

🔗 浏览器或小程序部署

🔗 浏览器或小程序部署

### 浏览器或小程序部署

**简介** 本文档介绍EasyDL的浏览器/小程序部署SDK的使用方法，

#### SDK支持范围 浏览器部署

PC浏览器: Chrome、Safari、Firefox

手机浏览器: Baidu App、Safari、Chrome、UC and QQ Browser

#### 小程序部署

小程序: 百度小程序、微信小程序

#### 支持的操作系统

系统: MacOS、Windows

**demo文件结构** SDK解压缩之后，目录结构如下

```
--public
| |--model
|   |--model.json
|   |--chunk_n.dat
|--src
| |--components
| |--App.vue
| |--config.json
| |--env.d.ts
| |--label.json
| |--main.ts
| |--modelInfo.json
| |--usePredict.ts
|--index.html
|--package.json
|--README.md
|--tsconfig.json
|--tsconfig.node.json
|--vite.config.ts
|--yarn.lock
```

demo基于vite，其中public/model下的model.json、chunk\_1.dat...chunk\_n.dat为模型文件，src下为业务代码，index.html为入口文件

**快速开始** 依赖node及npm，如果没有node，请前往[node官网](#)下载长期维护版本

安装依赖：npm install

启动项目：npm run dev

启动后控制台输出

```
vite v2.8.4 dev server running at:
> Local: http://localhost:3000/
> Network: use `--host` to expose
```

到浏览器打开 <http://localhost:3000/> 即可体验demo

### 模型预测结果示例 图像分类示例

```
[0.4450492858886719, 0.3961234986782074, 0.0122891990467906, 0.14653800427913666]
```

数组的index为对应的标签，值为置信度

### 物体检测示例

```
[[1, 0.2247152328491211, 0.11200979351997375, 0.07523892819881439, 0.8540866374969482, 0.5503567457199097], [2, 0.1224712328491211, 0.511200979351997375, 0.27523892819881439, 0.8540866374969482, 0.5503567457199097],...]
```

输出结果是一个二维数组，第二维的结果为：[标签，置信度，矩形框x1坐标，矩形框y1坐标，矩形框x2坐标，矩形框y2坐标]

### 浏览器开发

参考src/usePredict文件

```
// 加载推理引擎
import {Runner, env} from '@paddlejs/paddlejs-core';
// 使用webgl计算方案(暂不能使用wasm、webgpu等计算方案)
import '@paddlejs/paddlejs-backend-webgl';
...
// 注册引擎
const runner = new Runner({
  modelPath: '/model',
  keepRatio: config.rescale_mode === 'keep_ratio',
  mean: config.img_mean.reduce((memo, v) => [...memo, +(v / 255).toFixed(3)], [] as number[]),
  std: config.scale.reduce((memo, v) => [...memo, +(1 / 255 / v).toFixed(3)], [] as number[]),
  bgr: config.colorFormat === 'BGR',
  feedShape: {
    fw: config.resize[0],
    fh: config.resize[1]
  }
});
...
// init runner
await runner.init();
...
// predict and get result
await runner.predict(img);
```

更多可参考[PaddleJS工程页](#)

### 小程序开发

#### 微信小程序

微信小程序需添加 [Paddle.js微信小程序插件](#)

步骤：

小程序管理界面 --> 设置 --> 第三方设置 --> 插件管理 --> 添加插件 --> 搜索 wx7138a7bb793608c3 并添加

#### 掌上百度小程序

手百小程序需添加paddlejs百度智能小程序动态库 [引入动态库代码包](#)

代码示例：

```
{
  "dynamicLib": {
    // 定义一个别名，小程序中用这个别名引用动态库。
    "paddlejs": {
      "provider": "paddlejs"
    }
  }
}
```

### 使用动态库

在使用页面的json文件里配置如下信息：

```
{
  "usingSwanComponents": {
    "paddlejs": "dynamicLib://paddlejs/paddlejs"
  }
}
```

从而页面中可以使用此组件：

```
<view class="container">
  <view>下面这个自定义组件来自于动态库</view>
  <paddlejs />
</view>
```

### 示例

index.swan

```
<view class="container">
  <!-- index.wxml -->
  <image style="width:100%; height: 300px; " src="{{imgPath}}"></image>
  <button bindtap="chooseImage">选择图片</button>
  <button bindtap="doPredict" class="btn" type="primary">新鲜度预测</button>
  <!-- 返回结果 -->
  <view class="result" s-if="resultType">预测结果：{{resultType}}</view>
  <view class="result" s-if="resultVal">预测可信度：{{resultVal}}</view>
  <paddlejs options="{{options}}" status="{{status}}" imgBase64="{{imgBase64}}" bindchildmessage="predict" />
</view>
```

index.js

```
Page({
  data: {
    imgPath: '',
    content: '',
    resultType: '',
    resultVal: '',
    isShow: true,
    options: { // 模型配置项
      modelPath: 'http://localhost:3000/model',
      fileCount: 3,
      needPreheat: true,
      feedShape: {
        fw: 224,
        fh: 224
      },
    },
    fetchShape: [1, 7, 1, 1],
    fill: [255, 255, 255, 255],
    scale: 256,
    targetSize: { height: 224, width: 224 },
    mean: [0.485, 0.456, 0.406],
    std: [0.229, 0.224, 0.225]
  },
  status: '' // 初始值为'', 变为'predict'时会触发模型预测
},
```

```

/**
 * 选择图片
 */
chooseImage: function () {
  const me = this;
  this.setData({
    ishow: false
  });
  swan.chooseImage({
    count: 1,
    sizeType: ['original', 'compressed'],
    sourceType: ['album', 'camera'],
    success(res) {
      const path = res.tempFilePaths[0];
      swan.getFileSystemManager().readFile({
        filePath: path,
        encoding: 'base64',
        success: res => {
          me.setData({
            imgBase64: res && res.data,
            imgPath: path
          });
        },
        fail: res => {
          console.log(res);
        }
      });
    }
  });
},
predict(e) {
  const status = e && e.detail && e.detail.status;
  if (status === 'loaded') {
    this.setData({status: 'loaded', isShow: false});
  }
  else if (status === 'complete') {
    const data = e.detail.data;
    const maxItem = this.getMaxItem(data);
    this.setData({status: '', resultType: maps[maxItem.index], resultVal: maxItem.value});
  }
},
doPredict() {
  this.setData({status: 'predict'});
},
getMaxItem(datas = []) {
  let max = Math.max.apply(null, datas);
  let index = datas.indexOf(max);
  return {value: max, index};
},
});

```

### Prop

名称	类型	默认值	是否必选	描述
options	string		是	模型配置项，参考src/usePrdict
imgBase64	string		是	要预测的图像的base64
status	string	"	是	当前状态，status变化触发组件调用相应的api，当status变为predict时，组件会读取imgBase64作为输入的图像，调用模型预测API

[智能边缘控制台-单节点版](#)

[EasyEdge 智能边缘控制台-单节点版 IEC](#)

EasyEdge Intelligent Edge Console（以下简称IEC）是EasyEdge推出的边缘设备管理的本地化方案。可以运行于多种架构、多系统、多类型的终端之上。通过IEC，用户可以方便地在本地进行



- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活，服务管理
- 接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关事件
- 硬件信息的可视化查看

支持的系统+CPU架构包括：

- Windows x86\_64 (Windows 7 ~ Windows 10, 暂不支持Windows 11)
- Linux x86\_64 / arm32 / arm64

支持各类常见的AI加速芯片，包括：

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列
- 其他EasyDL/EasyEdge/BML支持的AI芯片

完整列表可参考[这里](#)

## Release Note

注意：2.0.0之后，默认以系统服务形式安装iec，无法兼容1.x版本的iec

版本号	发布时间	更新说明
2.2.0	2022-10-27	新增onvif/gb28181支持；完善端云通信逻辑
2.0.0	2022-03-22	支持连接中心节点IECC；支持以系统服务安装
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	IEC 第一版！

## 快速开始

从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#)：intel、AMD的64位x86\_64 CPU
- [Linux amd64](#)：intel、AMD的64位x86\_64 CPU
- [Linux arm](#)：树莓派等32位的ARM CPU
- [Linux arm64](#)：RK3399、飞腾等64位的ARM CPU

或者从纯离线服务管理页可下载智能边缘控制台

The screenshot shows the EasyDL web interface. The top navigation bar includes 'EasyDL', '产品介绍', '操作平台', '应用案例', and '使用文档'. The left sidebar lists various features like '物体检测模型', '创建模型', '训练模型', etc. The main content area is titled '纯离线服务' (Pure Offline Service). It contains a section '纯离线服务说明' (Pure Offline Service Introduction) with text explaining that models are deployed locally and can be managed via API or SDK. Below this text, there are three buttons: '发布新服务' (Publish New Service), '智能云控制台' (Smart Cloud Control Console), and '下载智能边缘控制台' (Download Smart Edge Control Console), with a red arrow pointing to the last one. At the bottom, there are tabs for '服务器' (Server), '通用小型设备' (General Small Devices), and '专项适配硬件' (Specialized Hardware). Below these tabs are 'SDK' and 'API' buttons. A footer note states: '此处发布、下载的SDK为未授权SDK，需要前往控制台获取序列号激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。'

您也可以通过先安装多节点版本IECC，通过中心节点来自动连接安装边缘节点。

**Linux 安装** 解压缩之后，目录结构如下

```
0 EasyEdge-IEC-v2.0.0-linux-amd64 > tree .
.
├── easyedge-iec
├── easyedge-iec-setup.sh
├── etc
│   ├── easyedge-iec.service-conf.init.d
│   ├── easyedge-iec.service-conf.systemd
│   ├── easyedge-iec.service-conf.upstart
│   ├── easyedge-iec.service.yml
│   └── easyedge-iec.yml
└── readme.txt

1 directory, 8 files
```

**以系统服务形式安装（推荐）** 以root用户运行./easyedge-iec-setup.sh install 即可

```
[setup]: sudo could not be found
[setup]: Start to install IEC...
[setup]: + bash -c "cp easyedge-iec /usr/sbin/easyedge-iec"
[setup]: + bash -c "chmod +x /usr/sbin/easyedge-iec"
[setup]: + bash -c "cp etc/easyedge-iec.service.yml /etc/easyedge-iec/easyedge-iec.yml"
[setup]: + bash -c "cp etc/easyedge-iec.service-conf.init.d /etc/init.d/easyedge-iec"
[setup]: + bash -c "chmod +x /etc/init.d/easyedge-iec"
[setup]: Install IEC success!
[setup]: + bash -c "service easyedge-iec start"
Starting easyedge-iec: success
[setup]: Start to check IEC status...
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: IEC status: OK!
[easyedge-iec]: default configure file: /etc/easyedge-iec/easyedge-iec.yml
[easyedge-iec]: default log file: /var/log/easyedge-iec/easyedge-iec.log
[easyedge-iec]: service usage: service easyedge-iec { start | stop }
[setup]: Done!
```

- 日志：/var/log/easyedge-iec/easyedge-iec.log
- 系统配置：/etc/easyedge-iec/easyedge-iec.yml
- 服务启动/停止：service easyedge-iec { start | stop } (不同操作系统内可能不同，具体命令参考安装日志)

**自定义安装（不推荐）** 自定义安装方法仅限于 安装脚本无法识别的情况。

- 拷贝 ./EasyEdge-IEC-v2.0.0/ 整个目录至自定义文件夹，如 /opt/EasyEdge-IEC
- 进入到 /opt/EasyEdge-IEC
- 通过 nohup 等方法运行 ./easyedge-iec-linux-*{您的系统架构}* amd64: intel、AMD的64位x86\_64 CPU arm：树莓派等32位的ARM CPU \* arm64：RK3399、飞腾等64位的ARM CPU
- 日志：./log/easyedge-iec.log
- 系统配置：./easyedge-iec.yml

**Windows 安装**

解压缩之后，安装目录如下所示：

```

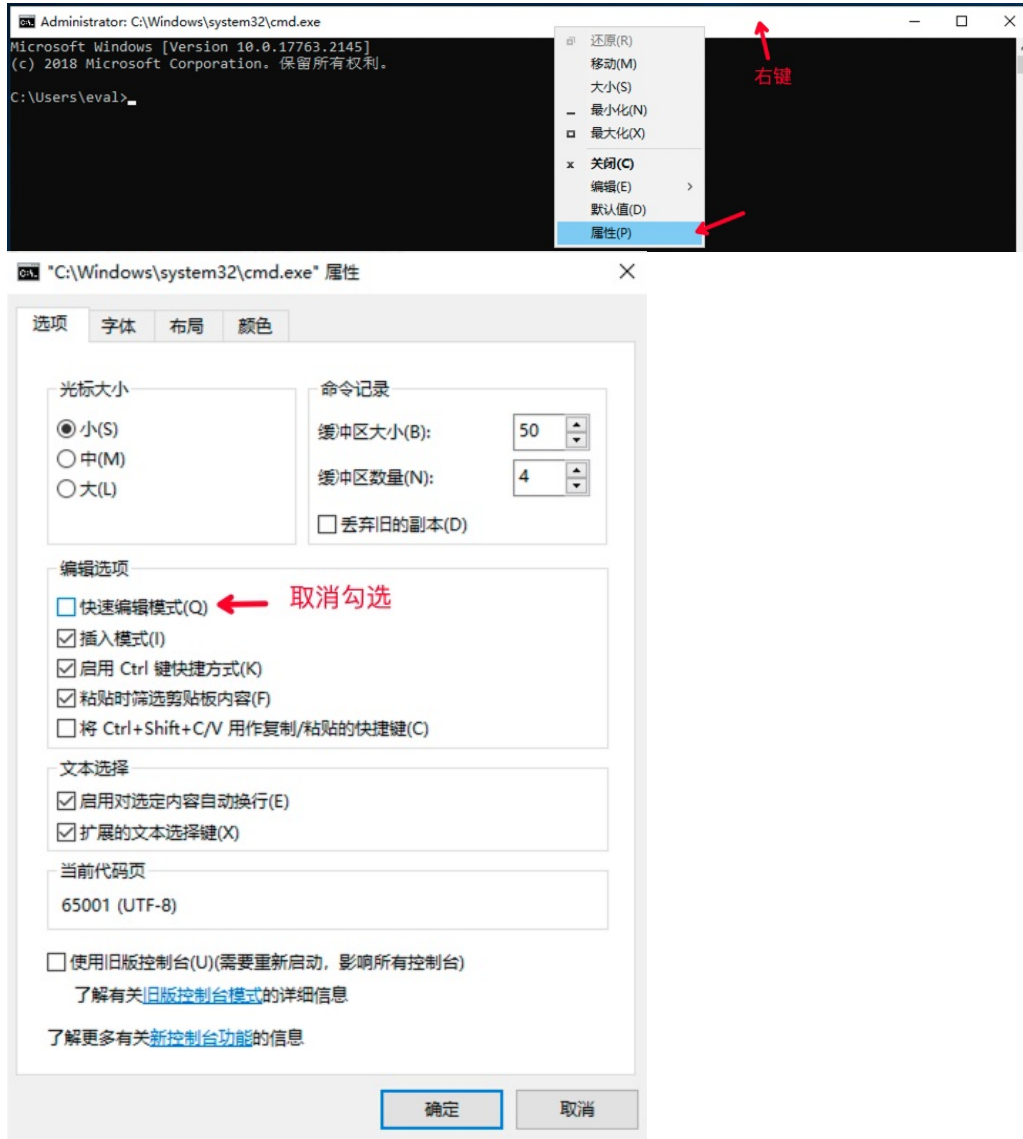
0 tmp2 > tree EasyEdge-IEC-v2.0.0-windows-amd64
EasyEdge-IEC-v2.0.0-windows-amd64
├── easyedge-iec.exe
├── easyedge-iec-setup.bat
├── etc
├── easyedge-iec.yml
└── readme.txt

1 directory, 4 files

```

打开命令行（非powershell）运行 `easyedge-iec-setup.bat install`。

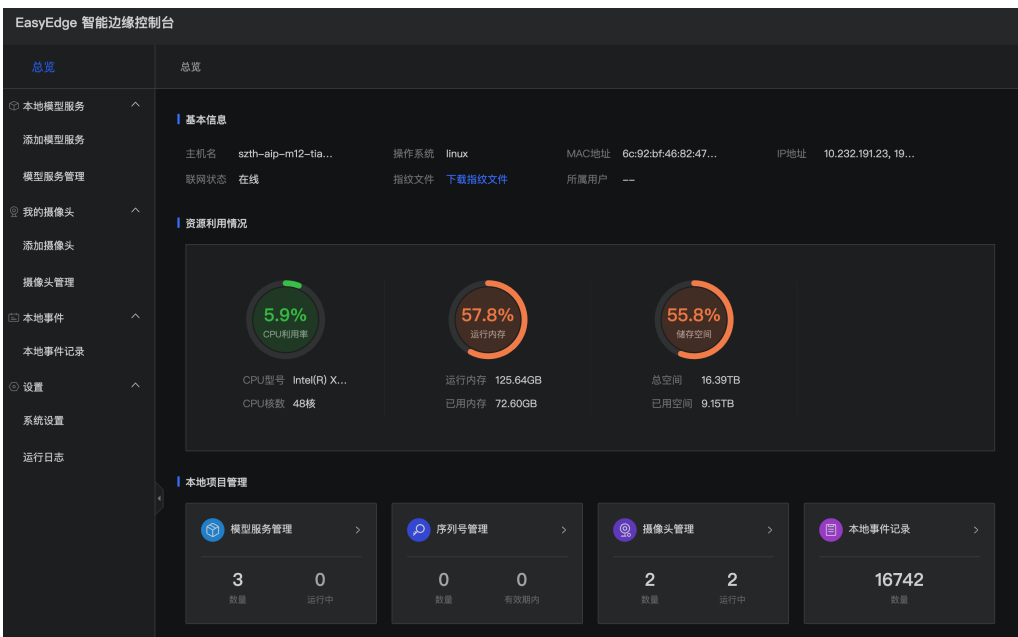
如果遇到hang住的情况，可修改命令行配置 启动之后，打开浏览器，访问 `http://{设备ip}:8702/easyedge/iec` 即可：



启动之后，打开浏览器，访问 `http://{设备ip}:8702/easyedge` 即可：

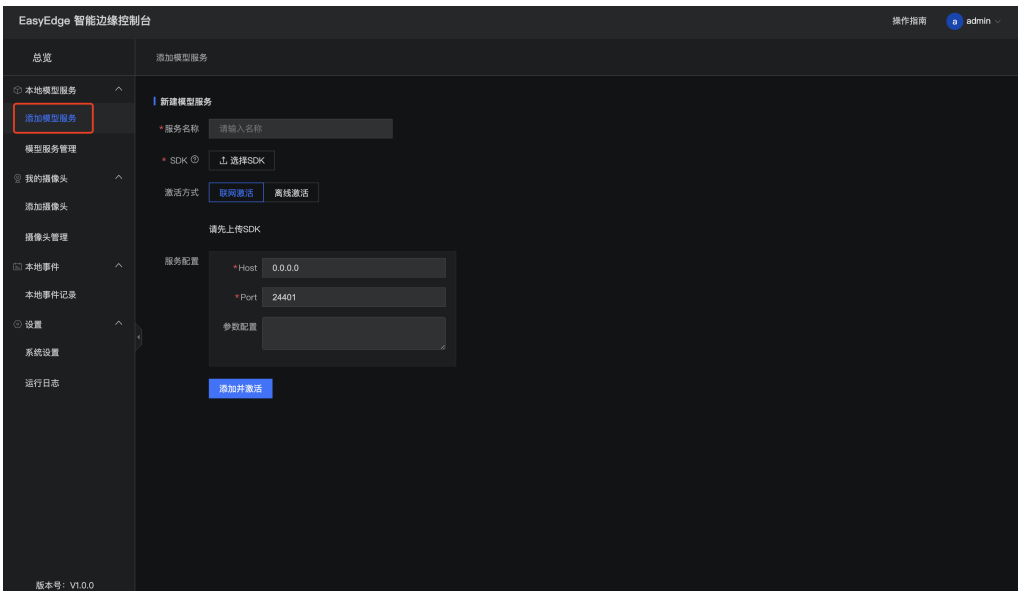


默认用户名密码为 admin / easyedge



### 功能使用说明

①**添加模型服务** 首先，点击导航栏的「本地模型服务」-「添加模型服务」。在页面中定义服务名称后，将已经下载好的Linux/Windows版本的SDK与IEC关联。关联完毕后可按两种激活方式，激活使用SDK。



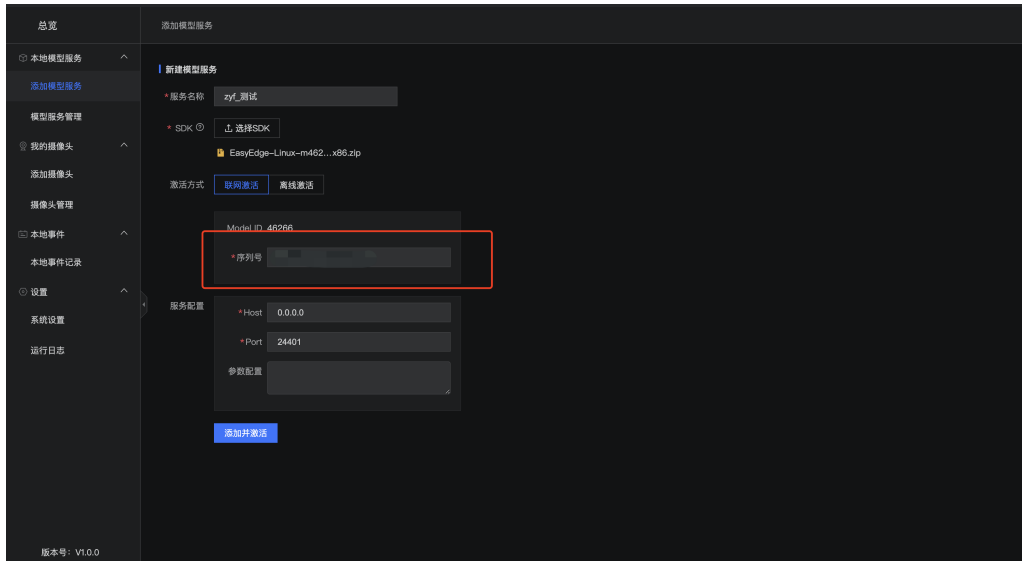
部分SDK需要提前安装系统依赖，如TRT等，具体请参考EasyDL/BML/EasyEdge SDK使用文件中的环境依赖安装说明

## 联网激活

1. 在关联SDK完成后，需要在百度智能云控制台对应部署方式管理页中新增测试序列号或购买正式序列号。（图中以服务器版SDK为例）



2. 再在IEC中填入所申请的序列号



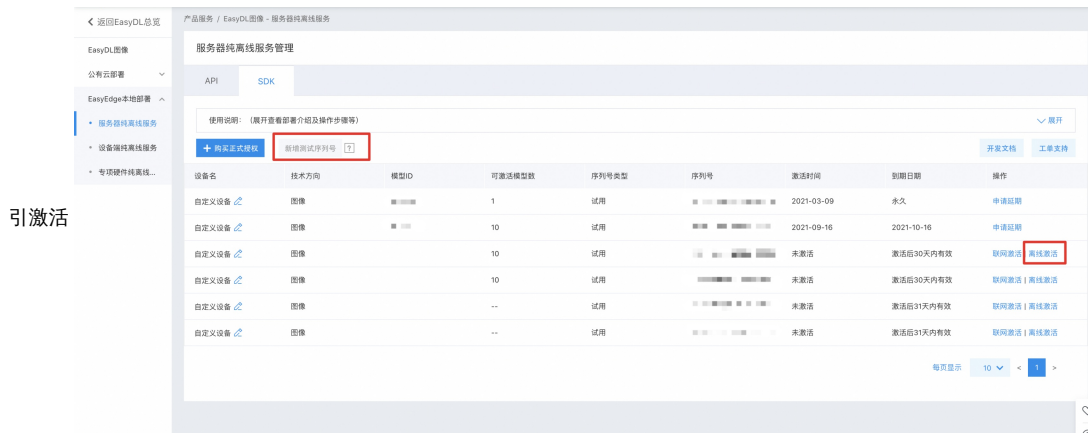
3. 配置服务，在服务端口不冲突占用的情况下，使用默认即可
4. 添加并激活

## 离线激活

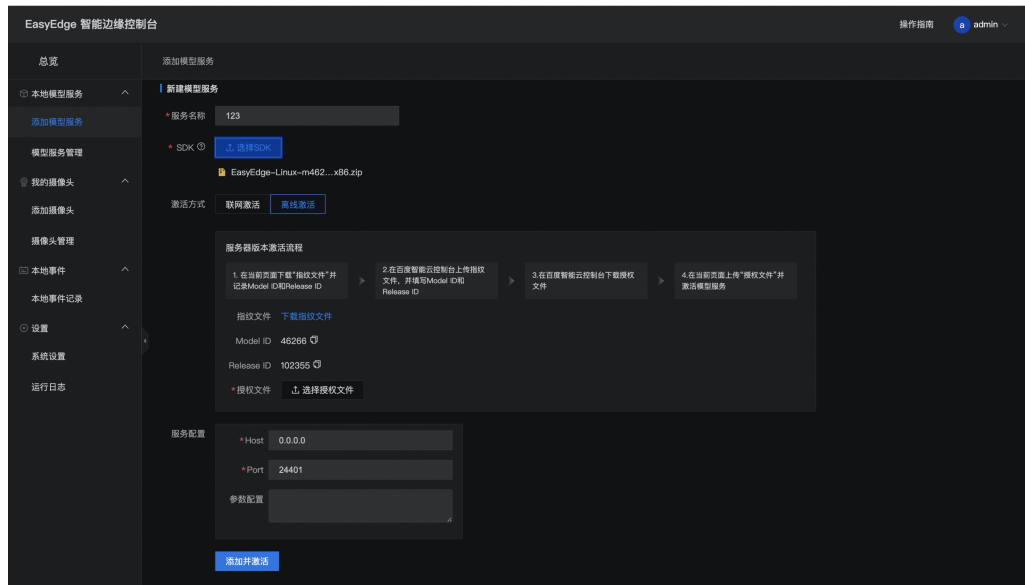
1. 在IEC总览页面下载「指纹文件」



2. 在百度智能云的**控制台**中找到SDK对应的管理列表，图中以服务器SDK为例。申请序列号后，点击对应序列号尾部的「离线激活」操作，按指



3. 在IEC的添加模型服务页面，上传下载好的授权文件，完成激活

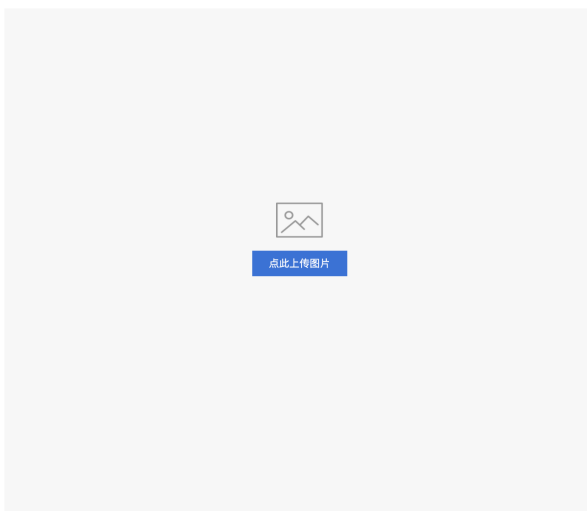


激活完成后即可在「模型服务管理」列表中启动服务，使用后续的操作栏功能。

### 体验本地demo

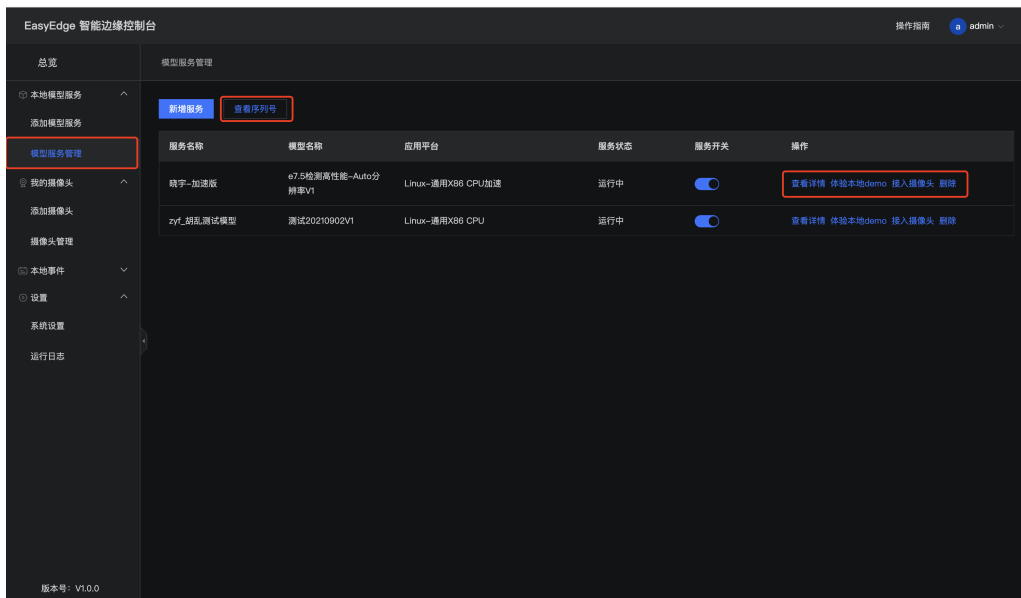
点击「本地demo体验」即可在立即上传图片进行预测

【物体检测】97741 e7.5检测高性能-Auto分辨率V1



### 接入摄像头

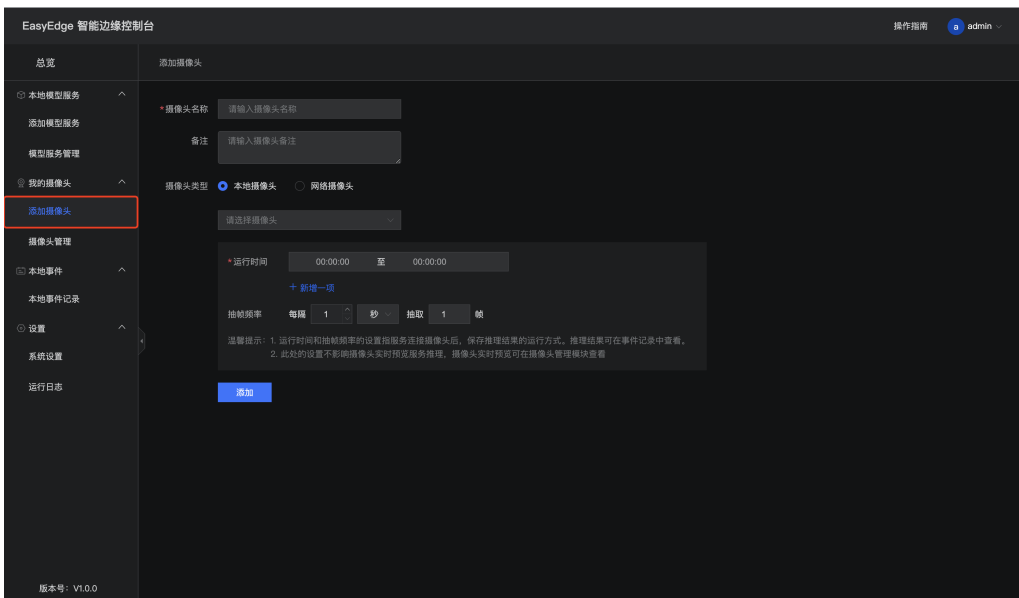
使用接入摄像头功能首先需要添加摄像头，请参考第②步，完成后按照第③步操作 注：服务启动后也可参考「模型发布」模块的技术文档进行开发使用，本文档主要介绍IEC使用功能



## 激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

②添加摄像头 导航栏点击「我的摄像头」-「添加摄像头」，定义摄像头名称、备注后即可添加摄像头。支持本地摄像头和网络摄像头。摄像头添加成功后即可设置摄像头的运行时间和频率



③摄像头接入模型服务预测 点击「本地模型服务」-「模型服务管理」中，所需接入预测的服务的「接入摄像头」



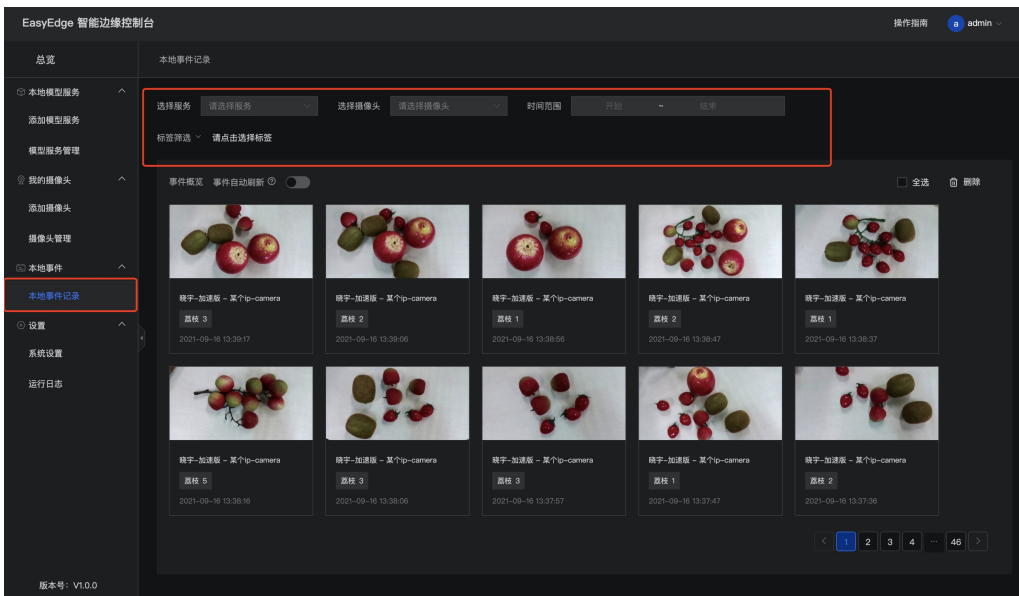
在弹出的弹窗中选择第②

步中添加的摄像头，此时点击确认即可在「摄像头管理」中的实时预览功能中查看摄像头预测结果，识别结果默认不保存。如需保存识别结果，可设置对应的「本地事件触发条件」，根据标签和置信度，将识别结果保存至本地事件记录当中。设置多个标签条件时，IEC会以“或”的逻辑来将所有满足条件的识别结果保存



④本地事件 点击导航栏「本地事件记录」，可通过服务名称、摄像头名称、事件记录的时间、标签及置信度来筛选识别结果查看，多个标签及置信度同样也是“或”的逻辑记录。如有想要删除的事件数据可选择后删除，全选为本页全选。





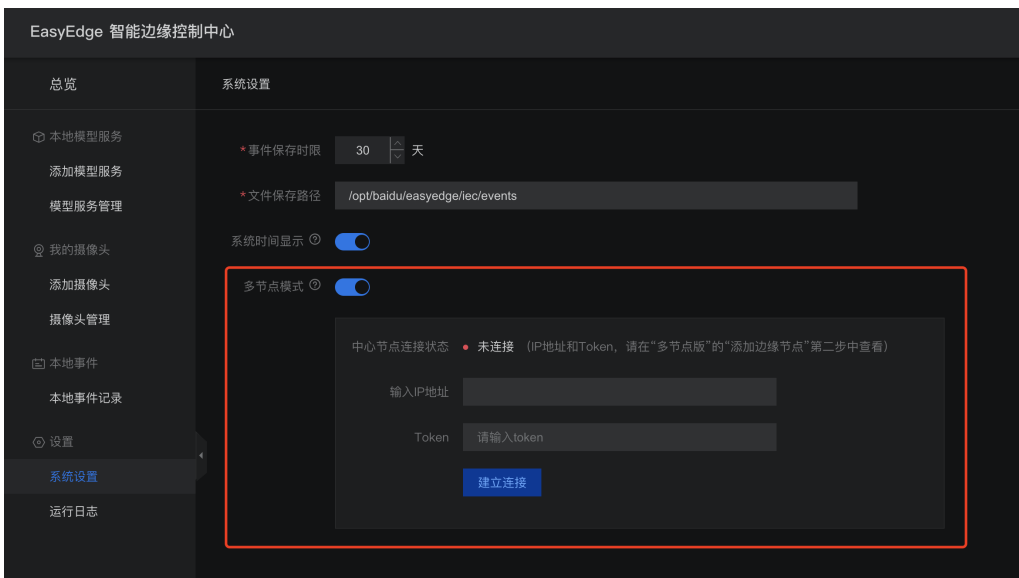
### ⑤ 连接到智能边缘控制台-多节点版 (IECC)

与中心节点连接之后，边缘节点主程序版本会自动随控制中心版本升级。（>2.0.0）

- Step 1 在IECC中添加边缘节点，选择「边缘节点已安装IEC」，并记录IP地址与Token



- Step 2 在IEC的系统设置中打开多节点模式，并填入刚才记录的IP地址与Token，点击建立连接



- 连接完成后即可在中心节点IECC去监控/管理/应用在边缘节点上的IEC

**配置项\***

配置文件etc/easyedge-iec.yml中有关于IEC的各项配置说明，一般无需修改，请确保理解配置项含义之后，再做修改。

```
##### IEC系统配置
##### ----- 高级配置一般无需修改 -----
##### !!!注意!!! 请确保理解配置项含义后再做修改
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
# role: edge
# 硬件利用率刷新时间间隔：过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# 事件监测触发扫描周期
eventTriggerIntervalSecond: 10
# IEC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: default
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge）
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
toFile: yes
loggingFile: /var/log/easyedge-iec/easyedge-iec.log
# 0:info; -1:debug; -2:verbose
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8702
listenHost: 0.0.0.0

sdk:
# GPU SDK所使用的cuda版本：9 / 10 / 10.2 / 11.0 / 11.1。请安装完cuda之后，这设置正确的版本号。
cudaVersion: 10.2
# AI服务启动时，额外配置的 LD_LIBRARY_PATH(linux) 或者 PATH(windows)
libPath: ./
# AI服务启动时，额外配置的其他环境变量。
ENVs:
EDGE_CONTROLLER_KEY_LOG_BRAND: EasyEdge
##### EDGE_CONTROLLER_KEY_XXX: XXXX

commu:
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2

##### 数据库相关配置
db:
sqliteDbFile: /var/lib/easyedge-iec/easyedge-iec.db
hubDbFile: /var/lib/easyedge-iec/easyedge-iec.hub.db
eventDbFile: /var/lib/easyedge-iec/easyedge-event.db
fileServerDbFile: /var/lib/easyedge-iec/easyedge-fileserver.hub.db
nodeMonitorDbFile: /var/lib/easyedge-iec/easyedge-nodemonitor.hub.db

##### 推流相关配置
mediaserver:
flvPort: 8715
rtmpPort: 8716

##### 视频流相关配置
edgestream:
logLevel: -1
listenHost: 127.0.0.1
```

```

listenPort: 8710
# 摄像头预览：识别结果绘制延迟消失
renderExtendFrames: 10
# 预测队列大小：如果设置为60，当摄像头fps=30时，视频延迟约为2秒。降低inferenceQueueSize可以降低预览延迟，但是根据硬件的算力情况，可能导致模型推理速度跟不上，没有识别结果，不建议设置太低
inferenceQueueSize: 60
videoEncodeBitRate: 400000
# 视频采样 & 视频实时预览分辨率设置
# 0: auto, 1: 1080p, 2: 720p, 3: 480p, 4: 360p, 5: 240p
resolution: 0
# 内置多媒体服务配置
# port设为0表示关闭
mediaServerHost: 127.0.0.1
mediaServerFlvPort: 8713
mediaServerRtmpPort: 8714
mediaServerRtspPort: 0

```

**FAQ 启动服务后，进程中出现两个easyedge-iec进程** 这是正常现象，IEC通过守护进程的方式来完成更新等操作。

**启动服务时，显示端口被占用port already been used** 通过修改 easyedge-iec.yml文件的配置后，再重新启动服务。

**安装服务时，报错permission denied** 请以管理员身份运行安装程序。

**中心节点重启后，边缘节点IEC一直离线** 中心节点短时间的离线，边缘节点会自动重连。如果中心节点已经恢复在线，边缘节点长时间未自动连接上，可通过边缘节点iec的方法来重新连接（右上角 admin - 重启系统）

**IEC 是否有Android / iOS 版本** 我们将会在近期发布对Android操作系统的支持

**添加SDK时，报错 SDK不支持该硬件。SDK not supported by this device** 一般是因为使用的SDK跟硬件不匹配，如 GPU的SDK，硬件没有GPU卡。对于Jetson，也可能是Jetpack版本不支持，可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表（cpp文件中的文件名来查看）来匹配。

☞ 智能边缘控制台-多节点版

☞ EasyEdge 智能边缘控制台——多节点版

## 整体介绍

智能边缘控制台 - 多节点版（EasyEdge Intelligent EdgeConsole Center 以下简称IECC），是EasyEdge推出的边缘资源管理、服务应用与管理一站式本地化方案。

通过IECC，用户可以方便地在中心节点管理子节点：

- 边缘硬件资源的管理与监控
- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活，服务管理
- 视频流解析，接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关视频流推理事件

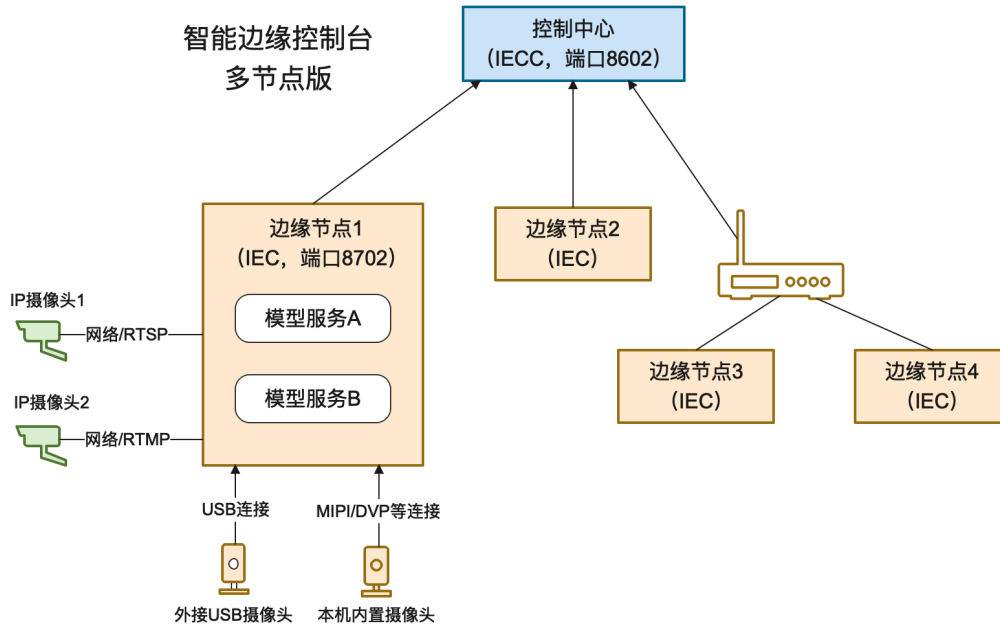
**支持的系统+CPU架构包括：**

- Windows x86\_64 (Windows 7 ~ Windows 10，暂不支持Windows 11)
- Linux x86\_64 / arm32 / arm64

**支持各类常见的AI加速芯片，包括：**

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列
- 其他EasyDL/EasyEdge/BML支持的AI芯片

连接说明 以下为 中心节点（控制中心）,边缘节点/子节点,摄像头的连接示意：



其中：

- 控制中心需要有固定IP,而边缘节点可以处于多级子网之下,只需IEC能够主动访问到控制中心节点即可
- 模型服务均运行于各边缘节点之上
- 摄像头均与边缘节点相连

Release Note

版本号	发布时间	更新说明
2.2.0	2022-10-27	边缘节点新增Android支持；新增onvif/gb28181支持；优化端云通信通道安全
2.0.0	2022-03-25	多节点版上线！
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	智能边缘控制台 - 单节点版 IEC 第一版！

安装 从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#) : intel、AMD的64位x86\_84 CPU
- [Linux amd64](#) : intel、AMD的64位x86\_84 CPU
- [Linux arm](#) : 树莓派等32位的ARM CPU
- [Linux arm64](#) : RK3399、飞腾等64位的ARM CPU

或者从纯离线服务管理页可下载智能边缘控制台



以Linux为例,解压缩后目录结构如下所示：

```
./EasyEdge-IECC-v{版本号}/
|-- easyedge-iecc
|-- easyedge-iecc-setup.sh
|-- etc/
|-- etc/easyedge-iecc.yml
|-- readme.txt
```

## Linux 系统

### 通过系统服务形式安装（推荐）

以管理员运行 `bash easyedge-iecc-setup.sh install` 即可。

```
0 EasyEdge-IEC-v2.0.0 > bash ./easyedge-iecc-setup.sh install
[setup]: sudo could not be found
[setup]: Start to install IECC...
[setup]: + bash -c "cp easyedge-iecc-linux-amd64 /usr/sbin/easyedge-iecc"
[setup]: + bash -c "chmod +x /usr/sbin/easyedge-iecc"
[setup]: + bash -c "cp easyedge-iecc-* /var/lib/easyedge-iecc/fs/tmp"
[setup]: + bash -c "cp etc/easyedge-iecc.service.yml /etc/easyedge-iecc/easyedge-iecc.yml"
[setup]: + bash -c "cp etc/easyedge-iecc.service-conf.init.d /etc/init.d/easyedge-iecc"
[setup]: + bash -c "chmod +x /etc/init.d/easyedge-iecc"
[setup]: Install IECC success!
[setup]: + bash -c "service easyedge-iecc start"
Starting easyedge-iecc: success
[setup]: Start to check IECC status...
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: IECC status: OK!
[easyedge-iecc]: default configure file: /etc/easyedge-iecc/easyedge-iecc.yml
[easyedge-iecc]: default log file: /var/log/easyedge-iecc/easyedge-iecc.log
[easyedge-iecc]: service usage: service easyedge-iecc { start | stop }
[setup]: Done!
```

出现success字样，表示安装成功。

- 日志：`/var/log/easyedge-iecc/easyedge-iecc.log`
- 系统配置：`/etc/easyedge-iecc/easyedge-iecc.yml`
- 服务启动/停止：`service easyedge-iecc { start | stop }` (不同操作系统内可能不同，具体命令参考安装日志)
- 配置服务自启动：可根据不同操作系统参考[这里](#)进行对应配置

可通过 `bash easyedge-iecc-setup.sh uninstall` 来卸载，以及 `bash easyedge-iecc-setup.sh upgrade` 来升级为当前安装包的版本

### 自定义安装（不推荐）

自定义安装仅限于 安装脚本无法识别您的操作系统的情况。

- 拷贝 `./EasyEdge-IEC-v2.0.0/` 整个目录至自定义文件夹，如 `/opt/EasyEdge-IEC`
- 进入到 `/opt/EasyEdge-IEC`
- 通过 `nohup` 等方法运行 `./easyedge-iecc-linux-{您的系统架构} --com.role=hub amd64: intel、AMD的64位x86_64 CPU arm：树莓派等32位的ARM CPU * arm64：RK3399、飞腾等64位的ARM CPU`
- 日志：`./log/easyedge-iecc.log`
- 系统配置：`./easyedge-iecc.yml`

**Windows 系统** 打开命令行（非powershell）运行 `easyedge-iecc-setup.bat install`。

注：如果遇到hang住的情况，可修改命令行配置



验证安装：启动之后，打开浏览器，访问 <http://{设备ip}:8602/easyedge> 即可：



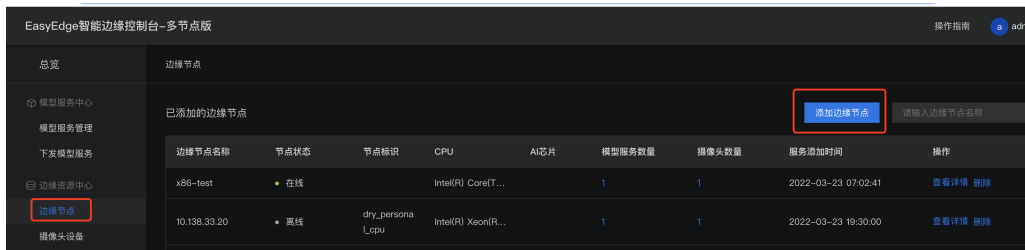
更新服务：关闭服务，下载最新的安装包，重新执行安装流程即可。

注：1. 中心节点更新到新版之后，已连接的边缘节点会自动跟随中心节点，自我升级到同样的版本。

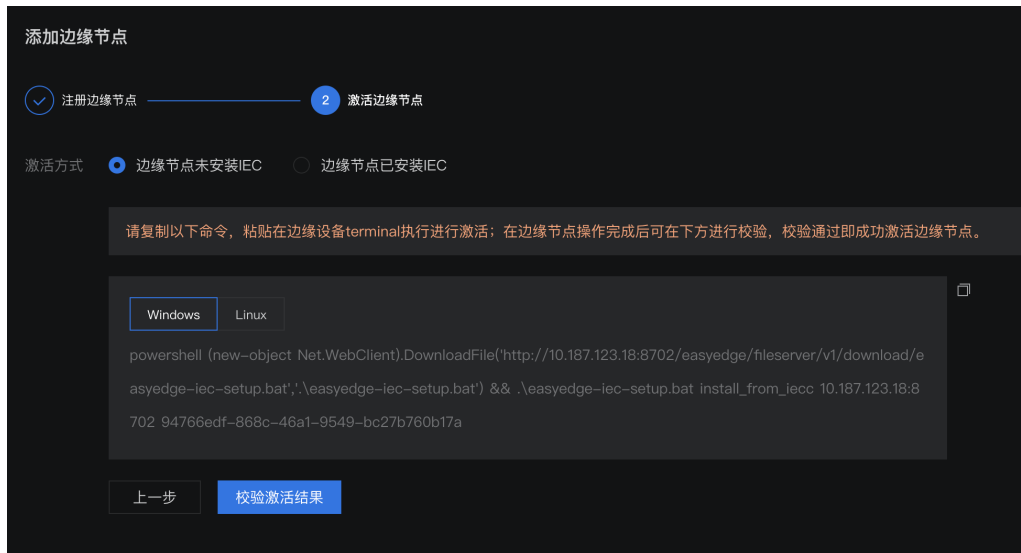
2. 报错: Text file busy. 一般是因为服务没有停止。

### 使用流程 Step 1 注册并激活边缘节点

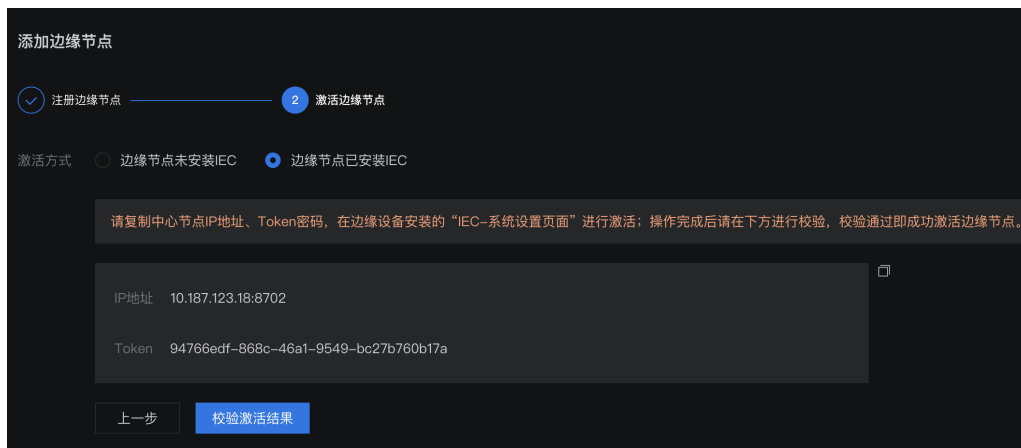
- 在IECC导航栏中点击边缘节点，点击页面中的添加边缘节点按钮



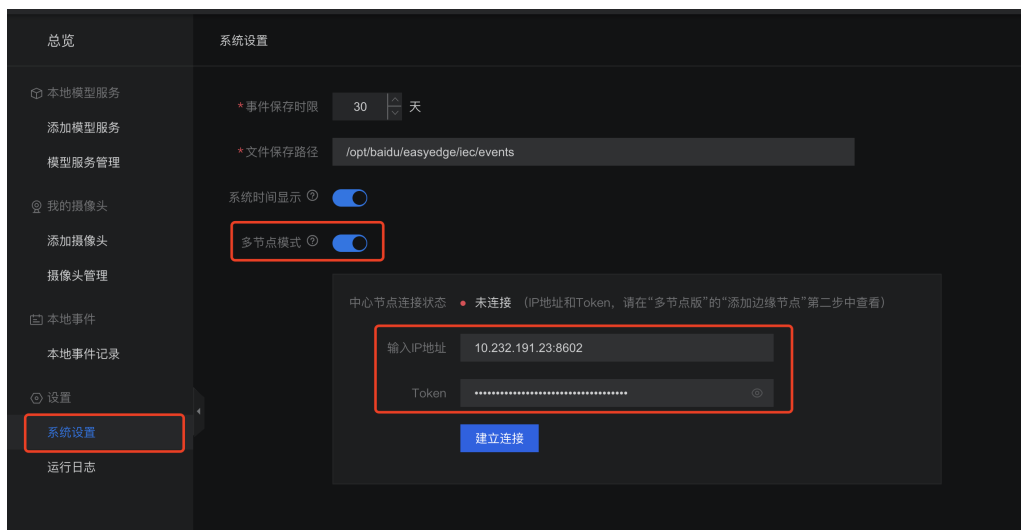
- 注册边缘节点，填写基本信息
- 激活边缘节点，根据边缘节点上是否安装智能边缘控制台-单节点版（IEC）分两种激活方式
  - 边缘节点未安装IEC：复制提供的命令，在边缘节点的终端中输入执行（命令会自动在当前目录，下载单节点版IEC并注册到控制中心）。终端命令执行完成后，在下方校验激活结果，如结果通过即可完成边缘节点的激活



- 边缘节点已安装IEC：记录页面中提供的IP地址和Token



- 在边缘节点的IEC-系统设置中，打开多节点模式开关，将刚才记录的IP地址和Token填入其中，建立连接





- 成功激活后可在边缘节点页面中看到一行状态为**在线**的记录

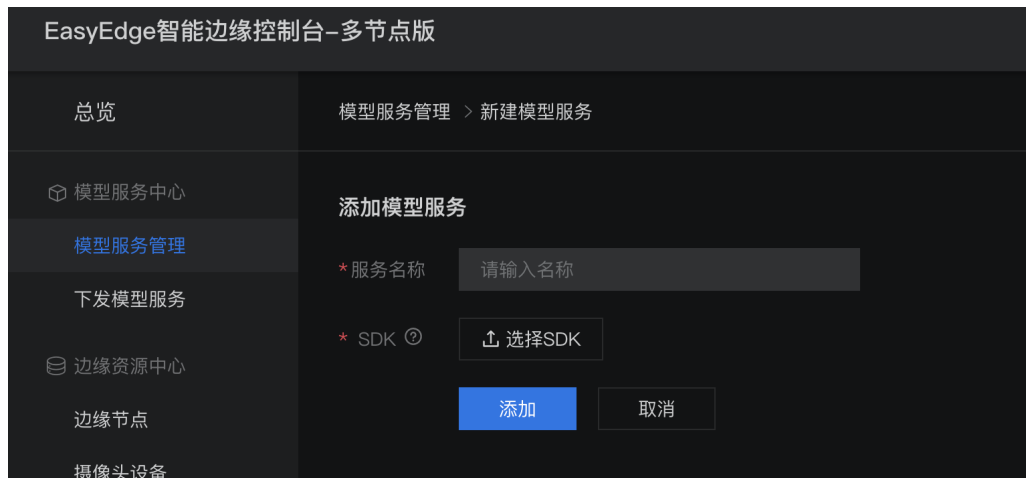


## Step 2 上传并下发模型服务

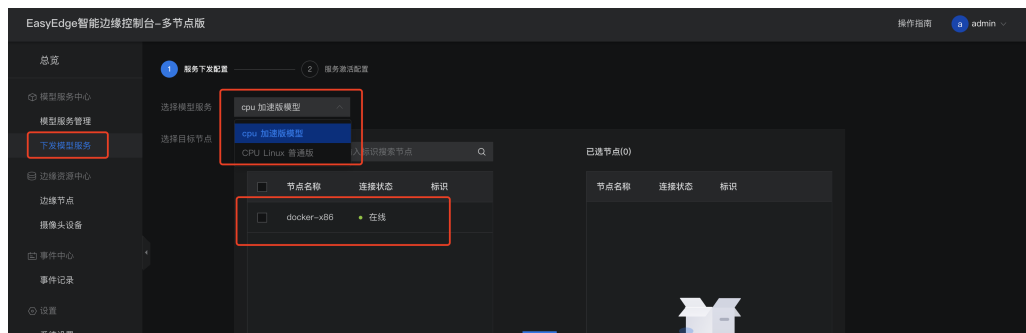
- 在模型服务管理-已添加的模型服务页面中点击**添加模型服务**



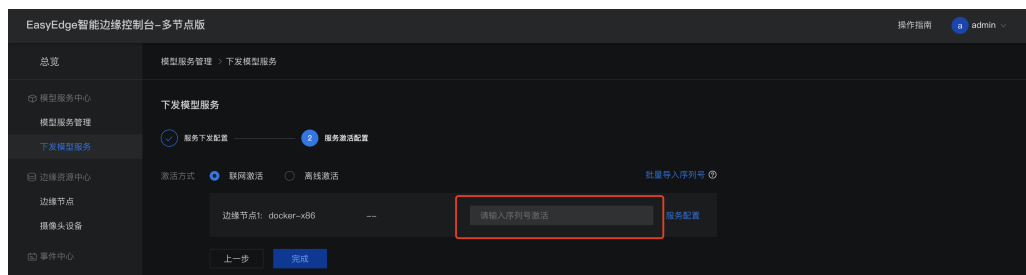
- 上传来自于EasyDL/BML的SDK，目前仅支持Windows/Linux的SDK



- 添加成功后可在已添加的模型服务页面查看添加的模型服务SDK
- 在模型服务SDK上传成功以及边缘节点也添加激活过后，即可将模型服务下发至边缘。点击导航栏-下发模型服务，选择已添加的模型服务，选择下发的目标节点（支持多节点批量下发）进行模型服务下发



- 确定下发配置后，填入模型服务在边缘节点联网激活运行的序列号（支持批量导入）即可完成模型服务下发，序列号可在[智能云控制台](#)获取。离线激活的过程可参考IECC中的具体指引



- 完成上述流程后即可在模型服务管理-已下发的模型服务列表中查看记录，并进行下一步应用功能体验

注：完成此步骤后即可在边缘节点进行二次集成已下发的模型服务，具体的集成方式可在文档-某图像任务类型-模型发布中查找对应的SDK开发文档进行集成开发



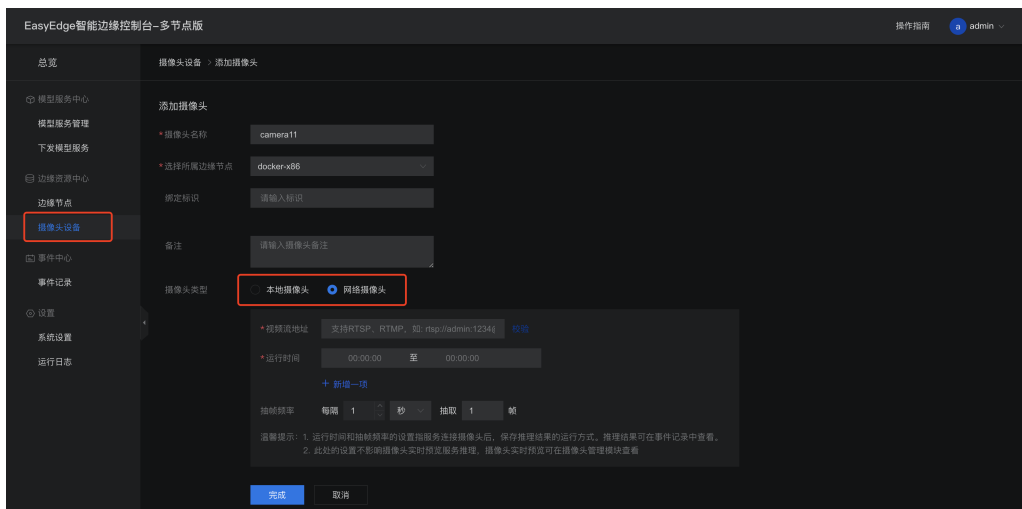


下发时可以通过高级配置设置服务运行的host和port。若不设置，默认host为0.0.0.0，port为系统随机分配的可用端口

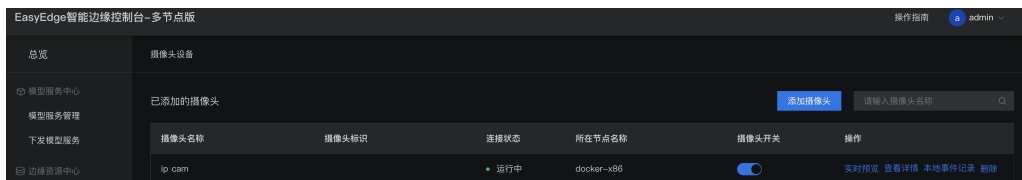
### Step 3 配置摄像头

Step 3 - 5 描述的是如何使用IECC可视化进行视频流式推理与应用，对此有需求的用户建议详细查看后续步骤内容。如仅需对下发的模型服务进行二次集成的用户无需进行后续操作，参考SDK对应的开发文档进行集成即可

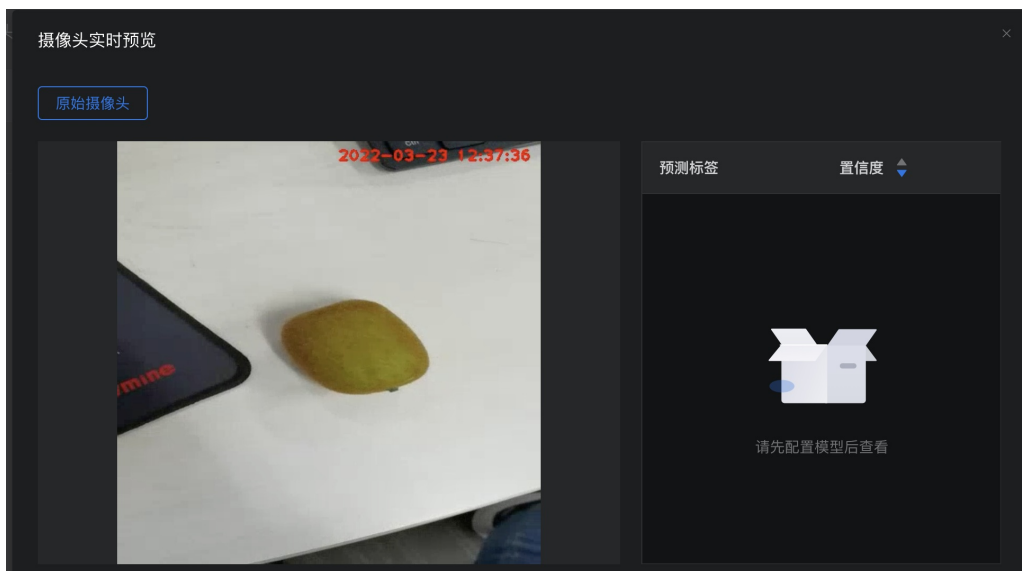
- 首先需要确定边缘节点已经接入物理摄像头，可通过USB插口接入，也可通过RTSP/RTMP流式协议接入。在摄像头设备页面点击添加摄像头按钮，填写对应的信息添加摄像头。支持设置摄像头的运行时间以及摄像头的抽帧频率



- 添加完成后可在摄像头设备页面查看记录

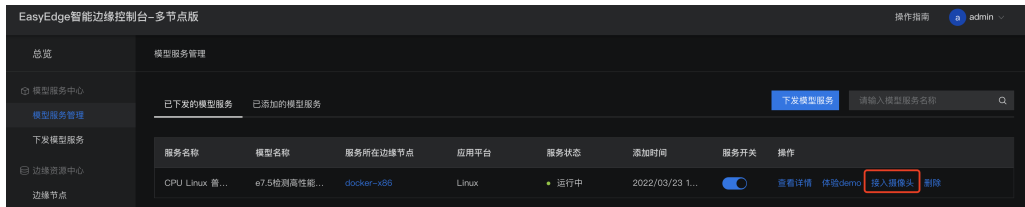


- 点击预览可查看摄像头预览画面

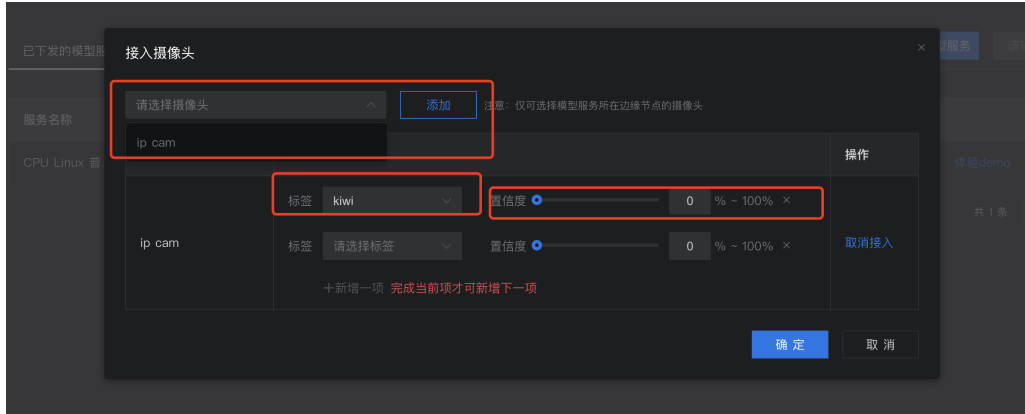


### Step 4 模型服务接入视频流预测

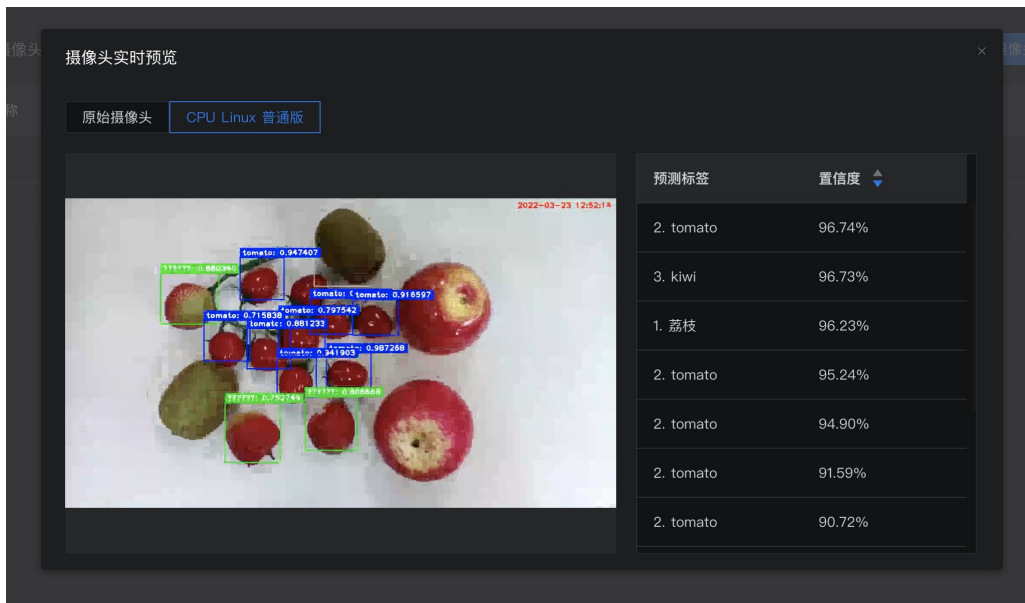
- 模型服务可接入摄像头直接进行预测，并可同时设置告警规则，出发告警条件的结果将会以事件的形式保存至IECC中。点击模型服务管理页面中对应服务的接入摄像头操作



- 将已添加至IECC的摄像头与模型服务关联，并在下方设置对应的事件告警条件。告警规则通过标签阈值的方式来建立，例如设置“猕猴桃”标签阈值80%-100%，则大于80%置信度的“猕猴桃”识别结果将会保存至事件记录中

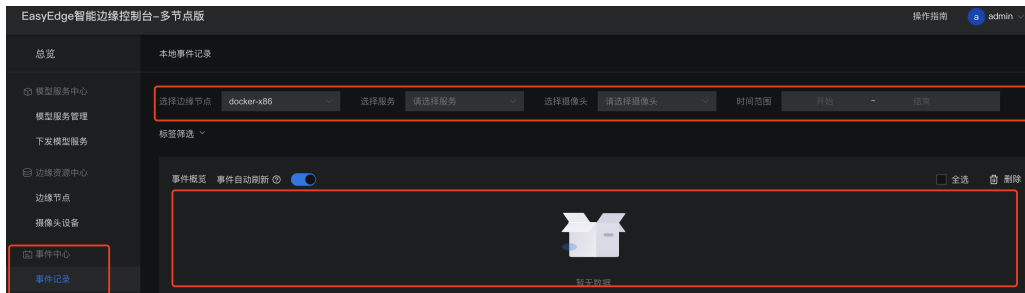


- 也可在摄像头设备页面-实时预览中查看实时的模型服务预测结果



Step 5 视频事件告警

- 可在事件中心-事件记录中查看满足时间告警条件的图片记录



高级配置说明 在系统设置 - 高级，可以修改控制中心的高级系统配置

```
##### IECc系统配置
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
role: hub
# 硬件利用率刷新时间间隔：过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# IECc保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: default
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge)
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
ToFile: yes
loggingFile: /var/log/easyedge-iecc/easyedge-iecc.log
# 0:info; -1:debug; -2:verbose
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8602
listenHost: 0.0.0.0

commu:
mqServer:
  host: 0.0.0.0
  port: 8632
  HTTPPort: 8620
  maxPayload: 8388608
  pingIntervalSecond: 30
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2
nodeRefreshIntervalSecond: 30

##### ----- 以下高级配置一般无需修改 -----
##### !!!注意!!! 请确保理解配置项含义后再做修改
##### 数据库相关配置
db:
  sqliteDbFile: /var/lib/easyedge-iecc/easyedge-iecc.db
  hubDbFile: /var/lib/easyedge-iecc/easyedge-iecc.hub.db
  eventDbFile: /var/lib/easyedge-iecc/easyedge-event.db
  fileServerDbFile: /var/lib/easyedge-iecc/easyedge-fileserver.hub.db
  nodeMonitorDbFile: /var/lib/easyedge-iecc/easyedge-nodemonitor.hub.db

##### 推流相关配置
mediaserver:
  flvPort: 8613
  rtmpPort: 8614

##### 文件服务器相关配置
fileserver:
  root: /var/lib/easyedge-iecc/fs
```

## FAQ

启动服务后，进程中出现两个 `easyedge-iec` 进程 这是正常现象，IEC通过守护进程的方式来完成更新等操作。

启动服务时，显示端口被占用 `port already been used` 通过修改 `easyedge-iecc.yml` 文件的配置后，再重新启动服务。

安装服务时，报错 `permission denied` 请以管理员身份运行安装程序。

添加SDK时，报错 **SDK不支持该硬件。SDK not supported by this device** 一般是因为使用的SDK跟硬件不匹配，如 GPU的SDK，硬件没有GPU卡。对于Jetson，也可能是Jetpack版本不支持，可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表（cpp文件中的文件名来查看）来匹配。

## 模型加速整体说明

**功能简介** 当您发布时纯离线服务时，平台已结合最新的量化、剪枝、蒸馏技术，推出丰富的模型压缩加速方案，以提高您的SDK部署效率。

**覆盖范围**：服务器、通用小型设备、专项适配硬件均支持该功能。

**具体原理**：针对目标芯片，对模型做深度优化压缩加速，加速后模型在推理速度、内存占用、体积大小等指标上表现更优。发布加速模型可能需要一段时间，同时会有微小的精度损失。发布完成后可通过性能报告对比具体加速效果。

**使用流程** **选择加速方式** 结合选择的系统与芯片不同，分别为您提供不同的压缩方式。

纯离线服务 > 发布新服务

操作文档 教学视频 常见问题 提交工单

说明：

- 本地服务器部署支持将模型部署于本地的CPU、GPU服务器上，提供API和SDK两种集成方式：[查看文档](#)
- 本地服务器SDK：将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活
- 集成步骤：① 申请SDK并在服务详情页下载SDK → ② 在控制台申请激活序列号 → ③ 根据开发文档集成SDK，并联网激活使用。如存在设备无法联网，需要在纯离线的环境下激活的情况，请[提交工单](#)联系我们。
- 个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

选择部署形式：① 选择部署形式 ② 填写个人信息

集成方式： SDK  API

选择模型：dog-cat-test

选择版本：V2

选择系统和芯片： Linux  Windows

通用X86 CPU 英伟达GPU 华为 Atlas 300 百度 昆仑XPU

模型加速： 基础-无加速 无加速  精度无损压缩加速 在精度尽可能无损的前提下加速模型  精度微损压缩加速-中 在部分芯片上，内存/存储空间占用降低，推理速度可以获得一定提升 [原封免费](#)

下一步

提示：基础SDK默认作为勾选选项存在，可后续与您的加速SDK进行效果与性能比对，方便您进一步挑选

**查看发布状态** 点击完成发布后，将自动跳转至列表页，可分别查看不同加速方案下的模型发布进度及发布时间。

服务器 通用小型设备 专项适配硬件

输入模型名称

此处发布、下载的SDK为未授权SDK，需要前往控制台[获取序列号](#)激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	操作
dog-cat-test	115215-V2	通用X86 CPU-Linux	基础版	● 发布中	2021-05-13 20:49	下载SDK
			精度无损压缩加速	● 发布中	2021-05-13 20:49	下载加速版SDK

## 常见问题

### 数据相关问题

**需要上传多少张图片才能训练出效果较好的模型？**

- 每种要识别的物体在所有图片中出现的数量需要大于50。如果某些要区分的物体具有相似性，需要增加更多图片。

**上传图片的总量有限制吗？**

- 每个账号下所有数据集的图片总数不能超过10万张。

### 训练相关问题

**数据处理失败或者状态异常怎么办？**

- 如是是图像分类模型上传处理失败，请先检查已上传的分类命名是否正确，是否存在中文命名、或者增加了空格；然后检查下数据图片量是否超过上限（10万张）；再检查图片中是否有损坏。如果自查没有发现问题请在百度智能云控制台内[提交工单](#)反馈

**模型训练失败怎么办？**

- 如果遇到模型训练失败的情况，请在百度智能云控制台内[提交工单](#)反馈

### 已经上线的模型还可以继续优化吗？

- 已经上线的模型依然可以持续优化，操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练，然后在模型列表中更新线上服务，完成模型的优化

点击我的模型列表——找到新训练好的模型版本——点击申请发布

应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V2	训练完成	未申请	未发布	top1准确率87.61% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">校验</a> <a href="#">训练</a>
高线SDK	V1	训练完成	未申请	未发布	top1准确率85.84% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">训练</a>

每页显示 12 < 1 >

在出来的弹窗中点击确定



### 🔗 模型效果相关问题

#### 物体检测模型如何正确标注？

- 所有图片中出现的目标物体都需要被框出（框可以重叠）
- 框应包含整个物体，且尽可能不要包含多余的背景
- 如果图片中存在很多相同标签的目标物体，可以使用右侧的锁定按钮。锁定标签后，只需要在左侧框选目标物体即可，不用再重复选择标签

#### 如何通过「完整评估结果」里的错误示例优化模型？

- 错误示例中，左侧是正确的结果，右侧是模型的识别结果
- 观察模型识别有误的图片有哪些共同点，并有针对性地补充训练数据。比如：当图片比较亮的时候模型都能识别正确，但比较暗的时候模型就识别错了。这时就需要补充比较暗的图片作为训练数据

#### 我的数据有限，如何优化效果？

- 先申请发布模型，并备注说明希望通过[云服务数据管理](#)功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

#### 实际调用服务时模型效果变差？

- 训练图片和实际场景要识别的图片拍摄环境应一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片
- 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 如果使用的是云服务，可以开通[云服务数据管理](#)功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

如果训练数据已经达到以上要求，且单个分类/标签的图片量超过200张以上，效果仍然不佳，请在百度智能云控制台内[提交工单](#)反馈

### 🔗 智能标注相关问题

智能标注功能目前已对物体检测模型开放，[了解功能详情](#)

#### “一键标注”和“立即训练”要如何选择？

- 当系统推荐“立即训练”，且系统预标注的框确实已非常精准时，可以不用标注剩余数据，直接开始模型训练。此时，仅用当前已标注图片训练

的模型，与标注所有数据后训练的模型相比，效果几乎等同

- 如果系统预标注的框还有些不精准，可以启动一键标注，人工确认系统标注的标注框后，再开始训练

选择了“立即训练”之后是否还可以“一键标注”？

- 选择“立即训练”之后，系统默认为您结束此次智能标注
- 再次启动智能标注后，您可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

智能标注结束后，又往数据集上传了新图片，是否可以直接“一键标注”新图片？

- 如果您创建了新的标签、或新上传的图片场景和之前的图片场景差异较大，建议不要使用一键标注，而是从头开始智能标注（即再次筛选关键图片）
- 如果不是以上情况，再次启动智能标注后，可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

智能标注中可以增删标签吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签
- 如果确实需要增删标签，可以先结束智能标注

智能标注中可以增删图片吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片
- 如果确实需要增删图片，可以先结束智能标注

智能标注中可以修改已标注图片的标注框吗？

- 可以。但为了保证智能标注的效果，建议不要大量改动
- 如果确实需要修改大量标注，建议先结束智能标注

为什么我已经人工标注了很多图片，但系统预标注依然不准？

- 系统预标注的结果会受以下因素影响：
  - 智能标注期间，对“已标注”图片的标签进行大量改动
  - 曾结束智能标注，并对标签、图片进行增删
- 如果您没有进行以上操作，系统标注结果依然不理想，请在百度智能云控制台内[提交工单](#)反馈

多个数据集是否可以同时启动智能标注？

- 目前每个账号同一时间仅支持对一个数据集启动智能标注

共享中的数据集是否可以启动智能标注？

- 暂不支持。智能标注中的数据集也暂不支持共享，如有疑问请在百度智能云控制台内[提交工单](#)反馈

智能标注失败了怎么办？

- 可以先尝试稍后重新启动
- 若再次遇到问题，请在百度智能云控制台内[提交工单](#)反馈

🔗 模型上线相关问题

希望加急上线怎么处理？

- 请在百度智能云控制台内[提交工单](#)反馈

每个账号可以上线几个模型？是否可以删除已上线的模型？

- 每个账号最多申请发布十个模型，已上线模型无法删除

申请发布模型审核不通过都是什么原因？

- 可能原因有，1、经过电话沟通当前模型存在一些问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝。2、电话未接通且模型效果较差，会直接拒绝。如果需要申诉，请在百度智能云控制台内[提交工单](#)反馈

#### ☞ 离线SDK发布问题

发布SDK时可以选择的操作系统、芯片和加速类型与什么有关？

- 与训练时选择的算法有关，以「物体检测-矩形框」为例，可以通过该链接查看适配的硬件以及不同算法在不同硬件下的性能表现：[算法性能和适配硬件](#)，同时也可观看视频介绍：



## 图像分割

### 整体介绍

#### ☞ 简介

Hi，您好，欢迎使用百度EasyDL图像。

EasyDL图像支持定制图像分类、物体检测、图像分割三类模型。三类模型的功能区别如下：

- 图像分类：识别一张图中是否是某类物体/状态/场景，适用于图片内容单一、需要给整张图片分类的场景
- 物体检测：检测图中每个物体的位置、名称。适合图中有多个主体要识别、或要识别主体位置及数量的场景
- 图像分割：对比物体检测，支持用多边形标注训练数据，模型可像素级识别目标。适合图中有多个主体、需识别其位置或轮廓的场景

以下是关于图像分割模型的技术文档。

#### ☞ 应用场景

- 专业检测：应用于专业场景的图像分析，比如在卫星图像中识别建筑、道路、森林，或在医学图像中定位病灶、测量面积等
- 智能交通：识别道路信息，包括车道标记、交通标志等

#### ☞ 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作。在数据已经准备好的情况下，最快几分钟即可获得定制模型。

下面将详细介绍每一步的操作方式和注意事项。如果文档没有解决您的问题，请在百度智能云控制台内[提交工单](#)反馈。



## 数据准备

#### ☞ 创建数据集

在训练之前需要在数据中心【创建数据集】。

如果训练数据需要多人分工标注，可以创建多个数据集。将训练数据分批上传到这些数据集后，再将数据集“共享”给自己的小伙伴，同步进行标注。

### 设计标签

在上传之前确定想要识别哪几种物体，并上传含有这些物体的图片。每个标签对应想要在图片中识别出的一种物体

**注意：标签的上限为1000种**

### 准备图片

基于设计好的标签准备图片：

- 每种要识别的物体在所有图片中出现的数量需要大于50
- 如果某些标签的图片具有相似性，需要增加更多图片
- 一个模型的图片总量限制4张~10万张

### 图片格式要求：

- 1、目前支持图片类型为png、jpg、bmp、jpeg，图片大小限制在14M以内
- 2、图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px

### 图片内容要求：

- 1、训练图片和实际场景要识别的图片拍摄环境一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片
- 2、每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

**如果需要寻求第三方数据采集团队协助数据采集，请在百度智能云控制台内[提交工单](#)反馈**

## 🔗 上传数据集并在线标注

在完成了设计标签与准备数据后，可以通过以下方式导入数据：

- 导入未标注的数据，在线进行数据标注
- 直接导入标注好的数据

### 导入未标注数据

#### 本地数据

支持上传图片、压缩包，或通过[API导入](#)

#### 已有数据集

支持选择百度云BOS导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他图像类模型的数据集



EasyDL
产品介绍
操作平台
应用案例
使用文档

**物体检测模型** ☰

- ☰ 总览
- 📁 模型中心
- 我的模型
- 创建模型
- 训练模型
- 校验模型
- 发布模型
- ☑ EasyData数据服务
- 数据总览
- 标签组管理
- 在线标注
- 云服务数据回流
- 摄像头数据采集
- ☰ 公有云服务
- 在线服务
- 批量预测
- ☑ EasyEdge本地部署
- 纯离线服务
- 端云协同服务 ^

我的数据总览 > 水果测试/V1/导入

**创建信息** ▼

数据集ID	287421	版本号	V1
备注	<a href="#">🔗</a>		

**标注信息** ▼

标注类型	物体检测	标注模板	矩形框标注
数据总量	19	已标注	5 (进度26.32%)
标签个数	2	标注框数	8
待确认	0	大小	0.5M

**数据清洗**

暂未做过数据清洗任务

**导入数据**

数据标注状态  无标注信息  有标注信息

导入方式 请选择 ^

一键导入 Labelme 已

- BOS目录导入
- 分享链接导入
- 平台已有数据集
- 摄像头采集数据
- 云服务回流数据

## 在线标注

### 标注方式

在【数据标注/上传】页面上传并在线标注图片：

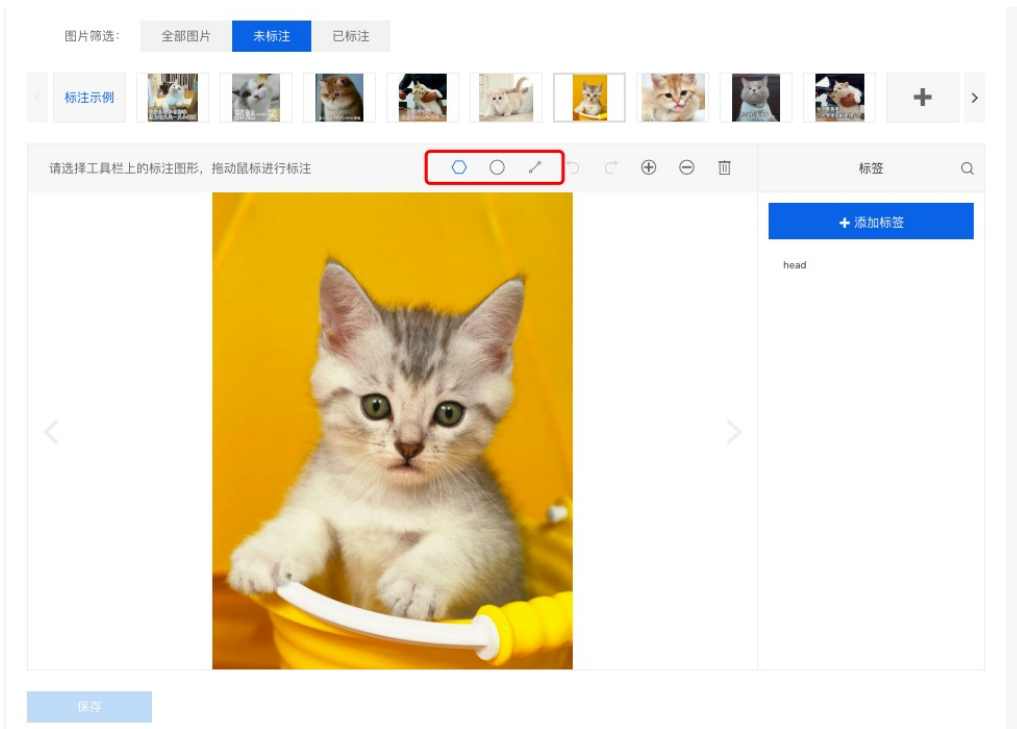
**Step 1**选择数据集

**Step 2**上传已准备好的图片

**Step 3**在标注区域内进行标注

首先在标注框上方找到工具栏，点击标注按钮在图片中拖动画框，圈出要识别的目标

然后在右侧的标签栏中，增加新标签，或选择已有标签



**自动识别轮廓标注** 推荐使用自动识别轮廓工具进行标注，鼠标左键点击目标即可自动出现标注区，鼠标右键点击误识别的区域可取消误识别区域的标注，反复操作即可精炼出十分准确的标注结果



**图片标注Tips**

- 所有图片中出现的目标物体都需要被框出（框可以重叠）



全部框出



部分框出

- 框应包含整个物体，且尽可能不要包含多余的背景



包含整个物体



未框全或包含多余背景

- 如果图片中存在很多相同标签的目标物体，可以使用右侧的锁定按钮。锁定标签后，只需要在左侧框选目标物体即可，不用再重复选择标签

## 🔗 数据集智能标注

使用智能标注功能可降低数据的标注成本。启动后，系统会从数据集所有图片中筛选出最关键的图片并提示需要优先标注。通常情况下，只需标注数据集30%左右的数据即可训练模型。与标注所有数据后训练相比，模型效果几乎等同

整体流程以物体检测的智能标注流程为例：

### 创建智能标注任务

启动物体检测数据集的智能标注前，请先检查以下是否已满足以下条件：

- 所有需要识别的标签都已创建
- 每个标签的标注框数不少于10个
- 所有需要标注的图片都已加入数据集，且所有不相关的图片都已删除

若已满足，即可从导航栏进入「数据服务」-「智能标注」，创建智能标注任务，系统会基于您选择数据类型及数据量级，自动预估任务运行时长



### 系统筛选难例

系统会分批筛选出最关键需标注的图片，即难例图片。

Tips：难例筛选需要一定时间，在此期间您可以正常进行其他未标注图片的标注



### 用户确认难例

智能标注任务启动后，系统为您自动筛选难例，您可以通过总览页查看进度按钮查看当前难例筛选进度，同时，进度图中也会全局展示您处于难例筛选的具体哪一环节，以便您的操作后续。筛选难例完成后，绿色进度条会进展到确认难例阶段，您可以点击【确认难例】完成对预标注结果的人工确认。

创建智能标注任务  图像智能标注任务  文本智能标注任务

序号	数据集ID	数据集名称	版本	智能标注状态	操作
1	3107	tj-智能标注-检测-demo	V2	已中止	<a href="#">重新启动</a> <a href="#">查看记录</a>
2	2751	xyf_test_data1	V1	已中止	<a href="#">重新启动</a> <a href="#">查看记录</a>
3	256	test123	V1	运行中	<a href="#">查看进度</a> <a href="#">难例确认</a> <a href="#">中止任务</a> <a href="#">查看记录</a>
4	2744	zzy-测试智能标注			<a href="#">查看记录</a>
5	3111	tj-智能标注-检测-1110			<a href="#">查看记录</a>
6	2831	py3升级-智能标注-物体检测			<a href="#">查看记录</a>
7	1965	物体检测-多人标注用			<a href="#">查看记录</a>
8	2981	赵鸾专属数据集1			<a href="#">查看记录</a>

当前您处于第1轮难例阶段（共4轮）  
已为您筛选出本轮难例图片，请确认该轮难例图片

● 任务完成

● 效果评估

● 下一轮筛选

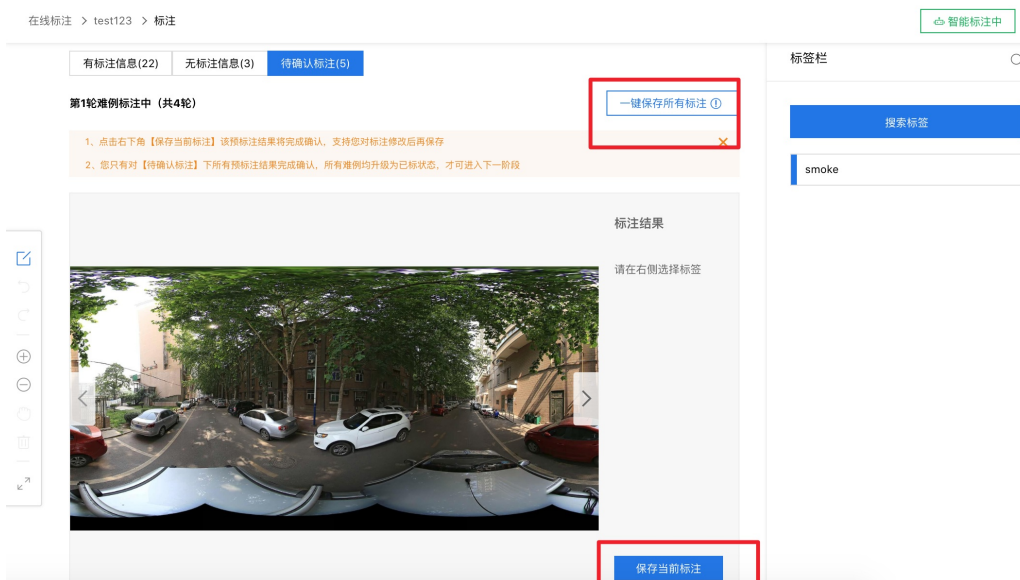
● 筛选难例

● 难例确认

[知道了](#)

我们为您的人工确认提供两种模式：

- 单张确认，在该模式下支持您对预标注结果进行修正后点击保存
- 一键保存所有标注，为提升您的确认效率，默认您对难例的预标注结果全部满意，即可进入下一阶段



标注难例的预训练模型，也会对您无标注信息下的图片进行预标注结果的展示，您有余力的情况下，可以完成标注确认，确认后该张图片将升级为已标状态，该环节并非是您进入智能标注下一阶段的必备要求。



**评估难例效果，完成任务**

当您对难例完成确认后，您可以根据本轮次预标注的结果是否满意，判断您是否还需要进入下一轮难例筛选阶段，如果满意本轮难例的预标注效果，系统将自动为您系统其他的未标图片打标签。

### 第1轮难例标注中（共4轮）

- 1、点击右上角【保存当前标注】该预标注结果将完成确认，支持您对标注修改后再保存
- 2、您只有对【待确认标注】下所有预标注结果完成确认，所有难例均升级为已标状态，才可进入下一阶段



### 中止任务

当您在任务运行中想要中止任务时，可实时点击标注页面右上方【中止任务】按钮，任务将被提前结束。



### 其他操作提示

- 在智能标注任务中，有任务上限吗？

支持五条智能标注任务同时运行，超过该上限您需要中止其他任务

- 智能标注中可以增删标签吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签 如果确实需要增删标签，可以先结束智能标注

- 智能标注中可以增删图片吗？

暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片。如果确实需要增删图片，可以先结束智能标注

- 智能标注中可以修改已标注图片的标注框吗？

可以。但为了保证智能标注的效果，建议不要大量改动。如果确实需要修改大量标注，建议先结束智能标注

- 为什么我已经人工标注了很多图片，但系统预标注依然不准？

系统预标注的结果会受以下因素影响：智能标注期间，对“已标注”图片的标签进行大量改动；曾结束智能标注，并对标签、图片进行增删

- 多个数据集是否可以同时启动智能标注？

目前每个账号同一时间仅支持对一个数据集启动智能标注

- 共享中的数据集是否可以启动智能标注？

暂不支持。智能标注中的数据集也暂不支持共享

- 智能标注失败了怎么办？

可以先尝试稍后重新启动，如多次失败请[提交工单](#)联系我们

## 问题反馈

您在使用EasyData过程中可以通过以下任何方式联系我们：

- 在社区咨询

在论坛发帖提交问题，也可以在论坛与其他用户一起交流。[前往论坛](#)

- 提交工单

如果使用EasyData遇到其他任何问题或任何bug，您可以点此[提交工单](#)

- 添加微信小助手留言

请在微信搜索“BaiduEasyDL”，并备注暗号“EasyData”，添加小助手后留言。

## 🔗 数据集多人标注

如果训练数据需要多人分工标注，可以创建多个数据集。将训练数据分批上传到这些数据集后，再将数据集“共享”给自己的小伙伴，同步进行标注。

共享方式如下：

### 1. 在「数据集管理」页面，点击需要共享的数据集对应操作栏中的「共享」

ID	名称	类型	标签数	图片数	状态	操作
44339	MM豆	物体检测	1	60	正常	查看 标注/上传 智能标注历史 删除 <b>共享</b>
38867	更上层楼	物体检测	2	23	正常	查看 标注/上传 智能标注历史 删除 共享
27672	new	物体检测	0	0	新建	查看 标注/上传 删除 共享
9798	super_band_2	物体检测	3	107	正常	查看 标注/上传 删除 共享
8994	super_band	物体检测	3	7638	正常	查看 标注/上传 删除 共享详情
8096	马	物体检测	2	20	正常	查看 标注/上传 删除 共享

### 2. 在共享页面，勾选被共享数据集的授权使用范围，生成共享链接。如需被共享人标注数据，则需勾选「修改」

温馨提示：为了更有序地进行标注，每个数据集仅支持共享给一位用户。若一批训练数据需要多人共同标注，请先将数据拆分并上传到不同的数据集。训练时可从多个数据集选择数据进行训练。

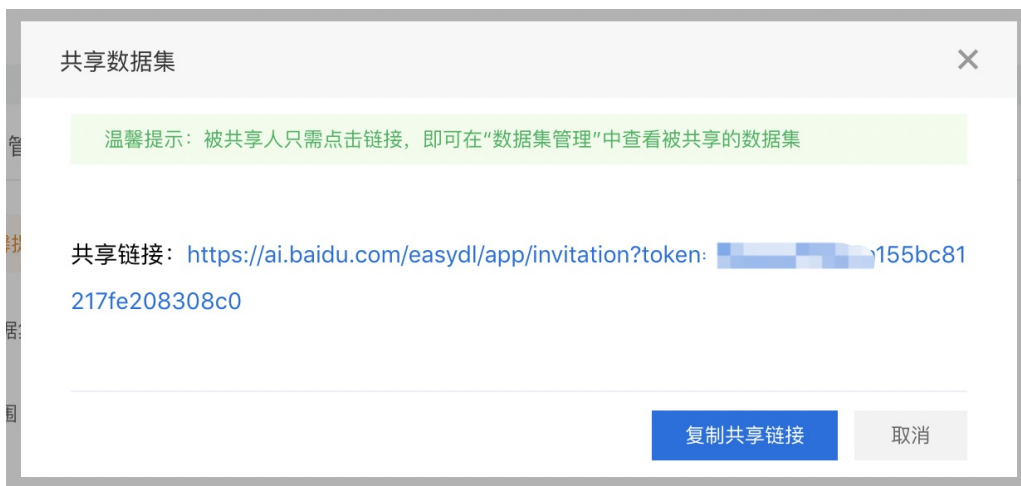
共享数据集：MM豆

授予范围： 查看  修改  使用

生成共享链接 返回

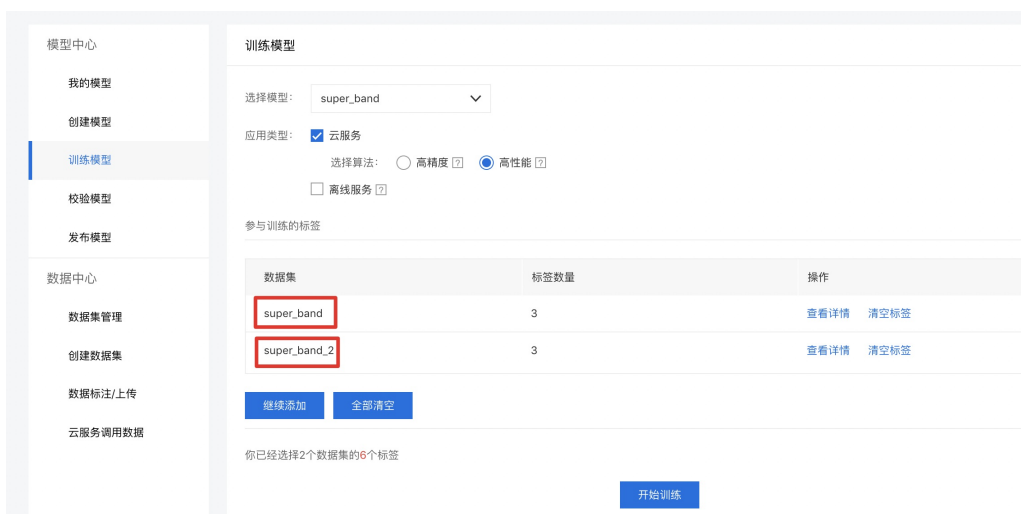
### 3. 复制共享链接，并发送给小伙伴





4. 被共享人打开链接后, 即可在「数据集管理」页面看到被共享的数据集, 并进行被授权的操作

5. 训练模型时, 在「训练模型」页面添加训练数据时, 可从多个数据集 (如多个被共享的数据集) 选择数据



## 🔗 数据集管理API

本文档主要说明当您线下已有大量的图片数据, 如何通过调用API完成图片的便捷上传和管理。EasyDL图像数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据, 只是在部分接口入参存在差异, 使用及接口地址完全一致。

### 数据集创建API

#### 接口描述

该接口可用于创建数据集。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法: POST

请求URL: <https://aip.baidubce.com/rpc/2.0/easydl/dataset/create>

URL参数:

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>



Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

#### 数据集列表API

##### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

#### 分类（标签）列表API

##### 接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

##### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

#### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
append_label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类10000个汉字</b>
entity_name	是	string	文件名

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 数据集删除API

#### 接口描述

该接口可用于删除数据集。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 分类（标签）删除API

##### 接口描述

该接口可用于删除分类（标签）。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL控制台](#)-公有云部署-应用列表页面创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/label/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

##### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 数据质检

**功能概述** 该功能旨在对您数据集中的图像数据进行质量检测，通过提供客观指标，为您对数据集的下一步操作（标注、清洗等）进行参照引导。

整体质检报告将包括对原图、标注信息两个层面的指标进行统计，本期先上线原图维度的质检指标，标注层面的质检指标敬请期待。

### 使用流程 Step 1 功能入口

您可从数据总览页操作列点击【质检报告】或查看页面点击【质检报告】进入该功能页面

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗状态	操作
V1	142909	5	● 已完成	图像分类	0% (0/5)	-	查看与标注 导出 删除 <b>质检报告</b>

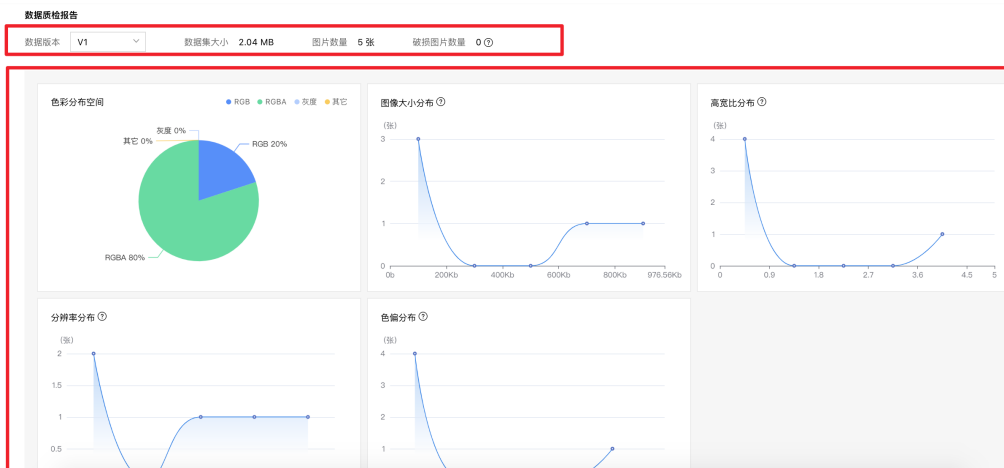
我的数据总览 > 【图片】的/V1查看

全部 (5) 有标注信息 (0) 无标注信息 (5) + 导入图片 **质检报告** 批量标注示例

的V1版本的图片列表 筛选  本页全选  删除

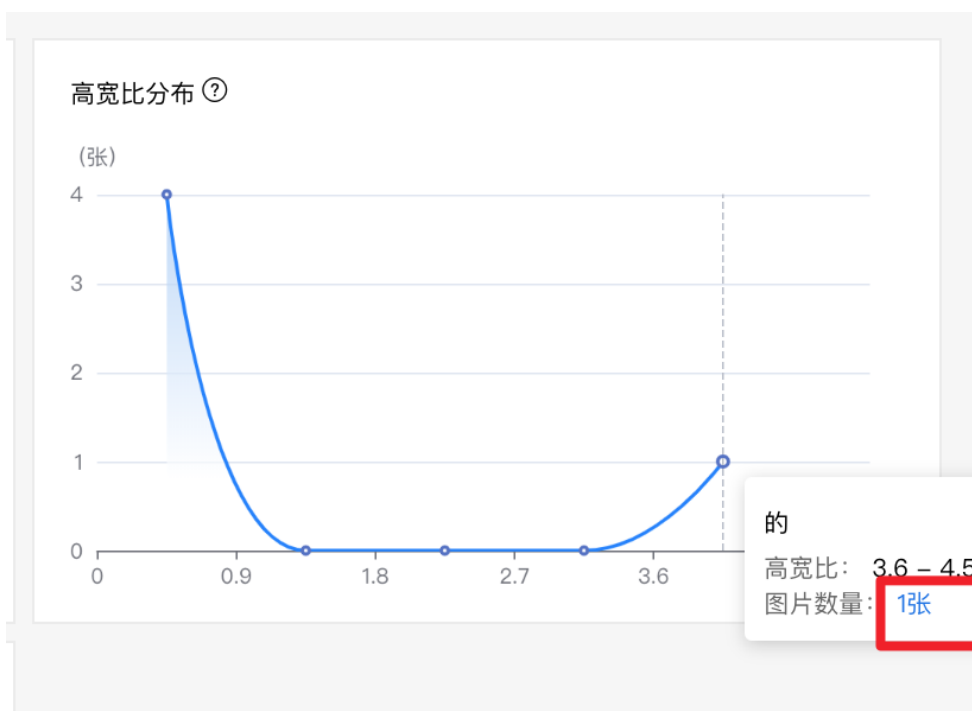
**Step 2 指标查看** 本期报告分为整体指标和分布指标两类。整体指标包括数据集存储大小、图片数量、破损图像数三类；分布指标包括色彩分布空间、图像存储大小分布、高宽比分布、分辨率分布、色偏分布五类。

可以通过切换数据集版本查看不同版本下质检报告。



**Step 3 对应处理** 可通过hover具体指标数值进行相关操作，以高宽比分布为例：

第一步，高宽比大于3.6的超长图hover显示有1张图片比，支持点击



第二步，点击后进入符合该指标的图片操作页，可针对筛选后图片进行删除、标注等操作



## 模型训练

### 🔗 图像分割创建模型

在导航【创建模型】中，选择任务场景，填写模型名称、联系方式、功能描述等信息，即可创建模型。

其中任务场景分为**实例分割**和**语义分割**

**语义分割**：图像分割指将每个像素点归属为对象类的过程。其中，语义分割适用于分割目标主体单一的场景，简单举例来说语义分割能够识别出图片中哪些像素是归属于“人”的标签，但无法区分“不同的人”

**实例分割**：图像分割指将每个像素点归属为对象类的过程。其中，实例分割会先定位目标再进行分割，简单举例来说，相比语义分割，实例分割能够识别出“人”的同时，还能区分“不同的人”



图像分割模型

模型列表 > 创建模型

模型类别 图像分割

任务场景 \* 实例分割 语义分割

模型名称 \*

您的身份 企业管理者 企业员工 学生 教师

公司名称 \* 请输入公司名称 点击完成企业认证即可获得企业专属权益礼包

企业认证流程较快, 认证过程中您可继续创建模型, 完成后系统会自动同步状态, 并为您发放企业权益礼包至账户

所属行业 \* 请选择行业

应用场景 \* 请选择应用场景

邮箱地址 \* z\*\*\*\*\*@baidu.com

联系方式 \* 135\*\*\*\*\*919

功能描述 \*

完成

模型创建成功后, 可以在【我的模型】中看到刚刚创建的模型。

1. 创建模型后可持续新增模型版本, 因此不必每次训练模型都创建模型
2. 目前单个用户在每种类型的模型下最多可创建10个模型, 每个模型均支持多次训练。
3. 如果您是企业用户, 建议您按照真实企业信息进行填写, 便于EasyDL团队后续更好的为您服务

#### 图像分割训练操作说明

数据提交后, 可以在导航中找到【训练模型】, 按以下步骤操作, 启动模型训练:

1. 启动训练前请确保数据已经标注完成, 否则无法启动训练
2. 下述训练功能点中, 标注为星号 (\*) 的功能为非必要选择项, 可根据实际需求考虑是否使用



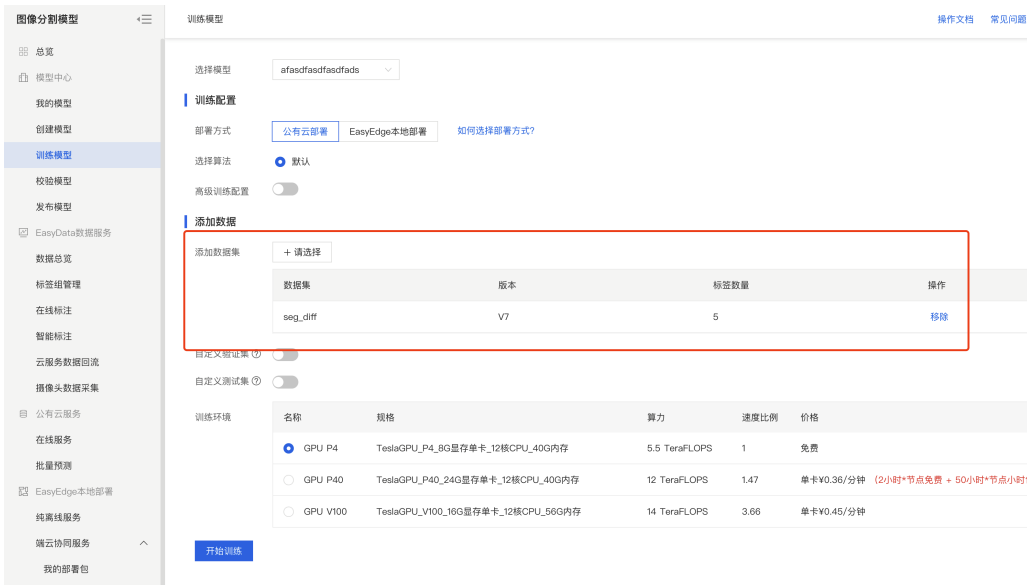
### ① 选择模型

选择此次训练的模型

### ② 添加训练数据

先选择数据集，再按标签选择数据集里的图片。可从多个数据集选择图片（相同标签的训练图片会被合并）

训练时间与数据量大小有关，1000张图片可能需要几个小时训练，请耐心等待



### ③ 选择部署方式

可选择「公有云API」、「EasyEdge本地部署」

#### 如何选择部署方式

**增量训练\*** 增量训练：在模型迭代训练时，用户在原训练数据上增加了训练数据，可通过加载原训练数据训练的模型参数进行模型训练。这样可以让模型收敛速度变快，训练时间变短，同时在数据集质量较高的情况下，可能获得的模型效果也会更好。

注：仅可选择同一部署方式下的训练的模型作为基准模型版本

### ④ 选择算法

当前仅支持选择默认算法

## ⑤ 训练模型

点击「开始训练」，训练模型。

- 训练时间与数据量大小有关，1000张图片可能需要几个小时训练，请耐心等待。
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面。
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

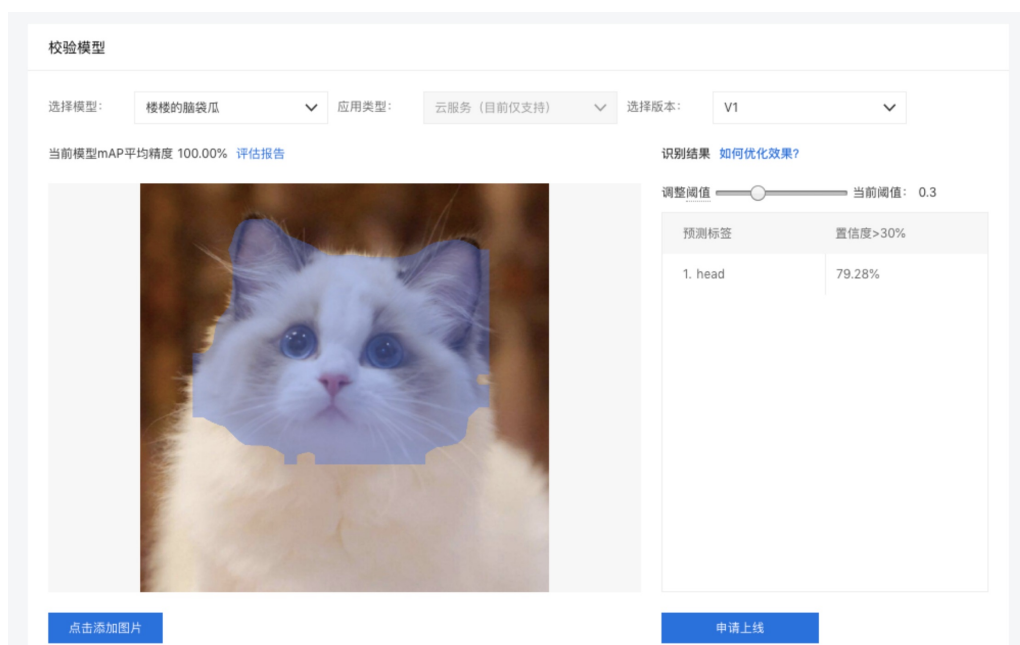
为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有图像分割操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

## 🔗 图像分割模型效果评估

可通过模型评估报告或模型校验了解模型效果：

- 模型评估报告：训练完成后，可以在【我的模型】列表中看到模型效果，以及详细的模型评估报告。
- 模型在线校验：可以在左侧导航中找到【校验模型】，在线校验模型效果。校验功能示意图：



## 模型评估报告

### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、mAP、精确率、召回率。这部分模型效果的指标是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如图片数量低于100个），参与评估的数据可能不超过30个，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

注意：若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口批量测试，获取更准确的模型效果。

### 实例分割

#### 整体评估

fenge01 V14效果优异, 建议针对识别错误的图片示例继续优化模型效果。由于目前训练集数据量较少, 该结论仅供参考, 建议扩充训练集得到更准确的评估效果。 [如何优化效果?](#)



查看模型评估结果时, 需要思考在当前业务场景, 更关注精确率与召回率哪个指标。是更希望减少误识别, 还是更希望减少漏识别。前者更需要关注精确率的指标, 后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况, 对于希望准确率与召回率兼具的场景, F1-score越接近1效果越好。评估指标说明如下

**F1-score**: 对某类别而言为精确率和召回率的调和平均数, 评估报告中指各类别F1-score的平均数

**mAP**: mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务, 每一类object都可以计算出其精确率(Precision)和召回率(Recall), 在不同阈值下多次计算/试验, 每个类都可以得到一条P-R曲线, 曲线下的面积就是average

**精确率**: 正确预测的物体数与预测物体总数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

**召回率**: 正确预测的物体数与真实物体数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

### 语义分割

#### 整体评估

dxu\_bisenet\_01 V2整体效果欠佳, 建议针对识别错误的图片示例继续优化模型效果。由于目前训练集数据量较少, 该结论仅供参考, 建议扩充训练集得到更准确的评估效果。 [如何优化效果?](#)



Kappa系数: 0.9147

**mIoU**: mIoU(mean intersection over union) 是语义分割常用评价指标, mean是对于类别的平均, 即每个类别IoU的平均值。某个类别的IoU计算方式如下为, (当前类别预测正确像素点个数) / (当前类别预测正确像素点个数 + 本属于当前类却被预测为其他类像素点个数 + 本属于其他类却被预测为当前类的像素点个数)。由于mIoU是像素级别的交并比评估, 数值上会略低于mAP, 不会影响实际使用效果。

**准确率**: 指类别预测正确的像素占总像素的比例, 准确率越高模型质量越好。

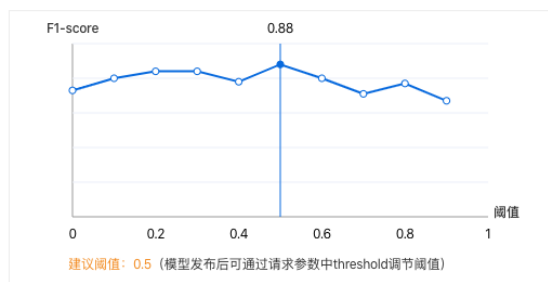
**Kappa系数**: 用于一致性检验的指标, 可以用于衡量分类的效果。取值为-1到1之间, 通常大于0, Kappa系数越高模型质量越好。

### 详细评估

在这个部分可以看到不同阈值下的F1-score, 以及模型识别错误的图片示例。

#### 详细评估

不同阈值下F1-score表现



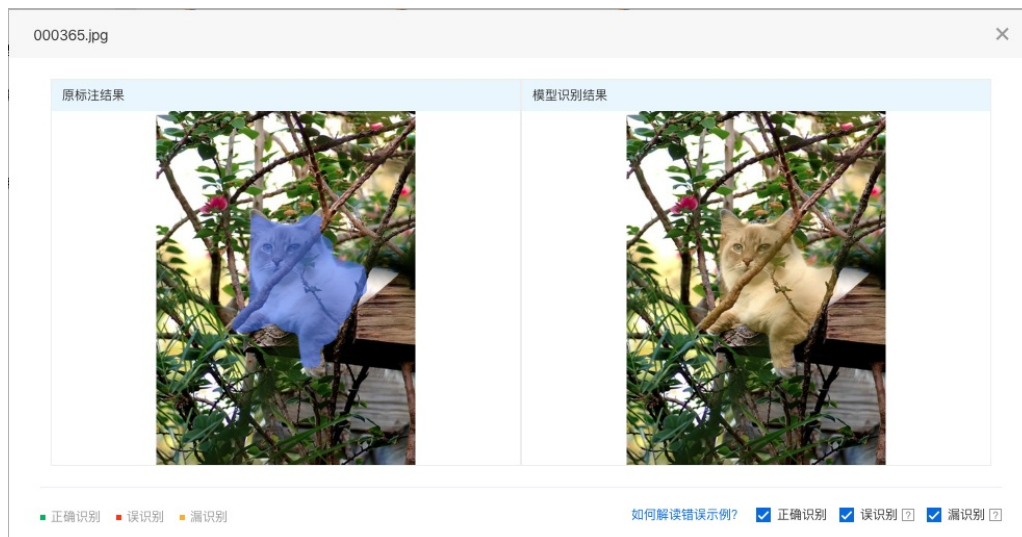
不同标签的mAP及对应的识别错误的图片

cat	84%	dog的错误结果示例 (点击查看识别错误详情)
dog	97%	

### 识别错误图片示例

通过分标签查看模型识别错误的图片, 寻找其中的共性, 进而有针对性的扩充训练数据。

如下图所示，可以通过勾选「误识别」、「漏识别」来分别查看两种错误识别的情况：



- **误识别**：红色遮盖内没有目标物体（准备训练数据时没有标注），但模型识别到了目标物体

观察误识别的目标有什么共性：例如，一个检测电动车的模型，把很多自行车误识别成了电动车（因为电动车和自行车外观上比较相似）。这时，就需要在训练集中为自行车特别建立一个标签，并且在所有训练集图片中，将自行车标注出来。

可以把模型想象成一个在认识世界的孩童，当你告诉他电动车和自行车分别是什么样时，他就能认出来；当你没有告诉他的时候，他就有可能把自行车认成电动车。

- **漏识别**：橙色遮盖内应该有目标物体（准备训练数据时标注了），但模型没能识别出目标物体

观察漏识别的目标有什么共性：例如，一个检测会议室参会人数的模型，会漏识别图片中出现的白色人种。这大概率是因为训练集中缺少白色人种的标注数据造成的。因此，需要在训练集中添加包含白色人种的图片，并将白色人种标注出来。

黄色人种和白色人种在外貌的差别上是比较明显的，由于几乎所有的训练数据都标注的是黄色人种，所以模型很可能认不出白色人种。需要增加白色人种的标注数据，让模型学习到黄色人种和白色人种都属于「参会人员」这个标签。

以上例子中，我们找到的是识别错误的图片中，目标特征上的共性。除此之外，还可以观察识别错误的图片在以下维度是否有共性，比如：图片的拍摄设备、拍摄角度，图片的亮度、背景等等。

## 🔗 图像分割模型如何提升效果

一个模型很难一次性就训练到最佳的效果，可能需要结合模型评估报告和校验结果不断扩充数据和调优。

为此我们设计了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，获得更好的模型效果。

**注意**：如果模型已经是上线状态（包括已付费的模型服务），依然支持模型迭代。只需要在训练完毕后发布新的版本，就可以获得更新后的模型服务。

想要提升模型效果，可以尝试以下两种方法：

### 检查并优化训练数据

1. 检查是否存在训练数据过少的情况，建议每个标签标注50个目标以上，如果低于这个量级建议扩充。
2. 检查不同标签的标注目标数是否均衡，建议不同标签的标注目标数数据量级相同，并尽量接近，如果有的标签标注的很多，有的标签标注的很少，会影响模型整体的识别效果。
3. 通过模型效果评估报告中的错误识别示例，有针对性地扩充训练数据。
4. 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

### 云服务调用数据管理

开通云服务调用数据管理功能后，可查找云服务模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，实现训练数据的持续丰富和模型效果的持续优化

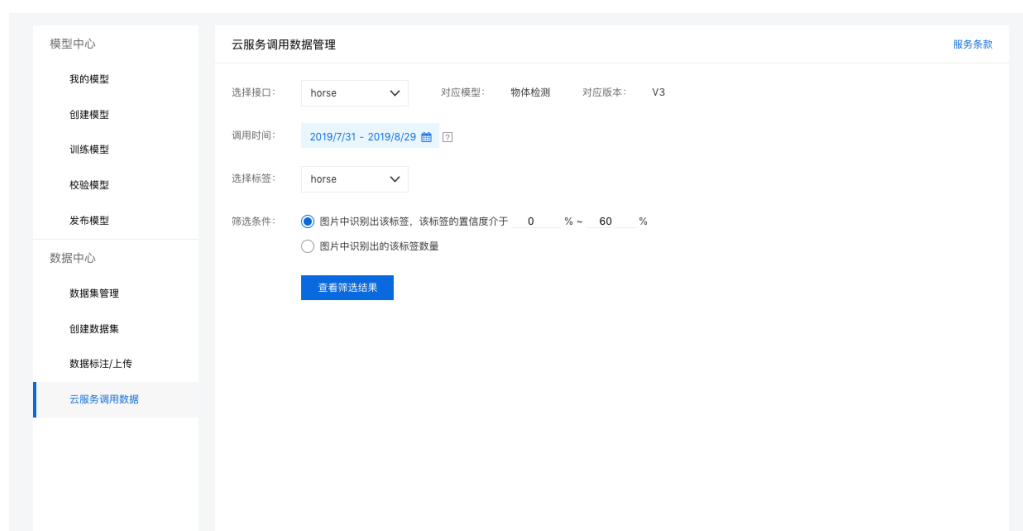
具体使用流程如下：

### 1. 为已上线接口开通云服务调用数据服务



### 2. 通过选择调用时间、标签，并设置筛选条件，查看疑似错误识别的图片

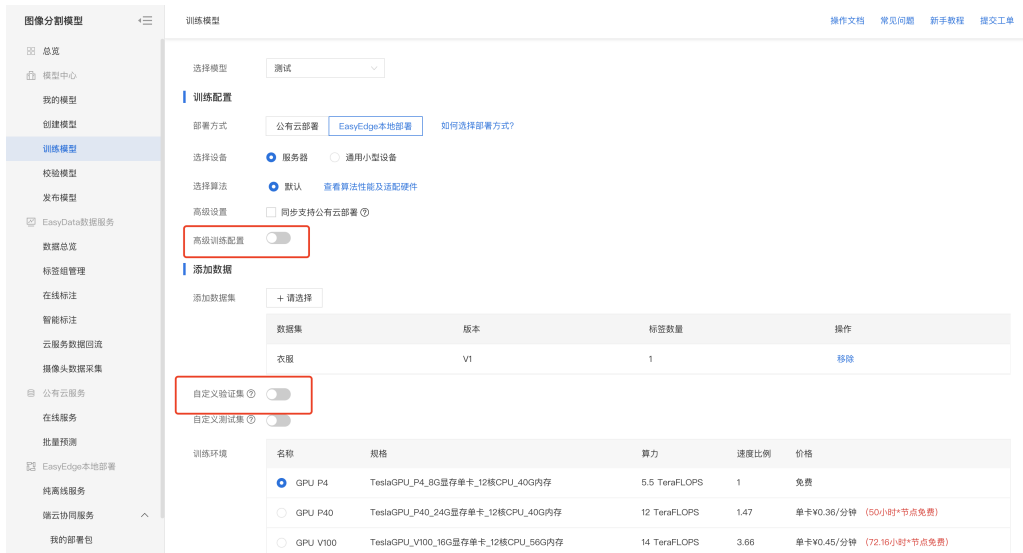
注意：数据将从开通功能后开始存储，最多存储30天的数据。当天调用的数据暂不支持即时查看，可在第二天查看



### 3. 将接口识别错误的图片添加到指定数据集（建议新建数据集）并纠正结果。后续训练模型时，只需增加包含接口数据的数据集，即可提升模型效果

**尝试不同的训练配置** 可前往训练配置页面尝试不同的配置组合，因不同数据集在不同的算法上可能表现不一致，所以建议您多尝试不同的算法选型后综合挑选精度最高的模型使用，你可以选择如下的配置项：

- 增量训练
- 在高级训练配置中增加输入图片分辨率



## 模型发布

### ☞ 图像分割模型发布整体说明

训练完成后，可将模型部署在公有云服务器、通用小型设备、本地服务器，或直接购买软硬一体方案，灵活适配各种使用场景及运行环境

#### 公有云在线服务

训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合

具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

支持查找云端模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，不断优化模型效果

**纯离线服务** 训练完成的模型整体打包为纯离线服务，可下载在本地稳定调用。纯离线服务按部署硬件芯片不同分为本地服务器部署、通用小型设备部署。为了提供更好的算法与硬件推理效果，EasyDL提供软硬一体方案部署。纯离线服务的整体支持与评测信息可见[算法与性能评测大表](#)

#### 本地服务器部署

可将训练完成的模型部署在私有CPU/GPU服务器上，支持服务器API和服务器SDK两种集成方式

模型服务性能表现更好，适用于对性能要求较高的场景，例如工业质检、流水线产品分拣等

#### 通用小型设备

训练完成的模型被打成适配智能硬件的SDK，可进行设备端离线计算。满足推理阶段数据敏感性要求、更快的响应速度要求

支持iOS、Android、Linux、Windows四种操作系统，基础接口封装完善，满足灵活的应用侧二次开发

## 端云协同服务

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新

断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）

联网状态下在平台管理设备运行状态、资源利用率

### ☞ 公有云部署

### ☞ 如何发布API

训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可发布公有云API：

- 选择模型
- 选择部署方式「公有云部署」



- 选择版本
- 自定义服务名称、接口地址后缀
- 申请发布

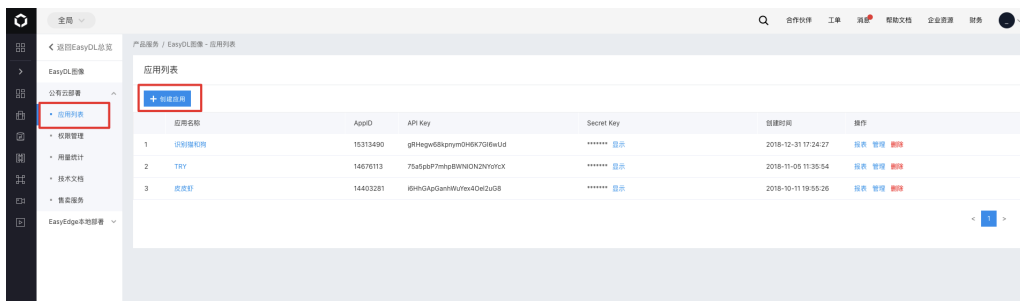
申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。如果需要加急、或者遇到莫名被拒的情况，请在百度智能云控制台内[提交工单反馈](#)

发布模型界面示意：



### 接口赋权

在正式使用之前，还需要做的一项工作为接口赋权，需要登录[EasyDL控制台](#)中创建一个应用，获得由一串数字组成的appid，然后就可以参考[接口文档](#)正式使用了

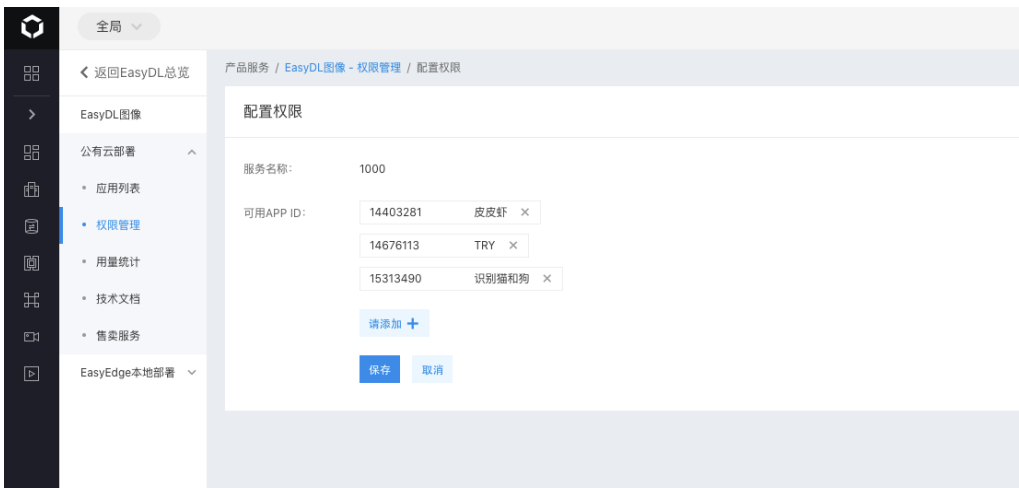


同时支持在「公有云服务管理」-「权限管理」中为第三方用户配置权限

示意图如下：







API调用文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

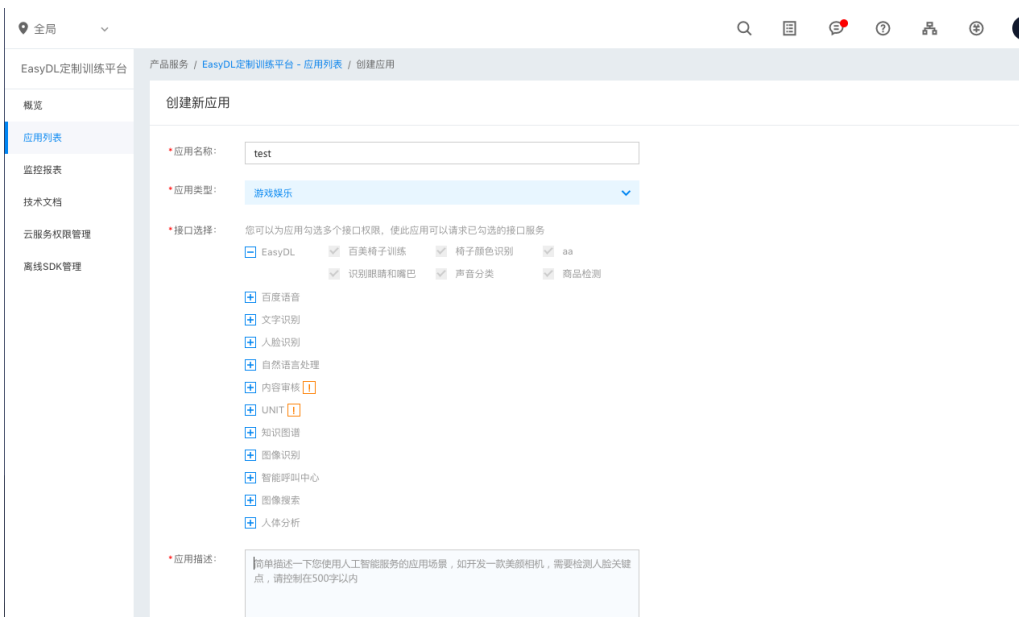
- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

接口描述

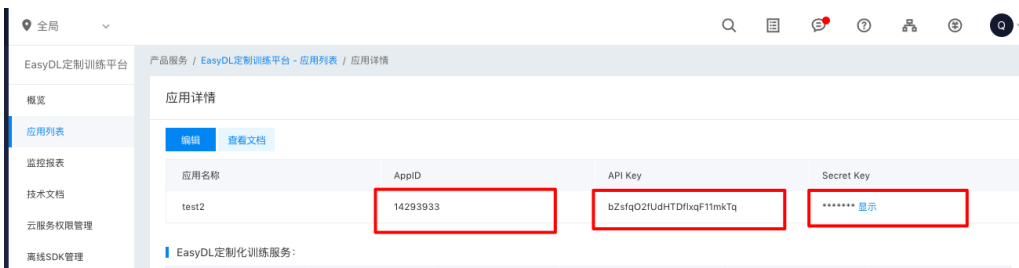
基于自定义训练出的图像分割模型，实现定制图像识别。

接口鉴权

1、在EasyDL控制台创建应用



2、应用详情页获取AK SK



请求说明

请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后申请上线，上线成功后可在服务列表中查看并获取url。

URL参数：

参数	值
input_type	当取值为 url 时，需在请求参数中传入图片的URL string
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px，通用算法训练的模型发布API支持jpg/png/bmp格式。肺炎CT影像识别专用算法训练发布的API在此基础上支持dicom格式，限制4M以内。注意请去掉头部
threshold	否	number	-	默认值为推荐阈值，请在我的模型列表-模型效果查看推荐阈值
url	否	string	-	如果在请求URL参数中增加“input_type=url”，则该参数必传，否则“image”参数必传。参数内容为URL string，用户需确保该string是有效的图片URL，否则会下载失败

请求代码示例

```
Python3

"""
EasyDL 物体检测 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标图片的 本地文件路径，支持jpg/png/bmp格式**
IMAGE_FILEPATH = "【您的测试图片地址，例如：./example.jpg】"
```

返回说明

实例分割返回参数

字段	类型	说明
log_id	number	唯一的log id, 用于问题定位
results	array(object)	识别结果数组
+name	string	分类名称
+score	number	置信度
+location		
++left	number	检测到的目标主体区域到图片左边界的距离
++top	number	检测到的目标主体区域到图片上边界的距离
++width	number	检测到的目标主体区域的宽度
++height	number	检测到的目标主体区域的高度
+mask	string	基于游程编码的字符串, 编码内容为和原图宽高相同的布尔数组: 若数组值为0, 代表原图此位置像素点不属于检测目标, 若为1, 代表原图此位置像素点属于检测目标。 <a href="#">下载解码SDK</a>

#### 语义分割返回参数

字段	类型	说明
log_id	number	唯一的log id, 用于问题定位
results	array(object)	识别结果数组
+name	string	分类名称
+mask	string	基于游程编码的字符串, 编码内容为和原图宽高相同的布尔数组: 若数组值为0, 代表原图此位置像素点不属于检测目标, 若为1, 代表原图此位置像素点属于检测目标。 <a href="#">下载解码SDK</a>

#### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#), 用于帮助开发者在线调试接口, 查看在线调用的请求内容和返回结果、复制和下载示例代码等功能, 简单易用。

#### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如Access Token失效返回:

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 本地服务器部署

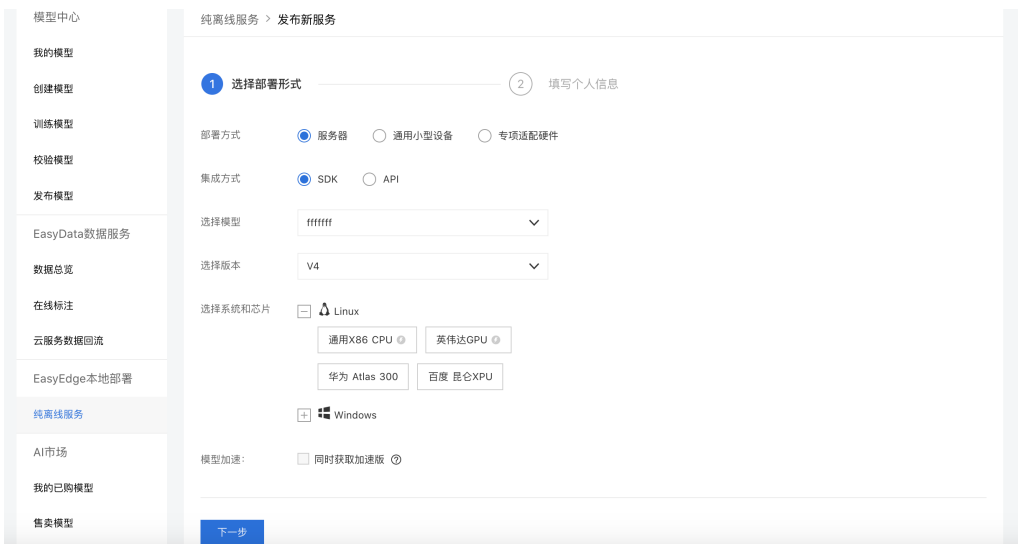
### 如何在服务器部署

训练完毕后，可以选择将模型通过「纯离线服务」或「端云协同服务」部署，具体介绍如下：

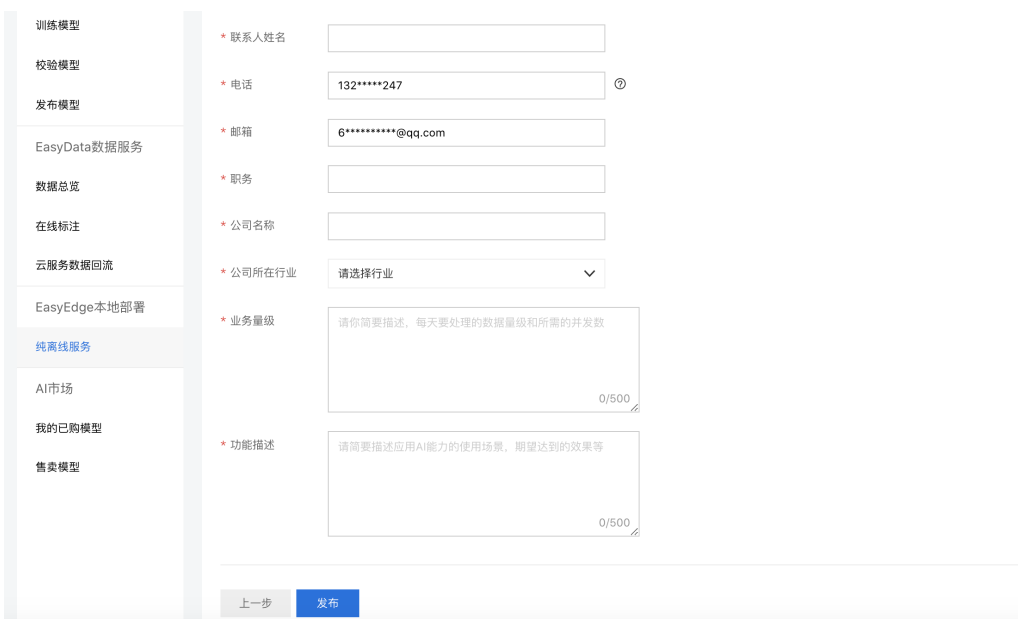
#### 纯离线服务部署

可以在左侧导航栏中找到「纯离线服务」，依次进行以下操作即可将模型部署到本地服务器：

- 选择部署方式「服务器」
- 选择集成方式
- 选择模型、版本、系统和芯片
- 点击下一步



- 填写部分信息（注：个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用）
- 点击发布



### ① 私有API

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

点击「发布」后，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

### ② 服务器端SDK

将模型封装成适配本地服务器（支持Linux和Windows）的SDK，可集成在其他程序中运行。首次联网激活后即可纯离线运行，占用服务器资源更少，使用方法更灵活

1. 点击「发布」后，前往[控制台](#)申请服务器端SDK的试用序列号
2. 点击「新增测试序列号」，根据模型类型选择「序列号类型」，填写「新增设备数」（所得序列号数量），点击确定即可



3. 离线SDK的激活和使用，请[参考文档](#)完成集成



### 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

### 本地服务器部署价格说明

EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。

如需购买永久使用授权，服务器SDK用户请在[控制台](#)点击「购买正式授权」，并按照对应步骤激活。

服务器API用户请微信搜索“BaiduEasyDL”添加小助手咨询，通过线下签订合同购买使用。

### 更多参考

[EasyDL官网入口](#)

[EasyDL开发文档](#)

[纯离线SDK说明](#)

[纯离线SDK简介](#)

本文档主要说明定制化模型发布后获得的服务器端SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### SDK说明

图像分割服务器端SDK支持Linux、Windows两种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux		CPU: x86_64 NVIDIA GPU: x86_64
Windows	64位 Windows7 及以上	NVIDIA GPU: x86_64  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015  GPU依赖： CUDA 9.x + cuDNN 7.x

### 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)页面查看评测信息表。

### 激活&使用步骤

离线SDK的激活与使用分以下三步：

- ① 下载SDK后，在[控制台](#)获取序列号
- ② 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

- ③ 正式使用

### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在[百度智能云控制台](#)内[提交工单](#)反馈

#### 1、激活失败怎么办？

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在[百度智能云控制台](#)内[提交工单](#)反馈

## Windows集成文档

### 简介

本文档介绍图像分割服务器端Windows SDK的使用方法。

- 硬件支持：
  - NVIDIA GPU（普通版，加速版）
- 操作系统支持
  - 64位 Windows 7 及以上
  - 64位 Windows Server 2012及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- GPU基础版（EasyEdge-win-x86-nvidia-gpu）依赖（必须安装以下版本）
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib：<http://www.winimage.com/zLibDll/zlib123dllx64.zip>，解压后将dll\_x64/zlibwapi.dll 拷贝到cuda的bin目录下) + 硬件计算能力(<https://developer.nvidia.com/cuda-gpus#compute>)达6.1及以上

- CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + 硬件计算能力达7.5及以上
- GPU加速版 (EasyEdge-win-x86-nvidia-gpu-tensorrt) 依赖 (必须安装以下版本)
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.x.x
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.x.x
- GPU加速版 (EasyEdge-win-x86-nvidia-gpu-paddletrt) 依赖 (必须安装以下版本)
  - CUDA 11.0.x + cuDNN 8.4.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.4.3.1 + 硬件计算能力达6.1及以上
  - CUDA 12.0.x + cuDNN 8.9.x(注意参照英伟达安装文档安装Zlib) + TensorRT 8.6.1.6 + 硬件计算能力达7.5及以上
- GPU加速版 (x86-nvidia-gpu-torch)
  - CUDA 11.0.x + cuDNN 8.0.5.x
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级, 修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级, 下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 优化模型算法; GPU CUDA9.0 CUDA10.0 标记为待废弃状态 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | GPU基础版推理引擎优化升级; GPU加速版支持自定义模型文件缓存路径; demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | 修复已知问题 | | 2021-08-19 | 1.3.2 | 新增支持EasyDL小目标检测, 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择; 目标追踪支持x86平台的GPU及加速版; 展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | 修复已知问题 | | 2021-01-27 | 1.2.1 | 新增模型支持; 性能优化; 问题修复 | | 2020-12-18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020-10-29 | 1.1.19 | 修复已知问题 | | 2020-09-17 | 1.1.18 | 支持更多模型 | | 2020.08.11 | 1.1.17 | 支持专业版更多模型 | | 2020.06.23 | 1.1.16 | 支持专业版更多模型 | | 2020.05.15 | 1.1.15 | 更新加速版tensorrt版本, 支持高精度检测 | | 2020.03.13 | 1.1.14 | 支持声音分类 | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | 支持物体检测高精度算法的CPU加速版, EasyDL 专业版支持 SDK 加速版 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版! |

## 快速开始

### 1. 安装依赖

#### 安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

#### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

#### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

#### 如果使用GPU版SDK, 请安装CUDA + cuDNN

<https://developer.nvidia.com/cuda>  
<https://developer.nvidia.com/cudnn>

#### 如果使用GPU版加速版SDK, 请安装TensorRT



<https://developer.nvidia.com/tensorrt>

根据cuda版本下载，下载后把lib目录下的所有dll，拷贝到SDK的dll目录下

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，SDK默认使用cuda9版本，如果需要cuda10请运行EasyEdge CUDA10.0.bat切换到cuda10版本，之后打开EasyEdge.exe，输入Serial Num，选择鉴权模式，点击“启动服务”，等待数秒即可启动成功，本地服务默认运行在

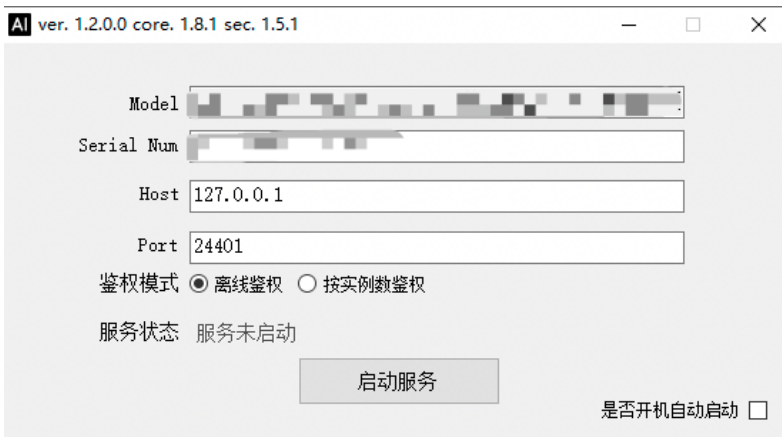
<http://127.0.0.1:24401/>

其他任何语言只需通过HTTP调用即可。

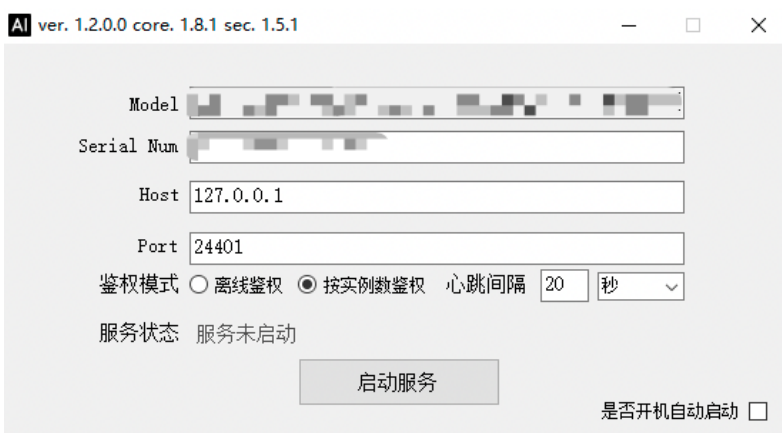
如启动失败，可参考如下步骤排查：



#### 2.1 离线鉴权（默认鉴权模式）首次联网激活，后续离线使用



#### 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

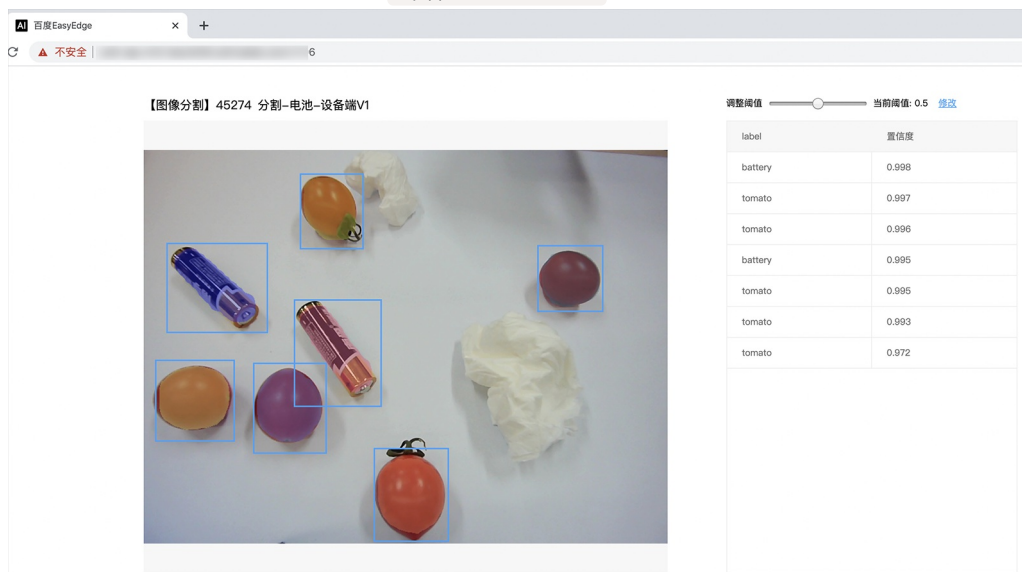
```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

2.3 序列号激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入http://127.0.0.1:24401，在h5中测试模型效果。



使用说明

调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img).json()
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**

**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----|-----| | confidence | float | 0~1 | 分割的置信度 | | label | string | | 分割的类别 | | index | number | | 分割的类别 | | mask | string | | 游程编码的mask | 代码参考 <https://github.com/Baidu-AIP/EasyDL-Segmentation-Demo>

### 集成指南

#### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

#### 基于c++ dll集成

#### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

### 集成方法

参考src目录中的CMakeLists.txt进行集成

### 基于c# dll集成

### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

### FAQ

#### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
 .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

GPU依赖，版本必须如下：  
 \* CUDA 11.0.x + cuDNN 8.4.x 或者 CUDA 11.7.x + cuDNN 8.4.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖，版本必须如下：  
 \* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.x.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖，版本必须如下：  
 \* CUDA 11.0.x + cuDNN 8.4.x + TensorRT 8.4.3.1

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

#### 4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

#### 7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## Linux集成文档-C++

### 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：图像分类，物体检测，图像分割，目标追踪
- 硬件支持：
  - CPU 基础版：- intel x86\_64 \* - AMD x86\_64 - 龙芯 loongarch64 - 飞腾 aarch64

- CPU 加速版 - Intel Xeon with Intel®AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE - AMD Core Processors with AVX2
- NVIDIA GPU: x86\_64 PC
- 寒武纪 Cambricon MLU270
- 比特大陆计算卡SC5+
- 百度昆仑XPU K200
  - x86\_64 - 飞腾 aarch64 - 百度昆仑XPU R200
  - x86\_64 - 飞腾 aarch64
- 华为Atlas 300
- 海光DCU: x86\_64 PC
- 寒武纪 MLU370 on x86\_64
- 操作系统支持：Linux

根据开发者的选择，实际下载的版本可能是以下版本之一：

- EasyDL图像
  - x86 CPU 基础版
  - x86 CPU 加速版
  - Nvidia GPU 基础版
  - Nvidia GPU 加速版
  - x86 mlu270基础版
  - x86 SC5+基础版
  - Phytium MLU270基础版
  - Phytium XPU基础版
  - Phytium Atlas300I基础版
  - Hygon DCU基础版

性能数据参考[算法性能及适配硬件](#)

\*intel 官方合作，拥有更好的适配与性能表现。

#### Release Notes

时间	版本	说明
2023.0 8.31	1.8.3	Atlas系列Soc支持语义分割模型，Atlas Cann升级到6.0.1，昆仑XPU后端推理引擎升级
2023.0 6.29	1.8.2	模型压缩能力升级
2023.0 5.17	1.8.1	支持物体检测自定义四边形模型精度无损压缩发布x86 CPU版SDK
2023.0 3.16	1.8.0	支持图像分类精度提升包本地部署
2022.1 2.29	1.7.2	模型性能优化；推理库性能优化
2022.1 0.27	1.7.1	新增语义分割模型http请求示例；升级海光DCU SDK，需配套rocm4.3版本使用；Linux GPU基础版下线适用于CUDA10.0及以下版本的SDK；Linux GPU加速版升级推理引擎版本

2022.0 9.15	1.7.0	Linux GPU加速版升级预测引擎；Linux GPU加速版适用于CUDA9.0、CUDA10.0的SDK为deprecated，未来移除；新增实例分割高性能模型离线部署；性能优化
2022.0 7.28	1.6.0	Linux CPU普通版、Linux GPU普通/加速版、Jetson新增目标追踪模型接入实时流的demo
2022.0 5.27	1.5.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2022.0 5.18	1.5.0	GPU加速版max_batch_size参数含义变更；修复GPU加速版并发预测时部分图片结果预测错误及耗时增加问题；CPU普通版预测引擎升级；新增版本号头文件；新增飞腾Atlas300I支持，并且在EasDL新增多种加速版本；示例代码移除frame_buffer，新增更安全高效的safe_queue；新增Tensor In/Out接口和Demo
2022.0 4.25	1.4.1	EasyDL, BML升级支持paddle2模型
2022.0 3.25	1.4.0	新增支持海光服务器搭配海光DCU加速卡；
2021.1 2.22	1.3.5	GPU加速版支持自定义模型文件缓存路径；新增支持飞腾MLU270服务器、飞腾XPU服务器
2021.1 0.20	1.3.4	CPU加速版推理引擎优化升级，新增支持飞腾CPU、龙芯CPU服务器、比特大陆计算卡SC5+ BM1684、寒武纪MLU270；大幅提升EasyDL GPU加速版有损压缩加速模型的推理速度
2021.0 8.19	1.3.2	CPU、GPU普通版及无损加速版新增支持EasyDL小目标检测，CPU普通版、GPU普通版支持检测模型的batch预测
2021.0 6.29	1.3.1	CPU普通版、GPU普通版支持分类模型的batch预测，CPU加速版支持分类、检测模型的batch预测；GPU加速版支持CUDA11.1；视频流解析支持调整分辨率；预测引擎升级
2021.0 5.13	1.3.0	新增视频流接入支持；模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告
2021.0 3.09	1.2.1	GPU新增目标追踪支持，http server服务支持图片通过base64格式调用，EasyDL高性能检测模型和均衡检测模型CPU加速版新增量化压缩模型
2021.0 1.27	1.1.0	EasyDL经典版分类高性能模型升级；部分SDK不再需要单独安装OpenCV
2020.1 2.18	1.0.0	1.0版本发布！安全加固升级、性能优化、引擎升级、接口优化等多项更新
2020.1 1.26	0.5.8	EasyDL经典版分类模型CPU加速版里新增量化压缩模型
2020.1 0.29	0.5.7	新增CPU加速版支持：EasyDL经典版高精度、超高精度物体检测模型和EasyDL经典版图像分割模型
2020.0 9.17	0.5.6	性能优化，支持更多模型
2020.0 8.11	0.5.5	提升预测速度；支持百度昆仑芯片
2020.0 5.15	0.5.3	优化性能，支持专业版更多模型
2020.0 4.16	0.5.2	支持CPU加速版；CPU基础版引擎升级；GPU加速版支持多卡多线程
2020.0 3.12	0.5.0	x86引擎升级；更新本地http服务接口；GPU加速版提速，支持批量图片推理
2020.0 1.16	0.4.7	ARM引擎升级；增加推荐阈值支持
2019.1 2.26	0.4.6	支持海思NNIE
2019.1 1.02	0.4.5	移除curl依赖；支持自动编译OpenCV；支持EasyDL 专业版 Yolov3；支持EasyDL经典版高精度物体检测模型升级
2019.1		

2019.1 0.25	0.4.4	ARM引擎升级,性能提升30%;支持EasyDL专业版模型
2019.0 9.23	0.4.3	增加海思NNIE加速芯片支持
2019.0 8.30	0.4.2	ARM引擎升级;支持分类高性能与高精度模型
2019.0 7.25	0.4.1	引擎升级,性能提升
2019.0 7.25	0.4.0	支持Xeye,细节完善
2019.0 6.11	0.3.3	paddle引擎升级;性能提升
2019.0 5.16	0.3.2	新增NVIDIA GPU支持;新增armv7l支持
2019.0 4.25	0.3.1	优化硬件支持
2019.0 3.29	0.3.0	ARM64 支持;效果提升
2019.0 2.20	0.2.1	paddle引擎支持;效果提升
2018.1 1.30	0.1.0	第一版!

2022-5-18:【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE含义变更。变更前:预测输入图片数不大于该值均可。变更后:预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理,在输入图片数小于该值时依然可正常运行,但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值,尽可能保持最小。

2020-12-18:【接口升级】参数配置接口从1.0.0版本开始已升级为新接口,以前的方式被置为deprecated,并将在未来的版本中移除。请尽快考虑升级为新的接口方式,具体使用方式可以参考下文介绍以及demo工程示例,谢谢。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES。更新模型时,如果SDK版本号有更新,请务必同时更新SDK,旧版本的SDK可能无法正确适配新发布出来部署包中的RES模型。

## 快速开始

SDK在以下环境中测试通过

- x86\_64, Ubuntu 16.04, gcc 5.4
- x86\_64, Ubuntu 18.04, gcc 7.4
- Tesla P4, Ubuntu 16.04, cuda 9.0, cudnn 7.5
- x86\_64, Ubuntu 16.04, gcc 5.4, XTCL r1.0
- aarch64, Kylin V10, gcc 7.3
- loongarch64, Kylin V10, gcc 8.3
- Bitmain SC5+ BM1684, Ubuntu 18.04, gcc 5.4
- x86\_64 MLU270, Ubuntu 18.04, gcc 7.5
- phytium MLU270, Kylin V10, gcc 7.3.0
- phytium XPU, Kylin V10, gcc 7.3.0
- hygon DCU, CentOS 7.8 gcc 7.3.0



- XPU K200, x86\_64, Ubuntu 18.04
- XPU K200 aarch64, Ubuntu 18.04
- XPU R200, x86\_64, Ubuntu 18.04
- XPU R200 aarch64, Ubuntu 18.04
- MLU370, x86\_64, Centos7.6.1810

#### 依赖包括

- cmake 3+
- gcc 5.4 (需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.11 (可选)
- cuda && cudnn (使用NVIDIA-GPU时必须, SDK内提供多个Cuda版本推理套件, 根据需要安装依赖的Cuda和Cudnn版本)
- XTCL 1.0.0.187 (使用昆仑服务器时必须)
- Rocm4.3, Miopen 2.14(使用海光DCU服务器时必须)

### 1. 安装依赖

以下步骤均可选, 请开发者根据实际运行环境选择安装。

#### (可选) 安装cuda&cudnn

##### 在NVIDIA GPU上运行必须(包括GPU基础版, GPU加速版)

对于GPU基础版, 若开发者需求不同的依赖版本, 请在[PaddlePaddle官网](#) 下载对应版本的libpaddle\_fluid.so或参考其文档进行编译, 覆盖lib文件夹下的相关库文件。

#### (可选) 安装TensorRT

##### 在NVIDIA GPU上运行GPU加速版必须

下载包中提供了对应 cuda9.0、cuda10.0、cuda10.2、cuda11.0+四个版本的 SDK, cuda9.0 和 cuda10.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.0.0.11, cuda10.2 及以上的 SDK 默认依赖的 TensorRT 版本为 TensorRT8.4, 请在[这里](#)下载对应 cuda 版本的 TensorRT, 并把其中的lib文件拷贝到系统lib目录, 或其他目录并设置环境变量。

(可选) 安装XTCL 使用昆仑服务器及对应SDK时必须 请安装与1.0.0.187版本兼容的XTCL。必要时, 请将运行库路径添加到环境变量。

#### (可选) 安装Rocm、Miopen

##### 使用海光DCU服务器对应SDK时必须

海光DCU SDK依赖Rocm 4.3和Miopen 2.14版本, 推荐使用easyedge镜像

(registry.baidubce.com/easyedge/hygon\_dcu\_infer:1.0.2.rocm4.3), SDK镜像内运行, 镜像拉取方式(wget https://aipe-easyedge-public.bj.bcebos.com/dcu\_docker\_images/hygon\_dcu\_rocm4.3.tar.gz && docker load -i hygon\_dcu\_rocm4.3.tar.gz), 关于海光DCU使用更多细节可参考[paddle文档](#)

### 2. 使用序列号激活 请在官网获取序列号

**纯离线服务说明**

发布纯离线服务, 将训练完成的模型部署在本地, 离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体方案专项适配硬件上。通过API, SDK进一步集成, 灵活适应不同业务场景。

[发布前准备](#) [控制台](#)

---

**服务器** 通用小型设备 专项适配硬件

[SDK](#) [API](#)

此处发布、下载的SDK为本授权SDK, 需要前往控制台[获取序列号](#)激活后才能正常使用。SDK内附有对应版本的Demo及开发文档, 开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
su_x小目标检测	134319-V1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

SDK内bin目录下提供预编译二进制文件, 可直接运行(二进制运行详细说明参考下一小节), 用于图片推理和模型http服务, 在二进制参数的

serial\_num(或者serial\_key)处填入序列号可自动完成联网激活（请确保硬件首次激活时能够连接公网，如果确实不具备联网条件，需要使用纯离线模式激活，请下载使用百度智能边缘控制台纳管SDK）

```
**SDK内提供的一些二进制文件，填入序列号运行可自动完成激活，以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}
```

如果是基于源码集成，设置序列号方法如下

```
global_controller()->set_licence_key("")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式（请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量或者源码设置）实例数鉴权环境变量设置方法

```
export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30
```

实例数鉴权源码设置方法

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)
```

3. 基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；在Intel CPU上运行CPU加速版，如果thirdparty里包含openvino文件夹的，必须在编译或运行demo程序前执行以下命令：source \${cpp\_kit位置路径}/thirdparty/openvino/bin/setupvars.sh 或者执行 source \${cpp\_kit位置路径}/thirdparty/openvino/setupvars.sh(openvino-2022.1+) 如果SDK内不包含setupvars.sh脚本，请忽略该提示

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号尽首次激活需要使用，激活后可不用填序列号也能运行二进制)

```
**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=./lib ./easyedge_image_inference ../.././RES /xxx/cat.jpeg "1111-1111-1111-1111"
```

demo运行效果：



图片加载失败

```
> ./easyedge_image_inference ../.././RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

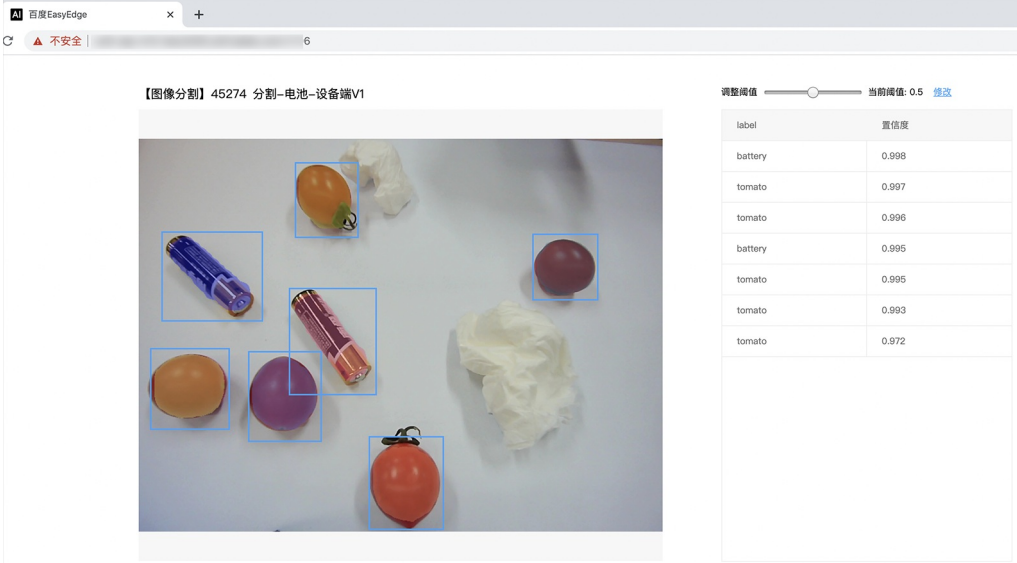
```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=./lib ./edgekit_serving --cfg=./edgekit_serving.yml

**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=./lib ./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

HTTP(or Webservice) is now serving at 0.0.0.0:24401

字样，此时，开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试，网页右侧会展示模型推理结果



label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

对于目标追踪的模型，请选择一段视频，并耐心等待结果



图片加载失败

同时，可以调用HTTP接口来访问服务。

**请求http服务** 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

python示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }

        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。编译demo项目 SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```

cd src
mkdir build && cd build
cmake .. && make
./easymage_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easymage_image_inference {模型RES文件夹} {测试图片路径}

```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEDGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```
// step 1: 配置模型资源目录
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 在这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}
}
```

## 输入图片不限制大小

**SDK参数配置** SDK的参数通过EdgePredictorConfig::set\_config和global\_controller()->set\_config配置。set\_config的所有key在easyedge\_xxxx\_config.h中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过EdgePredictorConfig::set\_config设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过global\_controller()->set\_config设置

以序列号为例，KEY的说明如下：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";
```

使用方法如下：

```
EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");
```

具体支持的运行参数配置列表可以参考开发工具包中的头文件的详细说明。

相关配置均可以通过环境变量的方法来设置，对应的key名称加上前缀EDGE\_即为环境变量的key。如序列号配置的环境变量key为EDGE\_PREDICTOR\_KEY\_SERIAL\_NUM，如指定CPU线程数环境变量key为EDGE\_PREDICTOR\_KEY\_CPU\_THREADS\_NUM。注意：通过代码设置的配置会覆盖通过环境变量设置的值。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image, std::vector<std::vector<EdgeResultData>>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测、图像分割时才有意义
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割的模型, 该字段才有意义
    // 请注意: 图像分割时, 以下两个字段会比较大, 使用完成之后请及时释放EdgeResultData
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask

    // 目标追踪模型, 该字段才有意义
    int trackid; // 轨迹id
    int frame; // 处于视频中的第几帧
    EdgeTrackStat track_stat; // 跟踪状态
};

```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

cv::Mat mask为图像掩码的二维数组

```

{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}

```

其中1代表为目标区域, 0代表非目标区域

### 关于图像分割mask\_rle

该字段返回了mask的游程编码, 解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding, 此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :



```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};       // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};             // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};      // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;        // frame存储为视频文件的路径
    bool save_all{false};        // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

1. 开启http服务 http服务的启动可以参考`demo_serving.cpp`文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

### 2. http接口详细说明

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片或视频来进行测试。

http 请求方式一：无额外编码 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img.json())
```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
```

Python请求示例 (视频测试, 注意: 区别于图片预测, 需指定Content-Type; 否则会调用图片推理接口)

```
import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data.json())
```

http 请求方法二: json格式, 图片传base64格式字符串 HTTP方法: POST Header如下:

参数	值
Content-Type	application/json

Body请求填写:

- 图像分类网络: body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据, base64编码, 要求base64图片编码后大小不超过4M,最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量, 不填该参数, 则默认返回全部分类结果

- 物体检测和实例分割网络: Body请求示例:

```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)

```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型，同其他CV模型不同，语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
        with open("gray_result.png", "wb") as fb:
            fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    },
  ]
}
```

其他配置

### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



### 2. CPU线程数设置

CPU线程数可通过 EdgePredictorConfig::set\_config配置

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_CPU_THREADS_NUM, 4);
```

### 3. 批量预测设置

```
int batch_size = 2; // 使用前修改batch_size再编译、执行
while (get_next_batch(imgs, img_files, batch_size, start_index)) {
  ...
}
```

**GPU 加速版 预测接口** GPU 加速版 SDK 除了支持上面介绍的通用接口外，还支持图片的批量预测，预测接口如下：

```

/**
 * @brief
 * GPU加速版批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& result
) = 0;

/**
 * @brief
 * GPU加速版批量图片推理接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;

```

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE`，其含义见下方参数配置接口的介绍。

**运行参数选项** 在上面的内容中我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对GPU加速版开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值：4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值：false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1：如果当前max_batch_size与历史编译产生的max_batch_size不相等时，则重新编译模型（推荐）
 * 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
 * 值类型: int
 * 默认值：1
 */

```

```

static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名, 默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置; 序列号不设置留空时, SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**: 首次加载模型会先对模型进行编译优化, 通过此值可以设置优化后的产出文件名, 这在多进程加载同一个模型的时候是有用的。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**: 首次加载模型经过编译优化后, 产生的优化文件会存储在这个位置, 可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**: 设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**: 此值用来控制批量图片预测可以支持的最大图片数, 实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**: 设置需要使用的 GPU 卡号。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**: 模型编译等级。通常模型的编译会比较慢, 但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制。当此值为 0 时, 表示忽略当前设置的 `max_batch_size` 而仅使用历史产出 (无历史产出时则编译模型); 当此值为 1 时, 会比较历史产出和当前设置的 `max_batch_size` 是否相等, 如不等, 则重新编译; 当此值为 2 时, 无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**: 通过此值设置单张 GPU 卡上可以支持的最大 `infer` 并发量, 其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源, 建议结合实际使用控制此值, 使用多少则设置多少。注意: 此值的增加会降低单次 `infer` 的速度, 建议优先考虑 `batch inference` 和 `multi predictor`。

**PREDICTOR\_KEY\_GTURBO\_FP16**: 默认是 `fp32` 模式, 置 `true` 可以开启 `fp16` 模式预测, 预测速度会有所提升, 但精度也会略微下降, 权衡使用。注意: 不是所有模型都支持 `fp16` 模式。目前已知不支持 `fp16` 的模型包括: 图像分类高精度模型。

**多线程预测** GPU 加速版 SDK 的多线程分为单卡多线程和多卡多线程两种。单卡多线程: 创建一个 `predictor`, 并通过

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY** 控制单卡所支持的最大并发量, 只需要 `init` 一次, 多线程调用 `infer` 接口。多卡多线程: 多卡的

支持是通过创建多个 predictor，每个 predictor 对应一张 GPU 卡，predictor 的创建和 init 的调用放在主线程，通过多线程的方式调用 infer 接口。

**已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时，部分结果错误** A：EasyDL图像分类高精度模型在有些显卡上可能存在此问题，可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

**2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object** A：部分显卡存在此问题，如果遇到此问题，请确认没有频繁调用 init 接口，通常调用 infer 接口即可满足需求。

**3. 开启 fp16 后，预测结果错误** A：不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括：图像分类高精度模型。目前不支持的将在后面的版本陆续支持。

**昆仑服务器** 昆仑服务器SDK支持将EasyDL的模型部署到昆仑服务器上。SDK提供的接口风格一致，简单易用，轻松实现快速部署。Demo的测试可参考上文中的测试Demo部分。

**参数配置接口** 在上面的内容中我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对昆仑服务器开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * 使用哪张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 设置需要同时预测的图片数量
 * 值类型：int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_KUNLUN_BATCH_SIZE = "PREDICTOR_KEY_KUNLUN_BATCH_SIZE";
```

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的加速卡的卡号。

**PREDICTOR\_KEY\_KUNLUN\_BATCH\_SIZE**：设置单次预测可以支持的图片数量。

使用方法：

```
int batch_size = 1;
config.set_config(easyedge::params::PREDICTOR_KEY_KUNLUN_BATCH_SIZE, batch_size);
```

**模型调优** 通过设置如下环境变量，可以在初始化阶段对模型调优，从而让预测的速度更快。

```
export XPU_CONV_AUTOTUNE=5
```

## FAQ

### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt



```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中, 其他需要的库视具体sdk中包含的库而定。

## 2. EasyDL SDK与云服务效果不一致, 如何处理?

后续我们会消除这部分差异, 如果开发者发现差异较大, 可联系我们协助处理。

## 3. NVIDIA GPU预测时, 报错显存不足 如以下错误字样:

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请根据显存大小和模型配置。调整合适的初始 fraction\_of\_gpu\_memory。参数的含义参考[这里](#)。

## 4. 如何将我的模型运行为一个http服务? 目前cpp sdk暂未集成http运行方式; 0.4.7版本之后, 可以通过start\_http\_server方法开启http服务。

## 5. 运行NNIE引擎报permission denied 日志显示:

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

## 6. 运行SDK报错 Authorization failed

情况一: 日志显示 Http perform failed: null respond 在新的硬件上首次运行, 必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二: 日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更, 包括 (但不局限于) 以下可能的情况:

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况, 请确保硬件无变更, 如果想更换序列号, 请先删除 ~/.baidu/easyedge 目录, 再重新激活。

## 7. 使用libcurl请求http服务时, 速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题, 添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

## 8. 运行二进制时, 提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath, 可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后, 再运行:

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 运行二进制时提示 libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 同上面8的问题类似, 没有正确设置动态库的查找路径, 可通过设置LD\_LIBRARY\_PATH为sdk的thirdparty/opencv/lib文件夹解决

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/thirdparty/opencv/lib
(tips: 上面冒号后面接的thirdparty/opencv/lib路径以实际项目中路径为准, 比如也可能是../thirdparty/opencv/lib)
```

10. 编译时报错: **file format not recognized** 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中, 再解压缩、编译

11. 进行视频解码时, 报错符号未找到、格式不支持、解析出的图片为空、无法设置抽帧 请确保安装OpenCV时, 添加了-DWITH\_FFMPEG=ON选项 (或者GStream选项), 并且检查OpenCV的安装日志中, 关于Video I/O段落的说明是否为YES。

```
-- Video I/O:
-- DC1394:      YES (ver 2.2.4)
-- FFMPEG:     YES
-- avcodec:    YES (ver 56.60.100)
-- avformat:   YES (ver 56.40.101)
-- avutil:     YES (ver 54.31.100)
-- swscale:    YES (ver 3.1.101)
-- avresample: NO
-- libv4l/libv4l2: NO
-- v4l/v4l2:   linux/videodev2.h
```

如果为NO, 请搜索相关解决方案, 一般为依赖没有安装, 以apt为例:

```
apt-get install yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
```

12. GPU加速版运行有损压缩加速的模型, 运算精度较标准模型偏低 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除, 并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true, 使用FP16的运算精度重新评估模型效果。若依然不理想, 可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false, 从而使用更高精度的FP32的运算精度。

## Linux集成文档-Python

### 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法, 适用于 EasyDL 和 BML。

EasyDL 通用版:

- 网络类型支持: 图像分类, 物体检测, 图像分割, 声音分类, 表格预测
- 硬件支持:
  - Linux x86\_64 CPU (基础版, 加速版)
  - Linux x86\_64 Nvidia GPU (基础版, 加速版)
- 语言支持: Python 3.5, 3.6, 3.7

BML:

- 网络类型支持: 图像分类, 物体检测, 声音分类
- 硬件支持:
  - Linux x86\_64 CPU (基础版)
  - Linux x86\_64 Nvidia GPU (基础版)
- 语言支持: Python 3.5, 3.6, 3.7

### Release Notes

时间	版本	说明
2023-03-16	1.3.7	迭代升级，新增支持文本类模型；新增GPU 多卡多进程推理demo
2022.10.27	1.3.5	新增华为Atlas300、飞腾Atlas300 Python SDK，支持图像分类、物体检测、人脸检测、实例分割
2022.09.15	1.3.3	EasyDL CPU普通版新增支持表格预测
2022.05.27	1.3.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2021.12.22	1.2.7	声音分类模型升级
2021.10.20	1.2.6	CPU基础版、CPU加速版、GPU基础版推理引擎优化升级
2021.08.19	1.2.5	CPU基础版、CPU无损加速版、GPU基础版新增支持EasyDL小目标检测
2021.06.29	1.2.4	CPU、GPU新增EasyDL目标跟踪支持；新增http server服务启动demo
2021.03.09	1.2.2	EasyDL CPU加速版新增支持分类、高性能检测和均衡检测的量化压缩模型
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.03.12	1.1.14	新增声音识别python sdk
2020.02.12	1.1.13	新增口罩模型支持
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持 SDK 加速版
2019.12.04	1.1.10	支持图像分割
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.08.29	1.1.8	CPU 加速版支持
2019.07.19	1.1.7	提供模型更新工具
2019.05.16	1.1.3	NVIDIA GPU 支持
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】 序列号的配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。
- 使用声音分类SDK需要安装额外依赖 \* pip 安装 `resampy pydub six librosa` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已基于sdk中无需额外安装，linux系统需要手动安装）
- 使用表格预测SDK需要安装额外依赖 `pip安装brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3 catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2 psutil==5.7.2 pymml==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0` 另外ml算法安装（目前只支持python3.7） `pip install BaiduAI_TabularInfer-0.0.0-cp37-cp37m-linux_x86_64.whl` 安装 **paddlepaddle**
- 使用x86\_64 CPU 基础版 预测时必须安装（目标跟踪除外）：

```
python -m pip install paddlepaddle==2.2.2 -i https://mirror.baidu.com/pypi/simple
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU 基础版预测时必须安装（目标跟踪除外）：

```
python -m pip install paddlepaddle-gpu==2.2.2.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA10.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2 -i https://mirror.baidu.com/pypi/simple #CUDA10.2的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post110 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.0的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post111 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post112 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.2的PaddlePaddle
```

不同cuda版本的环境，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持安装 OpenVINO 使用x86\_64 CPU 加速版 SDK 预测时必须安装。

1) 请参考 [OpenVINO toolkit 文档](#)安装 2021.4版本, 安装时可忽略Configure the Model Optimizer及后续部分

2) 运行之前，务必设置环境变量

```
source /opt/intel/opencvino_2021/bin/setupvars.sh
```

安装 cuda、cudnn

- 使用Nvidia GPU 加速版预测时必须安装。依赖的版本为 cuda9.0、cudnn7。版本号必须正确。

安装 pytorch (torch >= 1.7.0)

- 目标跟踪模型的预测必须安装pytorch版本1.7.0及以上（包含：Nvidia GPU 基础版、x86\_64 CPU 基础版）。
- 目标跟踪模型Nvidia GPU 基础版还需安装依赖cuda、cudnn。

关于不同版本的pytorch和CUDA版本的对应关系：[pytorch官网](#) 目标跟踪模型还有一些列举在requirements.txt里的依赖（包括torch >= 1.7.0），均可使用pip下载安装。

```
pip3 install -r requirements.txt
```

2. 安装 easyedge python wheel 包 安装说明

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。安装说明：[华为 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Atlas300-{版本号}-cp36-cp36m-linux_x86_64.whl
```

安装说明：[飞腾 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Phytium.Atlas-{版本号}-cp36-cp36m-linux_aarch64.whl
```

3. 使用序列号激活



## 获取序列号

此处发布、下载的SDK为未授权SDK，需要前往控制台获取序列号激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标test	134318-V1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英特尔GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

## 修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

## 4. GPU 加速版 使用 GPU 加速版，在安装完 whl 之后，必须：

1. 从[这里](#)下载 TensorRT7.0.0.11 for cuda9.0，并把解压后的 lib 放到 C++ SDK 的 lib 目录或系统 lib 目录
2. 运行时，必须在系统库路径中包含 C++ SDK 下的lib目录。如设置LD\_LIBRARY\_PATH

```
cd ${SDK_ROOT}
```

### \*\*1. 安装 python wheel 包\*\*

```
tar -xzf python/*.tar.gz
pip install -U {对应 Python 版本的 wheel 包}
```

### \*\*2. 设置 LD\_LIBRARY\_PATH\*\*

```
tar -xzf cpp/*.tar.gz
export EDGE_ROOT=$(readlink -f $(ls -h | grep "baidu_easyedge_linux_cpp"))
export LD_LIBRARY_PATH=$EDGE_ROOT/lib
```

### \*\*3. 运行 demo\*\*

```
python3 demo.py {RES文件夹路径} {测试图片路径}
```

如果是使用 C++ SDK 自带的编译安装的 OpenCV，LD\_LIBRARY\_PATH 还需要包括 C++ SDK 的 build 目录下的 `thirdparty/lib` 目录

如果没有正确设置 LD\_LIBRARY\_PATH，运行时可能报错：

```
ImportError: libeasyedge.so.0.4.3: cannot open shared object file: No such file or directory
ImportError: libopencv_core.so.3.4: cannot open shared object file: No such file or directory
```

## 5. 测试 Demo

### 5.1 图片预测

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



### 5.2 视频预测（适用于目标跟踪）

输入对应的模型文件夹（默认为RES）和测试视频文件路径 / 摄像头id / 网络视频流地址，运行：

```

**video_type: 输入源类型 type:int**
**1 本地视频文件**
**2 摄像头的index**
**3 网络视频流**
**video_src: 输入源地址, 如视频文件路径、摄像头index、网络流地址 type: string**
python3 demo.py {model_dir} {video_type} {video_src}

```

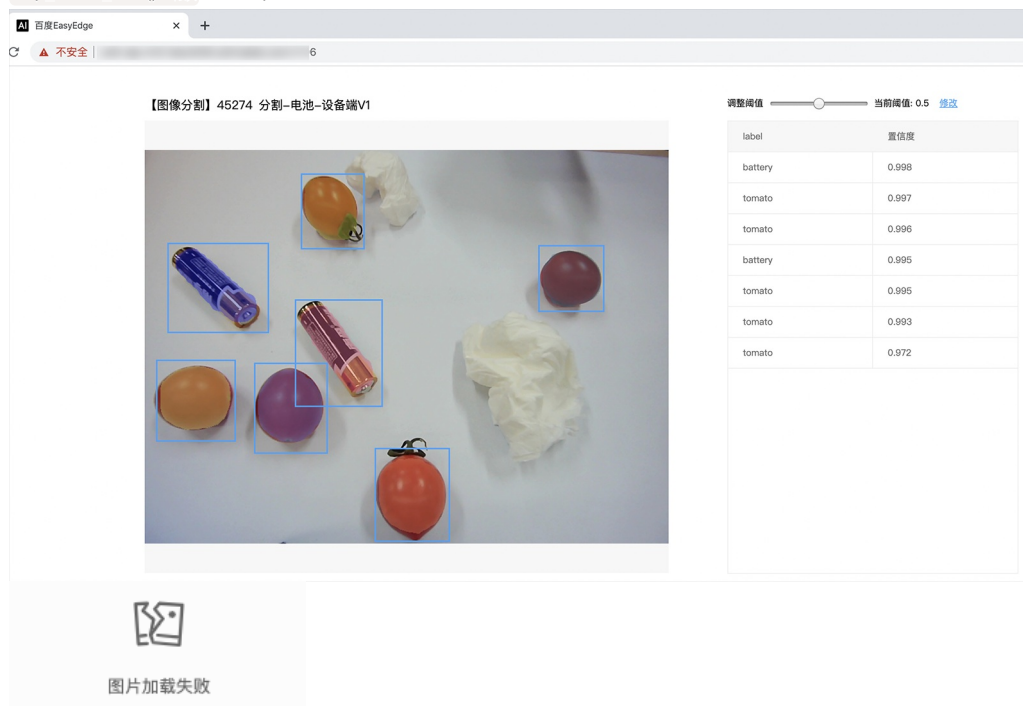
6. 测试Demo HTTP 服务 输入对应的模型文件夹（默认为RES）、序列号、设备ip和指定端口号，运行：

```
python3 demo_serving.py {model_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

后，会显示：

```
Running on http://0.0.0.0:24401/
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里 `http_client_test()`函数请求http服务进行推理。



## 使用说明

使用流程 `demo.py`

```

import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image((numpy.ndarray的图片))
pred.close()

```

`demo_serving.py`

```

import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir={RES文件夹路径}, license=serial_key)
**请参考同级目录下demo.py里:**
**pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)**
**对以下参数device\device_id和engine进行修改**
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)

```

## 初始化

- 接口

```
def init(self,
    model_dir,
    device=Device.CPU,
    engine=Engine.PADDLE_FLUID,
    config_file='conf.json',
    preprocess_file='preprocess_args.json',
    model_file='model',
    params_file='params',
    label_file='label_list.txt',
    infer_cfg_file='infer_cfg.json',
    device_id=0,
    thread_num=1
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device, 比如: Device.CPU
        engine: BaiduAI.EasyEdge.Engine, 比如: Engine.PADDLE_FLUID
        config_file: str
        preprocess_file: str
        model_file: str
        params_file: str
        label_file: str 标签文件
        infer_cfg_file: 包含预处理、后处理信息的文件
    device_id: int 设备ID
        thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success
    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

## 预测图像

- 接口

```
def infer_image(self, img,
                threshold=0.3,
                channel_order='HWC',
                color_format='BGR',
                data_type='numpy'):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测



```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

### 预测视频（目前仅限目标跟踪模型调用）

- 接口

```
def infer_frame(self, frame, threshold=None):
    """
    视频推理(抽帧之后)
    :param frame:
    :param threshold:
    :return:
    """
```

- 返回格式dict

字段	类型	说明
pos	dict1	当前帧每一个类别的追踪目标的像素坐标(tlwh)
id	dict2	当前帧每一个类别的追踪目标的id
score	dict3	当前帧每一个类别的追踪目标的识别置信度
label	dict4	class_idx(int)与label(string)的对应关系
class_num	int	追踪类别数

### 预测声音

- 使用声音分类SDK需要安装额外依赖 `pip 安装 resampy pydub` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已集成在sdk中无需额外安装，linux系统需要手动安装）

- 接口

```
def infer_sound(self, sound_binary,
                threshold=0.3):
    """

    Args:
        sound_binary: sound_binary
        threshold: confidence

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类的置信度
label	string		分类的类别
index	number		分类的类别

**升级模型** 适用于经典版升级模型，执行`bash update_model.sh`，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

## FAQ

**Q: EasyDL 离线 SDK 与云服务效果不一致，如何处理？** A: 后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**Q: 运行时报错 "非法指令" 或 "illegal instruction"** A: 可能是 CPU 缺少 `avx` 指令集支持，请在[paddle官网](#) 下载 `noavx` 版本覆盖安装

**Q: NVIDIA GPU预测时，报错显存不足：** A: 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过`export FLAGS_fraction_of_gpu_memory_to_use=0.3`来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 `allocate` 更多的显存。

**Q: 我想使用多线程预测，怎么做？** 如果需要多线程预测，可以每个线程启动一个Program实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的`thread_num`参数。

## Q: 运行SDK报错 Authorization failed

**情况一：日志显示 `Http perform failed: null respond`** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受`HTTP_PROXY` 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示`failed to get/check device id(xxx)`或者`Device fingerprint mismatch(xxx)`** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

情况三：Atlas Python SDK日志提示 `ImportError: libavformat.so.58: cannot open shared object file: No such file or directory` 或者其他类似so找不到 可以在 `LD_LIBRARY_PATH` 环境变量加上 `libs` 和 `thirdpartylibs` 路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内 `libs` 和 `thirdpartylibs` 路径的方法如下(以华为Atlas300 SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas300 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## 纯离线API集成说明

本文档主要说明定制化图像分割模型发布为本地服务器API（通过API部署包实现）后如何使用。如还未训练模型，请先前往[EasyDL](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

## 部署包使用说明

### 部署方法

EasyDL定制化图像分割模型的服务器API通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：`easyDL_服务名称_模型版本号`），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#) 使用 `python2` 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

### 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络联通性测试、容器关键报错日志输出等

使用方法：将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```
**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh
```

## 授权说明

部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

## 性能指标

图像分割模型可部署在CPU或GPU服务器上，单实例具体性能指标参见[算法性能及适配硬件](#)

## 接口描述

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请部署包，部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/ImageSegmentation](http://{IP}:{PORT}/{DEPLOY_NAME}/ImageSegmentation) IP：服务部署所在机器的ip地址 PORT：服务部署后获取的端口

DEPLOY\_NAME：申请时填写的服务名称

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
threshold	否	number	-	默认值为推荐阈值，请在我的模型列表-模型效果查看推荐阈值

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	识别结果数组
+name	否	string	分类名称
+score	否	number	置信度
+location	否		
++left	否	number	检测到的目标主体区域到图片左边界的距离
++top	否	number	检测到的目标主体区域到图片上边界的距离
++width	否	number	检测到的目标主体区域的宽度
++height	否	number	检测到的目标主体区域的高度
+mask	否	array	基于游程编码的字符串，编码内容为和原图宽高相同的布尔数组：若数组值为0，代表原图此位置像素点不属于检测目标，若为1，代表原图此位置像素点属于检测目标。 <a href="#">查看解码示例</a>

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
336005	图片解码失败	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度智能云控制台内 <a href="#">提交工单</a> 反馈
337000	Auth check failed	离线鉴权调用失败，

### 模型更新/回滚操作说明

#### 模型更新

1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。

两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 nvidia-smi命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如easycl\_\$(DEPLOY\_NAME)\_v2

\$(DEPLOY\_NAME) :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_$(DEPLOY_NAME)_v2
cd easedl_$(DEPLOY_NAME)_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh

**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/$(DEPLOY_NAME) /home/baidu/work/$(DEPLOY_NAME)_V1
**记录当前模型的端口号**
docker ps -a |grep $(DEPLOY_NAME)

**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：$(DEPLOY_NAME)，前面已备份**
python2 install.py remove $(DEPLOY_NAME)
**安装当前部署包内新的EasyDL服务：$(DEPLOY_NAME)**
python2 install.py install $(DEPLOY_NAME)

**(可选操作) 更新证书**
python2 install.py lu

```

## 模型回滚

以如下场景举例说明：[模型版本从V2回滚至V1](#)

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}

**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}

**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh

**（可选操作）进入V1版本部署包所在目录执行license更新操作，假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu
  
```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 端云协同服务说明

### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于[百度智能边缘](#)构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 联网状态下在平台管理设备运行状态、资源利用率

目前本地服务器的应用平台支持Linux-AMD64(x86-64)，具体使用流程请参考下方文档。

### 使用流程

#### Step 1 发布端云协同部署包

在[我的部署包](#)页面点击「发布端云协同部署包」

端云协同服务 > 我的部署包

**端云协同服务说明** [点击收起](#)

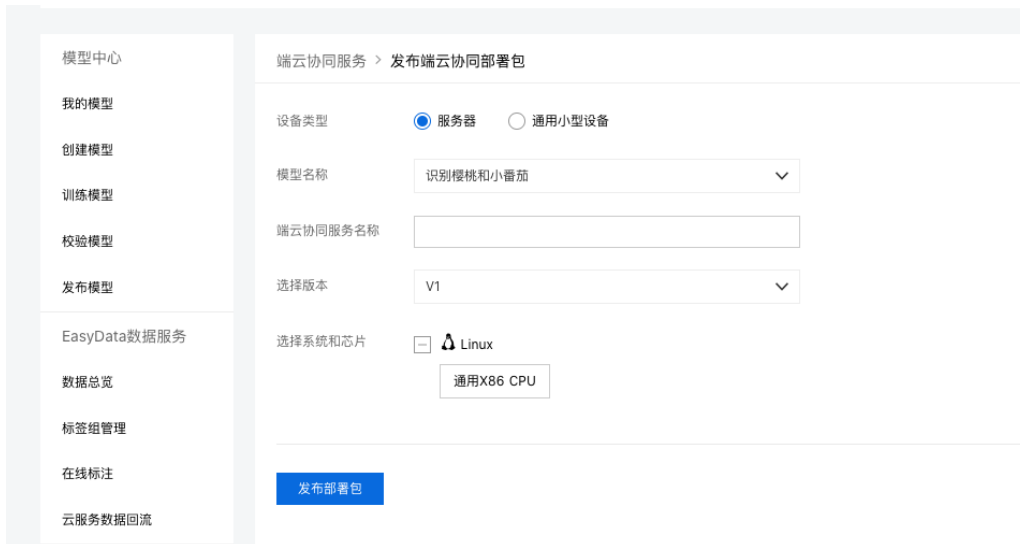
1. 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
2. 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
3. 联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

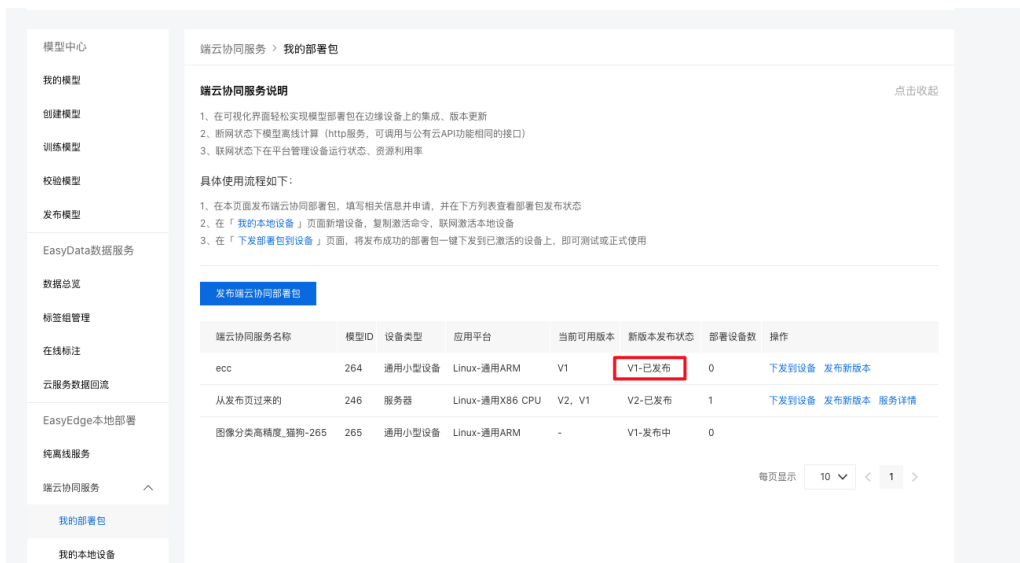
1. 在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
2. 在「我的本地设备」页面新增设备，复制激活命令，联网激活本地设备
3. 在「下发部署包到设备」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
暂无可用数据 请稍后再试							

填写服务名称，选择模型版本并提交发布



在列表查看部署包发布状态



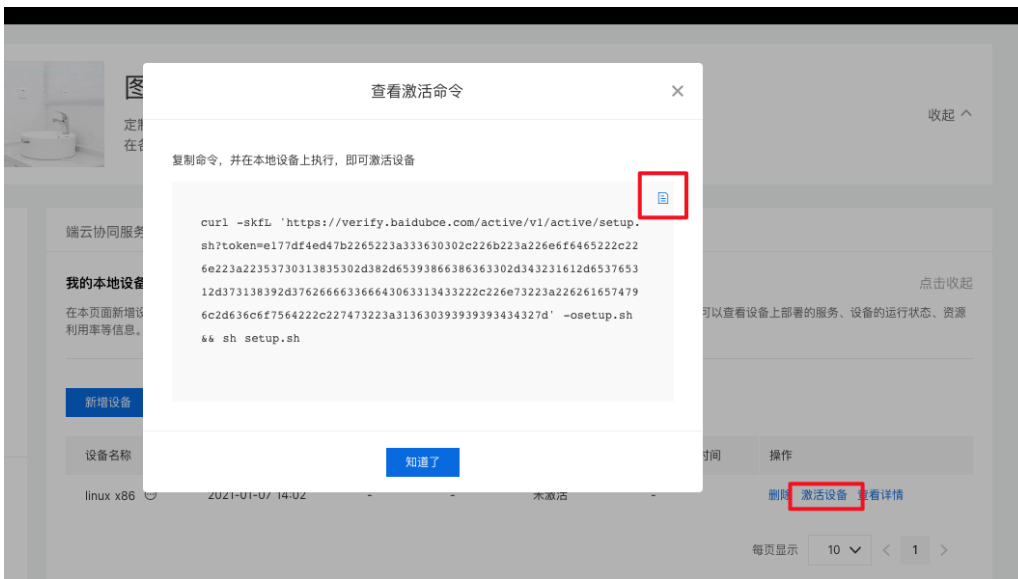
### Step 2 新增设备并激活

在**我的本地设备**页面新增设备



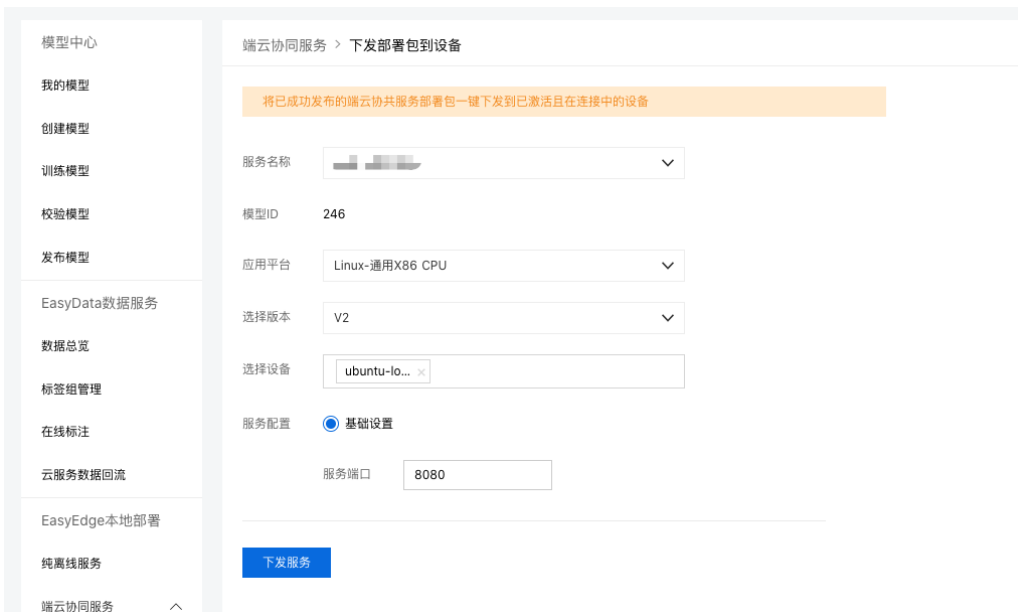


在列表中，点击设备对应的「激活设备」操作，复制激活命令并在本地设备上执行即可



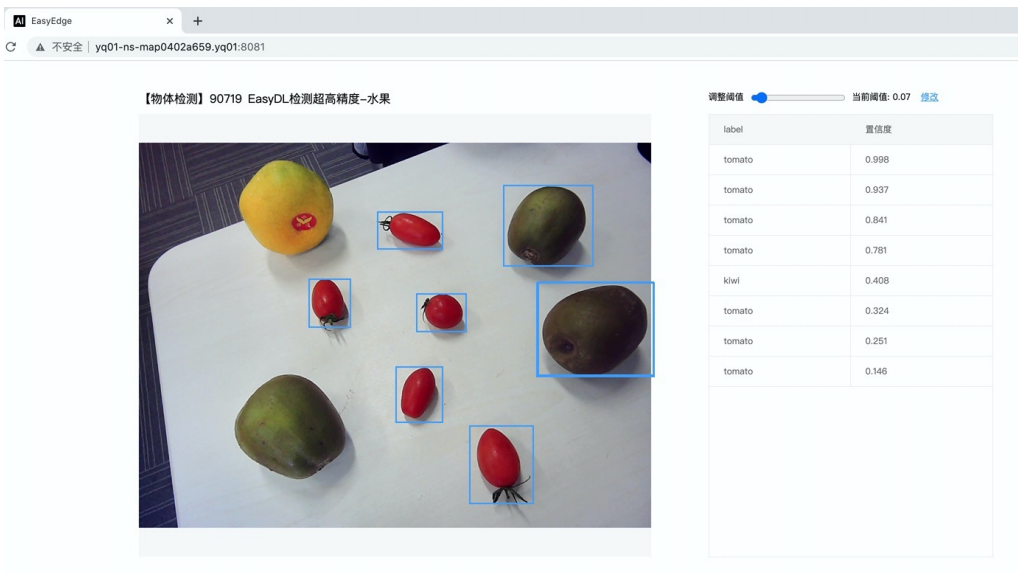
### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入http://{设备ip}:{服务端点，默认8080}，即可预览效果：





具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

### 云端管理说明

### 模型部署包管理

在[我的部署包](#)页面可以进行已发布的模型部署包的管理。

### 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。



新版本发布成功后，即可在「下发部署包到设备」页面或当前服务的「服务详情」页面，将新版本下发到本地设备上。



## 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。

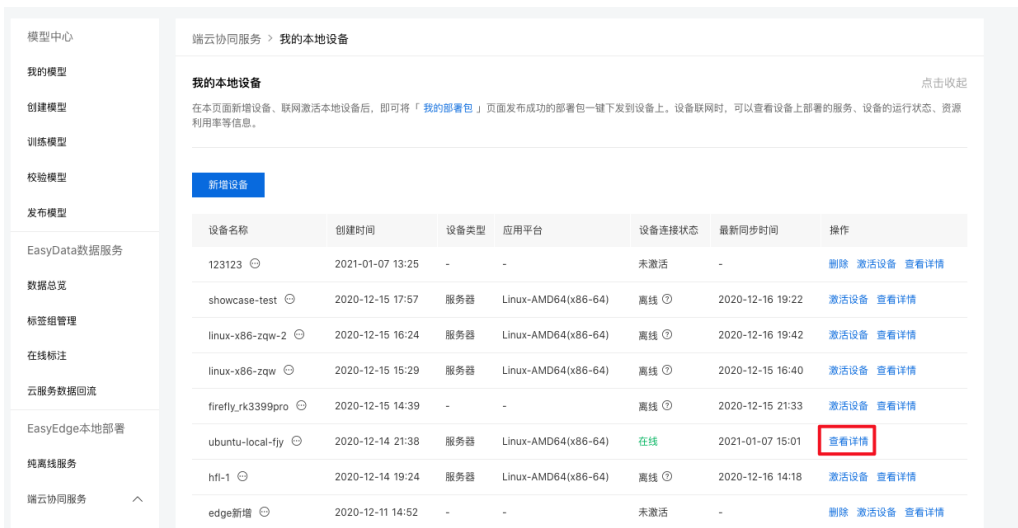


## 本地设备管理

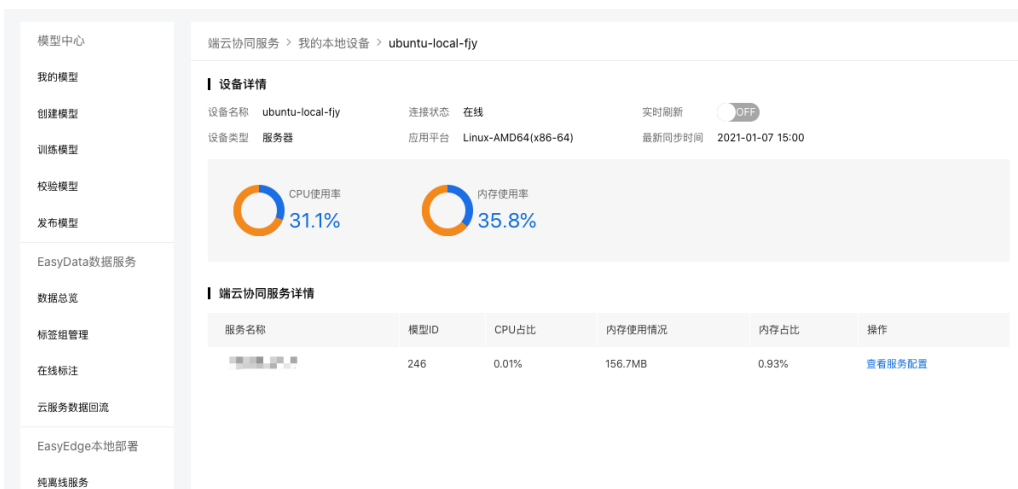
在[我的本地设备](#)页面可以进行所有本地设备的管理。

## 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：



设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况。



## 通用小型设备部署

### 如何在通用小型设备部署

训练完毕后，可以选择将模型通过「SDK-纯离线服务」或「API-端云协同服务」部署，具体介绍如下：

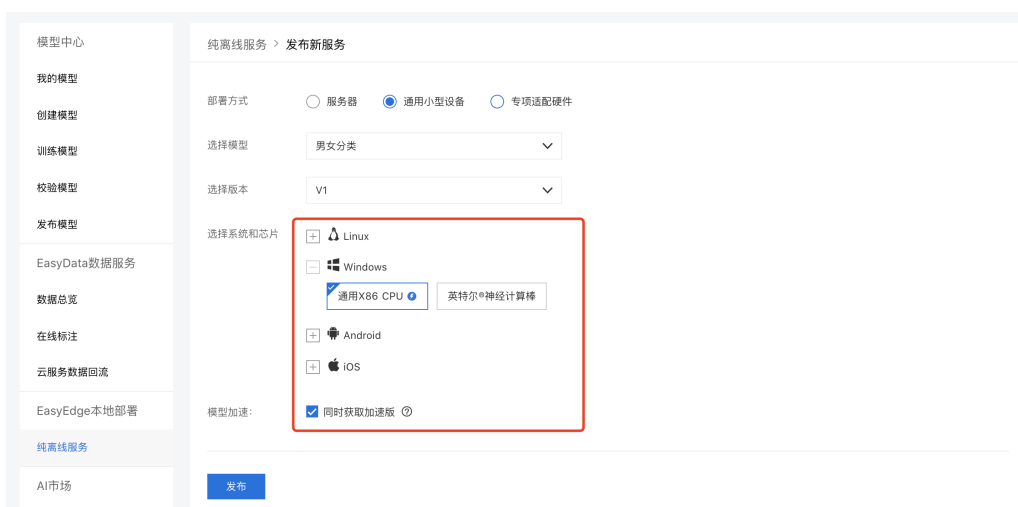
#### 纯离线服务部署

纯离线服务目前仅支持通过SDK集成，可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布设备端SDK：

- 选择模型
- 选择部署方式「EasyEdge本地部署」-「通用小型设备」
- 选择版本
- 选择集成方式
- 点击发布



- 再根据实际使用设备选择系统与芯片
- 点击发布



也可以直接在「EasyEdge本地部署」-「纯离线服务」页面点击发布新服务，按上图所述进行申请发布

## 端云协同服务部署

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

具体使用说明请参考[端云协同服务说明](#)

### 纯离线SDK说明

#### 纯离线SDK简介

本文档主要说明定制化模型发布后获得的SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

#### SDK说明

目前已支持Windows、Android、iOS、Linux四种操作系统，更多硬件支持敬请期待。

操作系统	系统支持	硬件环境要求
Windows	64位 Windows7 及以上	Intel CPU x86_64 环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015
Android	通用ARM: Android 19以上	绝大部分的手机和平板、比较耗时。支持armeabi-v7a arm-v8a CPU 架构
iOS	iOS 8.0 以上	ARMv7 ARM64 (Standard architectures) (暂不支持模拟器)
Linux		aarch64 armv7l

### 单次预测耗时参考

根据具体设备、线程数不同，数据可能有波动，请以实测为准

在[算法性能及适配硬件](#)页面查看评测信息表。

### 激活&使用SDK

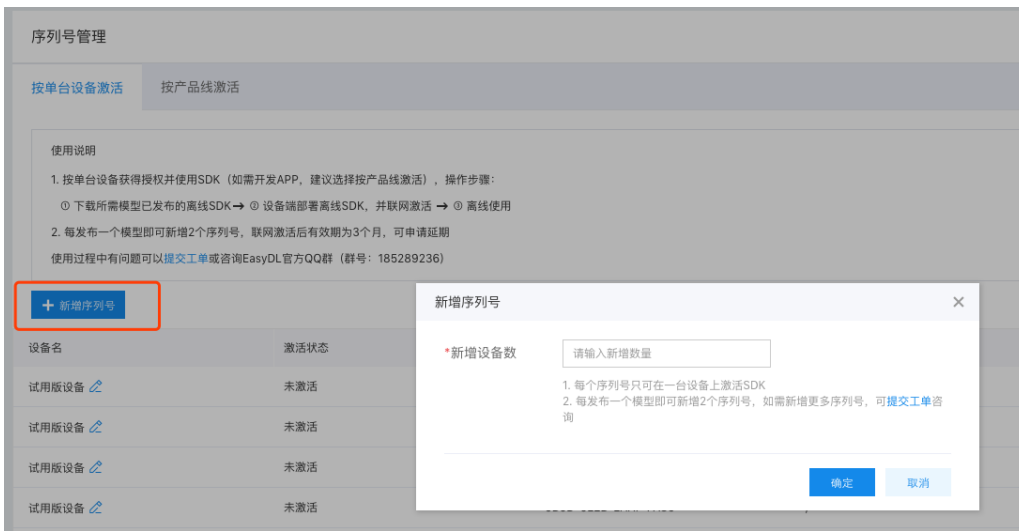
SDK的激活与使用分以下四步：

① 在【我的模型】-【服务详情】内下载SDK

The screenshot displays the '我的模型' (My Models) interface. On the left, a sidebar lists navigation options: 我的模型, 创建模型, 训练模型, 校验模型, 发布模型, 数据中心, 我的数据集, and 云服务调用数据. The main content area shows a list of models. The third model, 'cmc-高性能' (ID: 332), is selected. Its details are shown in a table with columns: 部署方式, 版本, 训练状态, 申请状态, 服务状态, 模型效果, and 操作. The '操作' column contains a red-bordered button labeled '服务详情' (Service Details).

② 在[控制台](#)获取序列号

按单台设备获得授权并使用SDK：



③ 本地运行SDK，并完成首次联网激活

④ 正式使用

### SDK常见问题

以下是通用FAQ，如您的问题仍未解决，请在百度智能云控制台内[提交工单](#)反馈

1、激活失败怎么办？

按设备激活时，激活失败可能由于以下几个原因造成：

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在百度智能云控制台内[提交工单](#)反馈

2、怎样申请序列号使用延期

序列号激活后有效期为三个月，可以在[控制台](#)进行申请，申请流程：

- 1) 填写申请信息
- 2) 等待审核：审核周期通常需要1-3个工作日左右，期间会有工作人员电话回访，请填写有效的联系方式并保证手机畅通

### Android集成文档

#### 简介

**1.1 Android SDK 硬件要求** Android 版本：支持 Android 5.0 (API 21) 及以上

硬件：支持 arm64-v8a 和 armeabi-v7a，暂不支持模拟器

通常您下载的SDK只支持固定的某一类芯片。

- **通用ARM**：支持大部分ARM 架构的手机、平板及开发板。**通常选择这个引擎进行推理。**
- **通用ARM GPU**：支持骁龙、麒麟、联发科等带GPU的手机、平板及开发板。
- **高端芯片AI加速模块**：
  - **高通骁龙引擎SNPE**：高通骁龙高端SOC，利用自带的DSP加速。其中 660 之后的型号可能含有 Hexagon DSP模块，具体列表见snpe高通骁龙引擎官网。

- **华为NPU引擎DDK**：华为麒麟980的arm-v8a的soc。具体手机机型为mate10，mate10pro，P20，mate20，荣耀v20等。
- **华为达芬奇NPU引擎DAVINCI**：华为NPU的后续版本，华为麒麟810，820，990，985的arm-v8a的soc。具体手机机型为华为mate30，p40，nova6，荣耀v30等。

**通用ARM**有额外的加速版，但是有一定的精度损失。

因GPU硬件限制，通用ARM GPU物体检测模型输入尺寸较大时会运行失败，可以在训练的时候将输入尺寸设为300\*300。

高端芯片AI加速模块，一般情况下推理速度较快。

运行内存不能过小，一般大于demo的assets目录大小的3倍。

### 1.2 功能支持 | 引擎 | 图像分类 | 物体检测 | 图像分割 | 文字识别

只支持EasyEdge | 姿态估计 | :: | :: | :: | :: | :: | :: | 通用ARM | √ | √ | √ | √ | √ | 通用ARM GPU | √ | √ | √ | √ | 高通骁龙引擎SNPE | √ | √  
 |||| | 华为NPU引擎DDK | √ | √ |||| | 华为达芬奇NPU引擎DAVINCI | √ | √ | √ |||

### 1.3 Release Notes

时间	版本	说明
2023.08.31	0.10.12	新增支持实例数鉴权；SNPE引擎升级；迭代优化
2023.06.29	0.10.11	迭代优化
2023.05.17	0.10.10	横屏兼容；迭代优化
2023.03.16	0.10.9	达芬奇NPU支持更多模型及语义分割模型；各芯片支持更多语义分割模型；精简版代码补充；迭代优化
2022.12.29	0.10.8	ARM / ARM-GPU 引擎升级；迭代优化
2022.10.27	0.10.7	达芬奇NPU新增适配麒麟985；迭代优化
2022.09.15	0.10.6	SNPE引擎升级；迭代优化
2022.07.28	0.10.5	迭代优化
2022.06.30	0.10.4	支持Android11；支持EasyEdge语义分割模型；迭代优化
2022.05.18	0.10.3	ARM / ARM-GPU 引擎升级；支持更多加速版模型发布；迭代优化
2022.03.25	0.10.2	ARM / ARM-GPU 引擎升级；支持更多检测模型；迭代优化
2021.12.22	0.10.1	DDK不再支持Kirin 970；迭代优化
2021.10.20	0.10.0	更新鉴权；更新达芬奇NPU、SNPE、通用ARM及ARM-GPU引擎；新增达芬奇NPU对检测模型的支持；支持更多姿态估计模型
2021.07.29	0.9.17	迭代优化
2021.06.29	0.9.16	迭代优化
2021.05.13	0.9.15	更新鉴权，更新通用arm及通用arm gpu引擎
2021.04.02	0.9.14	修正bug
2021.03.09	0.9.13	更新android arm的预处理加速
2020.12.18	0.9.12	通用ARM引擎升级；新增ARM GPU引擎
2020.10.29	0.9.10	迭代优化
2020.9.01	0.9.9	迭代优化
2020.8.11	0.9.8	更新ddk 达芬奇引擎
2020.7.14	0.9.7	支持arm版ocr模型，模型加载优化
2020.6.23	0.9.6	支持arm版fasterrcnn模型
2020.5.14	0.9.5	新增华为新的达芬奇架构np的部分图像分类模型
2020.4.17	0.9.4	新增arm通用引擎量化模型支持
2020.1.17	0.9.3	新增arm通用引擎图像分割模型支持
2019.12.26	0.9.2	新增华为kirin麒麟芯片的物体检测支持
2019.12.04	0.9.1	使用paddleLite作为arm预测引擎
2019.08.30	0.9.0	支持EasyDL专业版
2019.08.30	0.8.2	支持华为麒麟980的物体检测模型
2019.08.29	0.8.1	修复相机在开发版调用奔溃的问题
2019.06.20	0.8.0	高通手机引擎优化
2019.05.24	0.7.0	升级引擎
2019.05.14	0.6.0	优化demo程序
2019.04.12	0.5.0	新增华为麒麟980支持
2019.03.29	0.4.0	引擎优化，支持sd卡模型读取
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

## 快速开始

### 2.1 安装软件及硬件准备

扫描模型下载SDK处的网页上的二维码，无需任何依赖，直接体验



如果需要源码方式测试：

打开AndroidStudio，点击 "Import Project..."。在一台较新的手机上测试。

详细步骤如下：

1. 准备一台较新的手机，如果不是通用arm版本，请参见本文的“硬件要求”，确认是否符合SDK的要求
2. 安装较新版本的AndroidStudio，[下载地址](#)
3. 新建一个HelloWorld项目，Android Studio会自动下载依赖，在这台较新的手机上测试通过这个helloworld项目。注意不支持模拟器。
4. 解压下载的SDK。
5. 打开AndroidStudio，点击 "Import Project..."。即：File->New-> "Import Project..."，选择解压后的目录。
6. 此时点击运行按钮（同第3步），手机上会有新app安装完毕，运行效果和二维码扫描的一样。
7. 手机上UI界面显示后，如果点击UI界面上的“开始使用”按钮，可能会报序列号错误。请参见下文修改

## 2.2 使用序列号激活

如果使用的是EasyEdge的开源模型，无需序列号，可以跳过本段直接测试。

建议申请包名为"com.baidu.ai.easyaimobile.demo"的序列号用于测试。

本文假设已经获取到序列号，并且这个序列号已经绑定包名。

SDK默认使用离线激活方式，即首次联网激活，后续离线使用。SDK同时支持按实例数鉴权方式，即周期性联网激活，离线后会释放所占设备实例。按实例数鉴权的启用参考本节2.2.3说明

**2.2.1 填写序列号** 打开Android Studio的项目，修改MainActivity类的开头SERIAL\_NUM字段。 MainActivity 位于 app\src\main\java\com\baidu\ai\edge\demo\MainActivity.java文件内。

```
// 请替换为您的序列号
private static final String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX"; //这里填您的序列号
```

### 2.2.2 修改包名

如果申请的包名为"com.baidu.ai.easyaimobile.demo"，这个是demo的包名，可以不用修改

打开app/build.gradle文件，修改"com.baidu.ai.easyaimobile.demo"为申请的包名

```
defaultConfig {
    applicationId "com.baidu.ai.easyaimobile.demo" // 修改为比如"com.xxx.xxx"
}
```

修改序列号和包名后，可以运行测试，效果同扫描二维码的一致

**2.2.3 按实例数鉴权** 设置好序列号和包名后，调用配置类的以下方法启用并配置心跳间隔时间：

```
XXXConfig config = new XXXConfig();
// 启用按实例数鉴权，配置心跳间隔，单位：秒
config.setInstanceAuthMode(10000);
```

配置类的详细说明参考后续章节【调用流程】

## 2.3 测试精简版

对于通用ARM、高通骁龙引擎SNPE、华为NPU引擎DDK和达芬奇NPU引擎Davinci的常见功能，项目内自带精简版，可以忽略开发板不兼容的摄像头。

此外，由于实时摄像开启，会导致接口的耗时变大，此时也可以使用精简版测试。

目前以下硬件环境有精简版测试：

- 通用ARM：图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation)，姿态估计 (Pose)
- 通用ARM GPU：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)，姿态估计 (Pose)
- 高通骁龙引擎SNPE：图像分类 (Classify)，物体检测 (Detection)
- 华为NPU引擎DDK：图像分类 (Classify)，物体检测 (Detection)
- 华为达芬奇NPU引擎Davinci：图像分类 (Classify)，物体检测 (Detection)，图像分割 (Segmentation)

具体代码分别在infertest、snpetest、ddktest和davincitest目录下。

修改方法为（以通用ARM为例）：更改app/main/AndroidManifest.xml中的启动Activity。

```
<activity android:name=".infertest.MainActivity"> <!-- 原始的是".MainActivity" -->
  <intent-filter>
    <action android:name="android.intent.action.MAIN" />

    <category android:name="android.intent.category.LAUNCHER" />
  </intent-filter>
</activity>
```

开启后会自动选择图像分类 (Classify)，物体检测 (Detection)，文字识别 (OCR)，图像分割 (Segmentation) 或姿态估计 (Pose) 测试。

Demo APP 检测模型运行示例

精简版检测模型运行示例



```

Hello World!
ARM Detection
Start running: 0
Predict 0: (size:100, firstRe
confidence:0.6314938, bo
181)}}
Finish running
Task finished
  
```

### 识别结果

置信度  0.30

序号	名称	置信度
1	person	0.63
2	person	0.47
3	car	0.42
4	horse	0.40
5	dog	0.34
6	truck	0.34

BU

#### 使用说明

##### 3.1 代码目录结构

集成时需要“复制到自己的项目里”的目录或者文件：

1. app/libs

## 2. app/src/main/assets/xxxx-xxxx 如app/src/main/assets/infer

```

+app 简单的设置，模拟用户的项目
|---+libs 实际使用时需要复制到自己的项目里
    |---arm64-v8a v8a的so
    |---armeabi-v7a v7a的so
    |---easyedge-sdk.jar jar库文件
|---+src/main
    |---+assets
        |---demo demo项目的配置，实际集成不需要
        |---infer 也可能是其它命名，infer表示通用arm。实际使用时可以复制到自己的项目里
|---+java/com.baidu.ai.edge/demo
    |---+infertest 通用Arm精简版测试，里面有SDK的集成逻辑
        |--- MainActivity 通用Arm精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里
        面的序列号
            |--- TestInferArmClassifyTask 通用Arm精简版分类
            |--- TestInferArmDetectionTask 通用Arm精简版检测
            |--- TestInferArmOcrTask 通用Arm精简版OCR
            |--- TestInferArmPoseTask 通用Arm精简版姿态
            |--- TestInferArmSegmentTask 通用Arm精简版分割
            |--- TestInferArmGpuClassifyTask 通用ArmGpu精简版分类
            |--- TestInferArmGpuDetectionTask 通用ArmGpu精简版检测
            |--- TestInferArmGpuPoseTask 通用ArmGpu精简版姿态
            |--- TestInferArmGpuSegmentTask 通用ArmGpu精简版分割
    |---+snpetest SNPE精简版测试
        |--- MainActivity SNPE精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
        的序列号
            |--- TestSnpeDspClassifyTask SNPE DSP精简版分类
            |--- TestSnpeDspDetectionTask SNPE DSP精简版检测
            |--- TestSnpeGpuClassifyTask SNPE Gpu精简版分类
            |--- TestSnpeGpuDetectionTask SNPE Gpu精简版检测
    |---+ddktest DDK精简版测试
        |--- MainActivity DDK精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面的
        序列号
            |--- TestDDKClassifyTask DDK精简版分类
            |--- TestDDKDetectionTask DDK精简版检测
    |---+davincitest Davinci精简版测试
        |--- MainActivity Davinci精简版启动Activity，会根据assets目录判断当前的模型类型，并运行同目录的一个Task。使用时需要修改里面
        的序列号
            |--- TestDavinciClassifyTask Davinci精简版分类
            |--- TestDavinciDetectionTask Davinci精简版检测
            |--- TestDavinciSegmentTask Davinci精简版分割
        |--- CameraActivity 摄像头扫描示例，里面有SDK的集成逻辑
        |--- MainActivity 启动Activity，使用时需要修改里面的序列号
|--- build.gradle 这里修改包名
+camera_ui UI模块，集成时可以忽略

```

## 3.2 调用流程 以通用ARM的检测模型功能为例，

代码可以参考TestInferDetectionTask

1. 准备配置类，如InferConfig，输入：通常为一个assets目录下的文件夹，如infer。
2. 初始化Manager，比如InferManager。输入：第1步的配置类和序列号
3. 推理图片，可以多次调用 3.1 准备图片，作为Bitmap输入 3.2 调用对应的推理方法，比如detect 3.3 解析结果，结果通常是一个List，调用结果类的Get方法，通常能获取想要的结果
4. 直到长时间不再使用我们的SDK，调用Manger的destroy方法释放资源。

## 3.3 具体接口说明 下文的示例部分以通用ARM的检测模型功能为例

即接口为InferConfig， InferManager， InferManager.detect。

其它引擎和模型调用方法类似。

下文假设已有序列号及对应的包名

### 3.3.1. 准备配置类

- INFER：通用ARM，`InferConfig`
- ARM GPU：ArmGpuConfig
- SNPE：高通骁龙DSP，`SnpeConfig`
- SNPE GPU：高通骁龙GPU，`SnpeGpuConfig`
- DDK：华为NPU，`DDKConfig`
- DDKDAVINCI：华为达芬奇NPU，`DDKDaVinciConfig`

```
InferConfig mInferConfig = new InferConfig(getAssets(),
    "infer");
// assets 目录下的infer，infer表示通用arm
```

输入：assets下的配置  
输出：具体的配置类

### 3.3.2. 初始化Manager类

- INFER：通用ARM，`InferManager`
- ARM GPU：通用ARM GPU，`InferManager`
- SNPE：高通骁龙DSP，`SnpeManager`
- SNPE GPU：高通骁龙GPU，`SnpeManager`
- DDK：华为NPU，`DDKManager`
- DDKDAVINCI：华为达芬奇NPU，`DavinciManager`

```
String SERIAL_NUM = "XXXX-XXXX-XXXX-XXXX";

// InferManager 为例:
new InferManager(this, config, SERIAL_NUM); // config为上一步的InferConfig
```

#### 注意要点

1. 同一个时刻只能有唯一有效的InferManager。旧的InferManager必须调用destory后，才能新建一个new InferManager()。
2. InferManager的任何方法，都不能在UI线程中调用。
3. new InferManager() 及InferManager成员方法由于线程同步数据可见性问题，都必须在一个线程中执行。如使用android自带的ThreadHandler类。

输入：1.配置类；2.序列号  
输出：Manager类

### 3.3.3. 推理图片

- 接口可以多次调用，但是必须在一个线程里，不能并发
- confidence, 置信度[0-1]，小于confidence的结果不返回。填confidence=0，返回所有结果
- confidence可以不填，默认用模型推荐的。

准备图片，作为Bitmap输入，

- 输入为Bitmap，其中Bitmap的options为默认。如果强制指定的话，必须使用`Bitmap.Config.ARGB_8888`

调用对应的推理方法及结果解析 见下文的各个模型方法

### 3.3.4 分类Classify

```
public interface ClassifyInterface {
    List<ClassificationResultModel> classify(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 ClassifyInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 ClassificationResultModel  
异常：一般首次出现。可以打印出异常错误码。

ClassificationResultModel

- label：分类标签，定义在label\_list.txt中
- confidence：置信度，0-1
- labelIndex：标签对应的序号

### 3.3.5 检测Detect

对于EasyDL口罩检测模型请注意输入图片中人脸大小建议保持在88到9696像素，可根据场景远近程度缩放图片后传入

```
public interface DetectInterface {
    List<DetectionResultModel> detect(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 DetectInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 DetectionResultModel List  
异常：一般首次出现。可以打印出异常错误码。

DetectionResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- bounds：Rect，左上角和右下角坐标

### 3.3.6 图像分割Segmentation

```
public interface SegmentInterface {
    List<SegmentationResultModel> segment(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 SegmentInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
输出 SegmentationResultModel  
异常：一般首次出现。可以打印出异常错误码。

SegmentationResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- labelIndex：标签对应的序号
- box: Rect对象表示的对象框
- mask：byte[]表示的原图大小的0，1掩码，绘制1的像素即可得到当前对象区域

mask 字段说明，如何绘制掩码也可参考demo工程

```
1 0 1
image 1 1 0  => mask(byte[]) 101 110 011
0 1 1
```

### 3.3.7 文字识别OCR

暂时只支持通用ARM引擎，不支持其它引擎，暂时只支持EasyEdge的开源OCR模型。

```
public interface OcrInterface {
    List<OcrResultModel> ocr(Bitmap bitmap, float confidence) throws BaseException;

    // 如InferManger 继承 OcrInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 OcrResultModel List，每个OcrResultModel对应结果里的一个四边形。  
 异常：一般首次出现。可以打印出异常错误码。

OcrResultModel

- label：识别出的文字
- confidence：置信度
- List<Point>：4个点构成四边形

### 3.3.8 姿态估计Pose

暂时只支持通用ARM引擎，不支持其它引擎

```
public interface PoseInterface {
    List<PoseResultModel> pose(Bitmap bitmap) throws BaseException;
    // 如InferManger 继承 PoseInterface
```

输入 Bitmap 默认格式或者指定Bitmap.Config.ARGB\_8888  
 输出 PoseResultModel List  
 异常：一般首次出现。可以打印出异常错误码。

PoseResultModel

- label：标签，定义在label\_list.txt中
- confidence：置信度
- Pair<Point, Point>：2个点构成一条线

### 3.3.9 释放

释放后这个对象不能再使用，如果需要使用可以重新new一个出来。

```
public void destory() throws BaseException
```

### 3.3.10 整体示例

以通用ARM的图像分类预测流程为例：

```
try {
    // step 1: 准备配置类
    InferConfig config = new InferConfig(context.getAssets(), "infer");

    // step 2: 准备预测 Manager
    InferManager manager = new InferManager(context, config, "");

    // step 3: 准备待预测的图像，必须为 Bitmap.Config.ARGB_8888 格式，一般为默认格式
    Bitmap image = getFromSomeWhere();

    // step 4: 预测图像
    List<ClassificationResultModel> results = manager.classify(image, 0.3f);

    // step 5: 解析结果
    for (ClassificationResultModel resultModel : results) {
        Log.i(TAG, "labelIndex=" + resultModel.getLabelIndex()
            + ", labelName=" + resultModel.getLabel()
            + ", confidence=" + resultModel.getConfidence());
    }

    // step 6: 释放资源。预测完毕请及时释放资源
    manager.destory();
} catch (Exception e) {
    Log.e(TAG, e.getMessage());
}
```

### 3.3.11 高通骁龙引擎的额外配置

```
"autocheck_qcom": true, // 如果改成false, sdk跳过检查手机是否是高通的Soc, 非高通的Soc会奔溃直接导致app闪退
```

```
"snpe_runtimes_order": [],
// 不填写为自动, 按照 {DSP, GPU, GPU_FLOAT16, CPU}次序尝试初始化, 也可以手动指定如[2,1,3,0], 具体数字的定义见下段
```

```
public interface SnpeRuntimeInterface {
    int CPU = 0;
    int GPU = 1;
    int DSP = 2;
    int GPU_FLOAT16 = 3;
}
```

```
// SnpeManager 中, 使用public static ArrayList<Integer> getAvailableRuntimes(Context context) 方法可以获取高通SOC支持的运行方式
```

## 集成指南

1. 复制库文件libs
2. 添加Manifest权限
3. 复制模型文件
4. 添加调用代码(见上一步具体接口说明)

### 4.1 复制库文件libs A. 如果项目里没有自己的jar文件和so文件:

复制app/libs 至自己项目的app/libs目录。  
参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
    sourceSets {
        main {
            jniLibs.srcDirs = ['libs']
        }
    }
}
```

### B. 如果项目里有自己的jar文件, 但没有so文件

easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a放到app/src/main/jniLibs目录下

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a'
        }
    }
}
```

### C. 如果项目里有自己的jar文件和so文件



easyedge.jar文件同自己的jar文件放在一起  
arm64-v8a和armeabi-v7a取交集和自己的so放在一起，交集的意思是比如自己的项目里有x86目录，必须删除x86。

参照demo的app/build.gradle 中添加

```
android {
    ....
    defaultConfig {
        ndk {
            abiFilters 'armeabi-v7a', 'arm64-v8a' // abiFilter取交集，即只能少不能多
        }
    }
}
```

jar文件库如果没有设置成功的，编译的时候可以发现报错。

so库如果没有编译进去的话，也可以通过解压apk文件确认。运行的时候会有类似jni方法找不到的报错。

#### 4.2 Manifest配置

参考app/src/main/AndroidManifest.xml文件，添加：

```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />
<uses-permission android:name="android.permission.READ_PHONE_STATE" />
<!-- Android 11 支持 -->
<uses-permission
    android:name="android.permission.MANAGE_EXTERNAL_STORAGE"
    tools:ignore="ScopedStorage" />

<!-- 高版本 Android 支持 -->
<application
    android:requestLegacyExternalStorage="true"
    android:usesCleartextTraffic="true">
</application>
```

#### 4.3 混淆规则（可选） 请不要混淆SDK里的jar文件。

```
-keep class com.baidu.ai.edge.core.*.*{ *; }
```

#### 4.4 Android 11支持 除Manifest中必要配置外，请参考BaseActivity获取所有文件访问权限，否则可能影响SDK正常使用。

SDK 默认使用 easyedge-sdk.jar，未启用 AndroidX，若您的项目使用 AndroidX，并在集成中提示 android.support 相关错误，请参考 app/build.gradle 使用 etc/easyedge-sdk-androidx.jar 以支持 AndroidX：

```
// app/build.gradle

dependencies {
    implementation project(':camera_ui')
    implementation files('libs/easyedge-sdk-androidx.jar') // 修改 jar 包依赖
}
```

**错误码** | 错误码 | 错误描述 | 详细描述及解决方法 | |---|---|---| | 1001 | assets 目录下用户指定的配置文件不存在或不正确 | SDK使用assets目录下一系列文件作为配置文件。如果文件缺失或内容不正确，则有此报错 | | 1002 | json格式的配置文件解析出错 | 如缺少某些字段。正常情况下，配置文件请不要修改 | | 1003 | 应用缺少权限 | 请根据提示动态申请缺少的权限 | | 19xx | Sdk内部错误 | 请与百度人员联系 | | 2001 | XxxMANAGER 只允许一个实例 | 如已有XxxMANAGER对象，请调用destory方法 | | 2002 | XxxMANAGER 已经调用过destory方法 | 在一个已经调用destory方法的DETECT\_MANAGER对象上，不允许再调用任何方法 | | 2003 | 传入的assets下模型文件路径非法 | 比如缺少模型文件，XxxConfig.getModelFileAssetPath() 返回为null | | 2012 | JNI内存错误 | heap的内存不够 | | 2103 | license过期 | license失效或者系统时间有异常 | | 2601/2602 | assets 目录下模型文件打开/读取失败 | 请根据报错信息检查模型文件是否存在 | | 27xx | Sdk内部错误 | 请与百度人员联系 | | 28xx | 引擎内部错误 | 请与百度人员联系 | | 29xx | Sdk内部错误 | 请与百度人员联系 | | 3000 | so加载错误 | 请确认所有so文件存在于apk中 | | 3001 | 模型加载错误 | 请确认模型放置于能被加载到的合法路径中，并确保配置文件正确 | | 3002 | 模型卸载错误 | 请与百度人员联系 | | 3003 | 调用模型错误 | 在模型未加载正确或者so库未加载正确的情况下调用了分类接口 | | 31xx | SDK激活失败 | 请与百度人员联系 | | 4011 | SDK类型与设备硬件不匹配 | 比如适配DSP的SDK运行在麒麟芯片上会出现此报错，请在部署包支持的硬件上使用SDK | | 50xx | SDK调用异常 | 请与百度

人员联系 |

**报错日志收集** 通常 Logcat 可以看见日志及崩溃信息，若设备无法获取日志信息，可使用 Demo 中的 xCrash 工具：

```
// 1. 引入 app/build.gradle 的 xCrash 依赖
android {
    ...
    dependencies {
        implementation 'com.iqiyi.xcrash:xcrash-android-lib:2.4.5' // 可以保存崩溃信息，默认未引入
        ...
    }
}
// 2. 启用日志收集。日志将保存在 /sdcard/<包名>/xCrash
// app/src/main/java/com.baidu.ai.edge/demo/MyApplication.java
protected void attachBaseContext(Context context) {
    // 日志保存位置
    String basePath = Environment.getExternalStorageDirectory().toString() + "/" + context.getPackageName();
    // 启用
    XCrash.InitParameters params = new XCrash.InitParameters();
    params.setAppVersion(BaseManager.VERSION);
    params.setLogDir(basePath + "/xCrash");
    XCrash.init(this, params);
}
```

## 🔗 iOS集成文档

### 简介

本文档描述 EasyEdge/EasyDL iOS 离线预测SDK相关功能；

目前支持EasyEdge的功能包括：

- 图像分类
- 物体检测
- 人脸检测
- 姿态估计
- 百度OCR模型

目前支持EasyDL的功能包括：

- 图像分类
- 物体检测
- 图像分割

### 系统支持

系统：

- 通用arm版本：iOS 9.0 以上
- A仿生芯片版：iOS 15.0 及以上

硬件：arm64 (Standard architectures)（暂不支持模拟器）

内存：图像分割模型需要手机内存3GB以上，并尽量减少其他程序内存占用

### 离线SDK包说明

根据用户的选择，下载的离线SDK，可能包括以下类型：

- EasyEdge
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。

- EasyDL 通用版/全功能AI开发平台BML（原EasyDL专业版）
  - 通用ARM版：支持iPhone5s, iOS 9.0 以上所有手机。
  - A仿生芯片版：支持iPhone5s, iOS 15.0 以上手机。充分利用苹果A系列仿生芯片优势，在iPhone 8以上机型中能有显著的速度提升。
  - 自适应芯片版：同时整合了以上两种版本，自动在iOS 15以下中使用通用ARM版，在iOS 15以上系统中使用A仿生芯片版，自适应系统，但SDK体积相对较大。
- AI市场试用版SDK

#### SDK大小说明

SDK库的二进制与\_TEXT增量约3M。

资源文件大小根据模型不同可能有所差异。

物体检测(高性能)的DemoApp在iPhone 6, iOS 11.4下占用空间实测小于40M。

虽然SDK库文件很大（体现为SDK包文件很大，ipa文件很大），但最终应用在用户设备中所占用的大小会缩小很多。这与multi architectures、bitcode和AppStore的优化有关。

**获取序列号** 生成SDK后，点击获取序列号进入控制台获取。EasyEdge[控制台](#)、EasyDL[控制台](#)、BML[控制台](#)。

试用版SDK在SDK的RES文件夹中的SN.txt中包含试用序列号。

更换序列号、更换设备时，首次使用需要联网激活。激活成功之后，有效期内可离线使用。

#### Release Notes

时间	版本	说明
2023.08.31	0.7.13	新增按实例数鉴权；迭代优化
2023.06.29	0.7.12	迭代优化
2023.05.17	0.7.11	CoreML引擎升级，支持更多语义分割模型；兼容横屏；迭代优化
2023.03.16	0.7.10	支持更多语义分割模型；迭代优化
2022.12.29	0.7.9	ARM引擎升级；迭代优化
2022.10.27	0.7.8	支持更多检测模型；迭代优化
2022.09.15	0.7.7	支持更多检测模型；迭代优化
2022.07.28	0.7.6	迭代优化
2022.06.29	0.7.5	支持EasyEdge语义分割模型；CoreML引擎升级，新增EasyEdge检测模型支持；迭代优化
2022.05.18	0.7.4	ARM引擎升级；支持EasyDL物体检测超高精度模型；支持更多加速版模型发布；迭代优化
2022.03.25	0.7.3	ARM引擎升级；支持更多检测模型
2021.12.22	0.7.2	支持EasyEdge更多姿态估计模型；迭代优化
2021.10.20	0.7.1	ARM引擎升级
2021.07.29	0.7.0	迭代优化
2021.04.06	0.6.1	ARM引擎升级
2021.03.09	0.6.0	支持EasyEdge人脸检测及姿态估计模型
2020.12.18	0.5.7	ARM引擎升级
2020.09.17	0.5.6	CoreML引擎升级，支持AI市场试用版SDK
2020.08.11	0.5.5	CoreML支持EasyDL专业版模型，支持EasyEdge OCR模型
2020.06.23	0.5.4	ARM引擎升级
2020.04.16	0.5.3	ARM引擎升级；支持压缩加速版模型
2020.03.13	0.5.2	ARM引擎升级；支持图像分割模型
2020.01.16	0.5.1	ARM引擎升级；增加推荐阈值支持
2019.12.04	0.5.0	ARM引擎升级；增加coreml3的支持
2019.10.24	0.4.5	支持EasyDL专业版；ARM引擎升级
2019.08.30	0.4.4	支持EasyDL经典版图像分类高性能、高精度
2019.06.20	0.4.3	引擎优化
2019.04.12	0.4.1	支持EasyDL经典版物体检测高精度、高性能模型
2019.03.29	0.4.0	引擎优化，支持CoreML；
2019.02.28	0.3.0	引擎优化，性能与效果提升；
2018.11.30	0.2.0	第一版！

### 快速开始 文件结构说明

```

.EasyEdge-iOS-SDK
├── EasyDLDemo # Demo工程文件
├── LIB # 依赖库
├── RES
│   ├── easyedge # 模型资源文件夹
│   │   ├── model
│   │   ├── params
│   │   ├── label_list.txt
│   │   ├── infer_cfg.json
│   │   └── conf.json
└── DOC # 文档

```

### 测试Demo

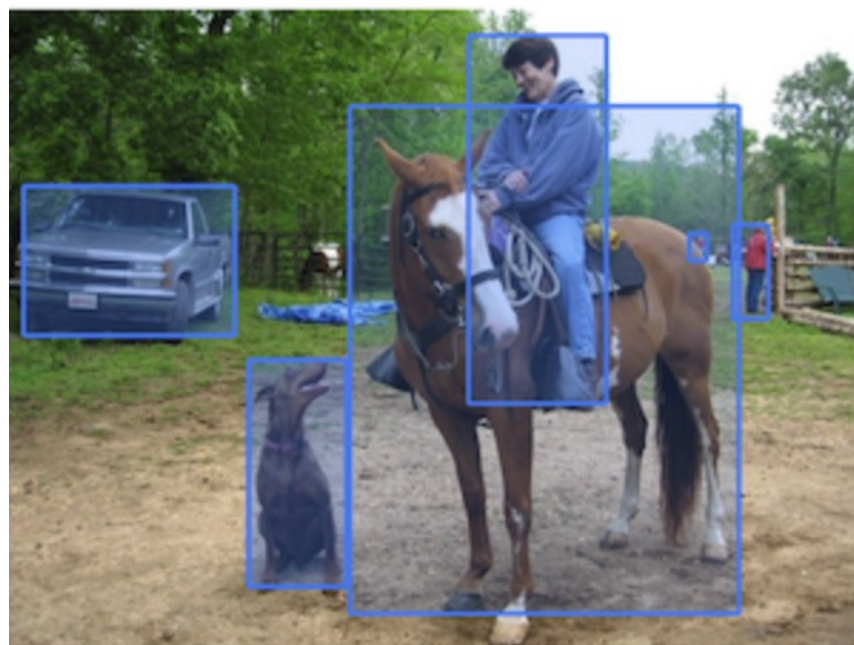
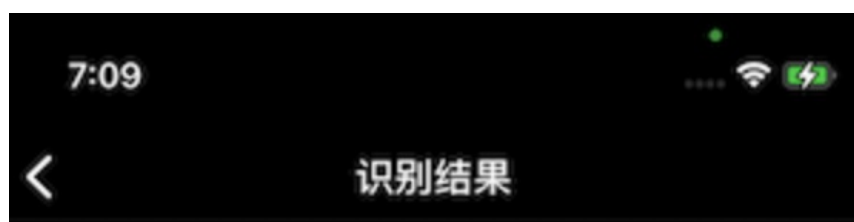
按如下步骤可直接运行 SDK 体验 Demo：

步骤一：用 Xcode 打开 EasyDLDemo/EasyDLDemo.xcodeproj

步骤二：配置开发者自己的签名

步骤三：连接手机运行，不支持模拟器

检测模型运行示例：



阈值：0.30



序号	名称	置信度
1	person	0.632
2	person	0.468
3	car	0.423
4	horse	0.400
5	dog	0.342

重新识别

SDK使用说明 集成指南 步骤一：依赖库集成 步骤二：import <EasyDL/EasyDL.h> , import <Vision/Vision.h>

### 依赖库集成

1. 复制 LIB 目录至项目合适的位置
2. 配置 Build Settings 中 Search paths: 以 SDK 中 LIB 目录路径为例

- Framework Search Paths : \${PROJECT\_DIR}/../LIB/lib
- Header Search Paths : \${PROJECT\_DIR}/../LIB/include
- Library Search Paths : \${PROJECT\_DIR}/../LIB/lib

集成过程如出现错误，请参考 Demo 工程对依赖库的引用

### 使用流程

1. 生成模型，下载SDK 开发者在官网下载的SDK已经自动为开发者配置了模型文件和相关配置，开发者直接运行即可。
2. 使用序列号激活 2.1. 离线激活（默认鉴权方式） 首次联网激活，后续离线使用

将前面申请的序列号填入：

```
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
```

根据序列号类型，序列号与BundleID绑定或与BundleID+设备绑定。

请确保设备时间正确。

- 2.2. 按实例数激活 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间

填入序列号，配置按实例数鉴权并设置心跳间隔：

```
// 设置序列号
[EasyDL setSerialNumber:@"!!!Enter Your Serial Number Here!!!"];
// 配置实例数鉴权及心跳间隔，单位：秒
[EasyDL setInstanceAuthMode:10000];
```

### 3. 初始化模型

```
EasyDLModel *_model = [[EasyDLModel alloc] initWithResourceDirectory:@"easymodel" withError:&err];
```

请注意相关资源必须以 folder reference 方式加入Xcode工程。也即默认的easymodel文件夹在Xcode文件列表里显示为蓝色。

### 4. 调用检测接口

```
UIImage *img = .....;
NSArray *result = [model detectUIImage:img withFilterScore:0 andError:&err];

/**
 * 检测图像
 * @param image 带检测图像
 * @param score 只返回得分高于score的结果(0 ~ 1)
 * @return 成功返回识别结果，NSArray的元素为对应模型的结果类型；失败返回nil，并在err中说明错误原因
 */
- (NSArray *)detectUIImage:(UIImage *)image
  withFilterScore:(CGFloat)score
    andError:(NSError **)err;
```

返回的数组类型如下，具体可参考 EasyDLResultData.h 中的定义：

模型类型	类型
图像-图像分类	EasyDLClassfiData
图像-物体检测/人脸检测	EasyDLObjectDetectionData
图像-实例分割/语义分割	EasyDLObjSegmentationData
图像-姿态估计	EasyDLPoseData
图像-文字识别	EasyDLOcrData

### 错误说明

SDK的方法会返回NSError错，直接返回的NSError的错误码定义在EEasyDLErrorCode中。NSError附带message（有时候会附带NSUnderlyingError），开发者可根据code和message进行错误判断和处理。

### FAQ

#### 1. 如何多线程并发预测？

SDK内部已经能充分利用多核的计算能力。不建议使用并发来预测。

如果开发者想并发使用，请务必注意EasyDLModel所有的方法都不是线程安全的。请初始化多个实例进行并发使用，如

```

- (void)testMultiThread {
    UIImage *img = [UIImage imageNamed:@"1.jpeg"];
    NSError *err;
    EasyDLModel * model1 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];
    EasyDLModel * model2 = [[EasyDLModel alloc] initWithResourceDirectory:@"easyedge" withError:&err];

    dispatch_queue_t queue1 = dispatch_queue_create("testQueue", DISPATCH_QUEUE_CONCURRENT);
    dispatch_queue_t queue2 = dispatch_queue_create("testQueue2", DISPATCH_QUEUE_CONCURRENT);

    dispatch_async(queue1, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model1 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"1: %@", res[0];
        }
    });

    dispatch_async(queue2, ^{
        NSError *detectErr;
        for(int i = 0; i < 1000; ++i) {
            NSArray * res = [model2 detectUIImage:img withFilterScore:0 andError:&detectErr];
            NSLog@"2: %@", res[0];
        }
    });
}

```

#### 2. 编译时出现 Undefined symbols for architecture arm64: ...

- 出现 `cx11, vtable` 字样：请引入 `libc++.tbd`
- 出现 `cv::Mat` 字样：请引入 `opencv2.framework`
- 出现 `CoreML, VNRequest` 字样：请引入 `CoreML.framework` 并务必 `#import <CoreML/CoreML.h>`

#### 3. 运行时报错 Image not found: xxx ...

请Embed具体报错的库。4.编译时报错：Invalid bitcode version 这个可能是开发者使用的xcodes低于12导致，可以升级至12版本。

### Windows集成文档

#### 简介

本文档介绍图像分割服务器端Windows SDK的使用方法。

- 硬件支持：
  - Intel CPU 普通版 \* x86\_64



- CPU 加速版 - Intel Xeon with AVX2 and AVX512 - *Intel Core Processors with AVX2* - Intel Atom Processors with SSE \* - AMD Core Processors with AVX2
- Intel Movidius Myriad2/Myriad X (仅支持Win10)
- 操作系统支持
  - 普通版：64位 Windows 7 及以上，64位Windows Server2012及以上
  - 加速版：64位 Windows 10，64位Windows Server 2019及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015-2019
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

\*intel 官方合作，拥有更好的适配与性能表现

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | 优化模型算法 | | 2022-09-15 | 1.7.0 | 新增支持表格预测 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | CPU基础版推理引擎优化升级；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | CPU加速版推理引擎优化升级 | | 2021-08-19 | 1.3.2 | 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版 | |

## 快速开始

### 1. 安装依赖

必须安装：

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

Visual C++ Redistributable Packages for Visual Studio 2015-2019

<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-160>

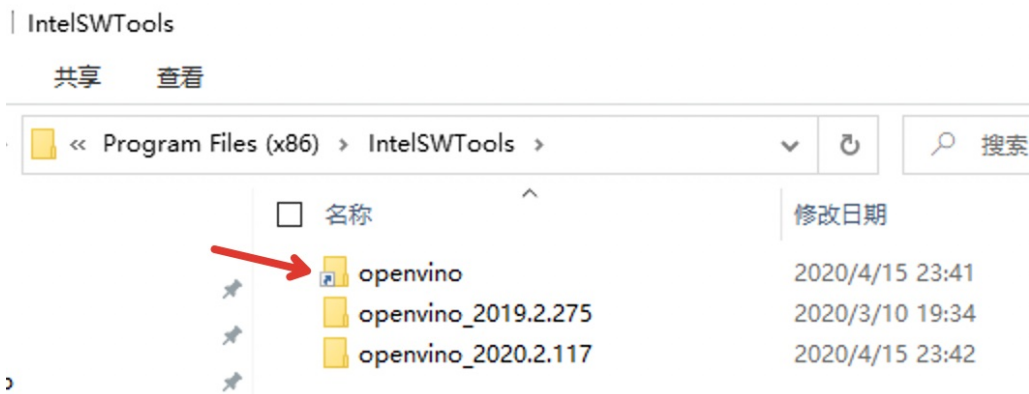
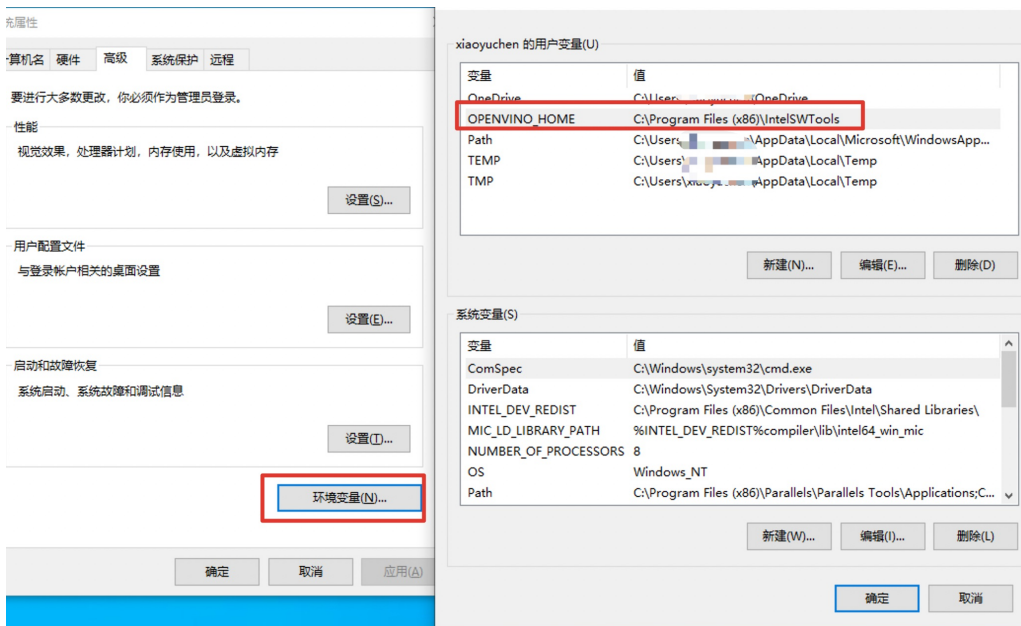
可选安装：



**Openvino (仅使用Python Intel Movidius必须)**

- 使用 OpenVINO™ toolkit 安装, 请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装, 请参考 [Openvino Inference Engine文档](#) 编译安装 2020.3.1LTS (必须) 版本。

安装完成后, 请设置环境变量OPENVINO\_HOME为您设置的安装地址, 默认是C:\Program Files (x86)\IntelSWTools, 并确保文件夹下的openvino的快捷方式指到了2020.3.1LTS版本。

**注意事项**

1. 安装目录不能包含中文
2. Windows Server 请自行开启, 选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“, 点击安装, 安装之后重启即可。

**2. 运行离线SDK**

解压下载好的SDK, 打开EasyEdge.exe, 输入Serial Num, 选择鉴权模式, 点击“启动服务“, 等待数秒即可启动成功, 本地服务默认运行在

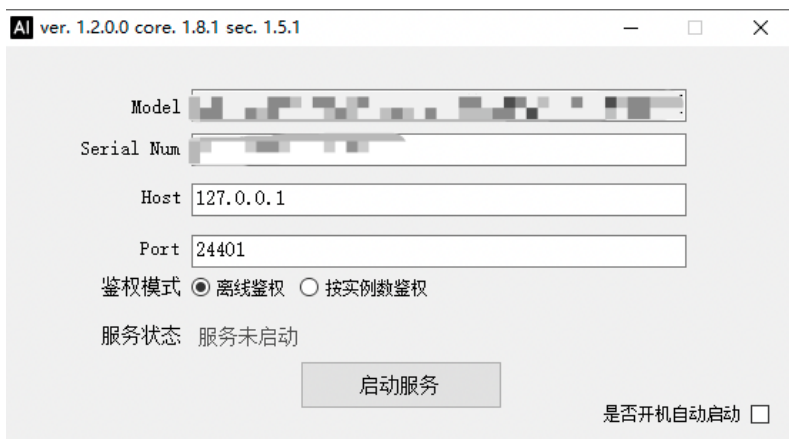
`http://127.0.0.1:24401/`

其他任何语言只需通过HTTP调用即可。

如启动失败, 可参考如下步骤排查:



## 2.1 离线鉴权（默认鉴权模式） 首次联网激活，后续离线使用



## 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

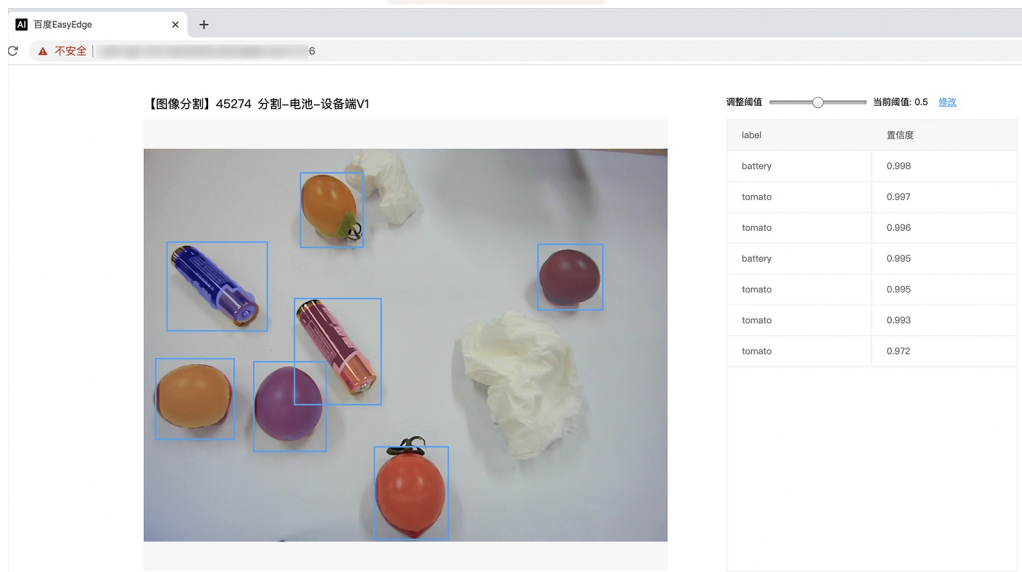
```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

## 2.3 序列号激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。



使用说明

调用说明

使用示例代码如下

```
python

c#

C++

java
```

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img).json()

print(result)
```

结果 获取的结果存储在response字符串中。 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|----|-----| | confidence | float | 0~1 | 分割的置信度 | | label | string | | 分割的类别 | | index | number | | 分割的类别 | | mask | string | | 游程编码的mask | 代码参考 <https://github.com/Baidu-AIP/EasyDL-Segmentation-Demo>

## 集成指南

### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

### 基于c++ dll集成

#### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

#### 集成方法

参考src目录中的CMakeLists.txt进行集成

### 基于c# dll集成

#### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

#### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
 .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是CPU加速版，需额外确保Opencv安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

#### 4. JAVA、C#等其他语言怎么调用SDK?

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败, 缺失DLL? 打开EasyEdge.log, 查看日志错误, 根据提示处理 缺失DLL, 请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些, 请自行下载安装

6. 启动失败, 报错NotDecrypted? Windows下使用, 当前用户名不能为中文, 否则无法正确加载模型。

#### 7. 启动失败, 报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更, 包括 (但不局限于) 以下可能的情况:

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况, 请确保硬件无变更, 如果想更换序列号, 请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录, 再重新激活。

#### 8. 勾选“开机自动启动”后, 程序闪退

一般是写注册表失败。

可以确认下HKEY\_CURRENT\_USER下Software\Microsoft\Windows\CurrentVersion\Run能否写入 (如果不能写入, 可能被杀毒软件等工具管制)。也可以尝试基于bin目录下的easyedge\_serving.exe命令行形式的二进制, 自行配置开机自启动。

**其他问题** 如果无法解决, 可到论坛发帖: <https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题, 我们将及时回复您的问题。

### Linux集成文档-C++

#### 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持: - 图像分类 - 物体检测 - 图像分割
- 硬件支持:
  - CPU: aarch64 armv7hf
  - GPU: ARM Mali G系列
  - ASIC: Hisilicon NNIE1.1 on aarch64 (Hi3559AV100/Hi3559CV100等)
  - ASIC: Hisilicon NNIE1.2 on armv7l (Hi3519AV100/Hi3559V200等)
  - Intel Movidius Myriad2 / Myriad X on x86\_64
  - Intel Movidius Myriad2 / Myriad X on armv7l
  - Intel Movidius Myriad2 / Myriad X on aarch64
  - Intel iGPU on x86\_64
  - 比特大陆 Bitmain SE50 (BM1684)
  - 瑞芯微 RK3399Pro / RV1109 / RV1126 / RK3568 / RK3588
  - 华为 Atlas200
  - 晶晨 A311D
  - 寒武纪 MLU220 on aarch64
  - 英特尔 iGPU
- 操作系统支持:

- Linux (Ubuntu, Centos, Debian等)
- 海思HiLinux
- 树莓派Raspbian/Debian
- 瑞芯微Firefly

性能数据参考[算法性能及适配硬件](#)

**Release Notes** | 时间 | 版本 | 说明 | |---|---|---| | 2023.08.31 | 1.8.3 | Atlas系列Socs支持语义分割模型, Atlas Cann版本升级至6.0.1 | | 2023.06.29 | 1.8.2 | 比特大陆版本升级至V23.03.01 | | 2023.05.17 | 1.8.1 | 新增支持intel iGPU + CPU异构模式 | | 2023.03.16 | 1.8.0 | 新增支持瑞芯微RK3588 | | 2022.10.27 | 1.7.1 | 新增语义分割模型http请求示例 | | 2022.09.15 | 1.7.0 | 新增瑞芯微 RK3568 支持, RK3399Pro、RV1126升级到RKNN1.7.1 | | 2022.07.28 | 1.6.0 | 引擎升级; 新增英特尔 iGPU 支持 | | 2022.04.25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022.03.25 | 1.4.0 | EasyDL新增上线支持晶晨A311D NPU预测引擎; Arm CPU、Arm GPU引擎升级; atlas 200在EasyDL模型增加多个量化加速版本; | | 2021.12.22 | 1.3.5 | RK3399Pro, RV1109/RV1126 SDK扩展模型压缩加速能力, 更新端上推理库版本;边缘控制台IEC功能升级, 适配更多通用小型设备, NNIE 在EasyDL增加量化加速版本; Atlas200升级到Cann5.0.3 | | 2021.06.29 | 1.3.1 | 视频流解析支持调整分辨率; 预测引擎升级; 设备端sdk新增支持瑞芯微RV1109、RV1126 | | 2021.05.13 | 1.3.0 | 新增视频流接入支持; EasyDL模型发布新增多种加速方案选择; 目标追踪支持x86平台的CPU、GPU加速版; 展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.0 | http server服务支持图片通过base64格式调用 | | 2021.01.27 | 1.1.0 | EasyDL经典版分类高性能模型升级; 部分SDK不再需要单独安装OpenCV; 新增RKNNPU预测引擎支持; 新增高通骁龙GPU预测引擎支持 | | 2020.12.18 | 1.0.0 | 1.0版本发布! 安全加固升级、性能优化、引擎升级、接口优化等多项更新 | | 2020.10.29 | 0.5.7 | 优化多线程预测细节 | | 2020.09.17 | 0.5.6 | 支持linux aarch64架构的硬件接入intel神经计算棒预测; 支持比特大陆计算盒SE50 BM1684 | | 2020.08.11 | 0.5.5 | 支持linux armv7hf架构硬件(如树莓派)接入intel神经计算棒预测 | | 2020.06.23 | 0.5.4 | arm引擎升级 | | 2020.05.15 | 0.5.3 | 支持EasyDL 专业版新增模型; 支持树莓派(armv7hf, aarch64) | | 2020.04.16 | 0.5.2 | Jetson系列SDK支持多线程infer | | 2020.02.23 | 0.5.0 | 新增支持人脸口罩模型; Jetson SDK支持批量图片推理; ARM支持图像分割 | | 2020.01.16 | 0.4.7 | 上线海思NNIE1.2, 支持EasyEdge以及EasyDL; ARM引擎升级; 增加推荐阈值支持 | | 2019.12.26 | 0.4.6 | 海思NNIE支持EasyDL专业版 | | 2019.11.02 | 0.4.5 | 移除curl依赖; 支持自动编译OpenCV; 支持EasyDL 专业版 Yolov3; 支持EasyDL经典版高精度物体检测模型升级 | | 2019.10.25 | 0.4.4 | ARM引擎升级, 性能提升30%; 支持EasyDL专业版模型 | | 2019.09.23 | 0.4.3 | 增加海思NNIE加速芯片支持 | | 2019.08.30 | 0.4.2 | ARM引擎升级; 支持分类高性能与高精度模型 | | 2019.07.25 | 0.4.1 | 引擎升级, 性能提升 | | 2019.06.11 | 0.3.3 | paddle引擎升级; 性能提升 | | 2019.05.16 | 0.3.2 | 新增armv7l支持 | | 2019.04.25 | 0.3.1 | 优化硬件支持 | | 2019.03.29 | 0.3.0 | ARM64 支持; 效果提升 | | 2019.02.20 | 0.2.1 | paddle引擎支持; 效果提升 | | 2018.11.30 | 0.1.0 | 第一版! |

【1.0 接口升级】参数配置接口从1.0.0版本开始已升级为新接口, 以前的方式被置为deprecated, 并将在未来的版本中移除。请尽快考虑升级为新的接口方式, 具体使用方式可以参考下文介绍以及demo工程示例。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES一起使用。更新模型时, 如果SDK版本号有更新, 请务必同时更新SDK, 旧版本的SDK可能无法正确适配新发布出来的RES。

## 快速开始

SDK在以下环境中测试通过

- aarch64(arm64), Ubuntu 16.04, gcc 5.3 (RK3399)
- Hi3559AV100, aarch64, Ubuntu 16.04, gcc 5.3
- Hi3519AV100, armv7l, HiLinux 4.9.37, (Hi3519AV100R001C02SPC020)
- armv7hf, Raspbian, (Raspberry 3b)
- aarch64, Raspbian, (Raspberry 4b)
- armv7hf, Raspbian, (Raspberry 3b+)
- armv7hf, Ubuntu 16.04, (RK3288)
- Bitmain se50 BM1684, Debian 9
- Rockchip rk3399pro, Ubuntu 18.04
- Rockchip rv1126, Debain 10
- Rockchip rk3568, Ubuntu 20.04
- Rockchip rk3588, Ubuntu 20.04

- Atlas200(华为官网指定的Ubuntu 18.04版本)
- Amlogic A311D, Ubuntu 20.04
- MLU220, aarch64, Ubuntu 18.04

## 安装依赖

### 依赖包括

- cmake 3+
- gcc 5.4 以上(需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.5 (可选)

**依赖说明：**树莓派 树莓派Raspberry默认为armv7hf系统，使用SDK包中名称中包含 armv7hf\_ARM\_的tar包。如果是aarch64系统，使用SDK包中名称中包含 aarch64\_ARM\_的tar包。

在安装前可通过以下命令查看是32位还是64位：

```
getconf LONG_BIT
32
```

**依赖说明：**比特大陆SE计算盒 需要安装SophonSDK V23.05.01及以上版本，SDK的默认安装位置为 /opt/sophon/，如SDK安装在自定义地址，需在CMakeList.txt中指定SDK安装地址：

```
**这里修改并填入所使用的SophonSDK路径**
set(EDGE_BMSDK_ROOT "{这里填写sdk路径}")
```

可通过命令 `bm-smi` 查看内部SDK和驱动的版本号（SophonSDK V23.05.01对应的内部SDK和驱动为0.4.6）。对于使用旧版BM1684 SDK或者低版本SophonSDK的用户，可参考**SophonSDK安装包**中的《LIBSOPHON 使用手册》先卸载旧版BM1684 SDK，安装、升级SophonSDK。

**依赖说明：**海思开发板 海思开发板需要根据海思SDK文档配置开运行环境和编译环境，SDK和opencv都需要在该编译环境中编译。NNIE1.2用 arm-himix200-linux交叉编译好的opencv，下载链接：<https://pan.baidu.com/s/13QW0ReeWx4ZwgYg4lretyw> 密码:yq0s。下载后修改SDK CMakeList.txt

**依赖说明：**RK3399Pro 所有用例基于 Npu driver版本1.7.1的RK3399pro开发板测试通过，SDK采用预编译模式，请务必确保板上驱动版本为1.7.1 查看RK3399Pro板上driver版本方法：`dpkg -l | grep 3399pro`

**依赖说明：**RV1109/RV1126 所有用例基于Rknn\_server版本1.7.3的RV1126开发板测试通过，SDK采用预编译模式，请务必确保板上驱动版本为1.7.3 查看RV1109/RV1126板上Rknn\_server版本方法：`strings /usr/bin/rknn_server | grep build`

**依赖说明：**RK3568 所有用例基于Rknn\_server版本1.2.0的RK3568开发板测试通过，查看RK3568板上Rknn\_server版本方法：`strings /usr/bin/rknn_server | grep build`

**依赖说明：**RK3588 RK3588开发板需要确保环境正确安装了RKNPU驱动，平台用例基于v0.8.0版本的RKNPU驱动测试通过，查看RK3588NPU驱动版本的方法：`sudo cat /sys/kernel/debug/rknpu/version`

**依赖说明：**晶晨A311D 所有用例基于晶晨A311D开发板测试通过，需要驱动版本为 6.4.4.3（下载驱动请联系开发版厂商）查看晶晨A311D开发板驱动版本方法：`dmesg | grep Galcore`

**依赖说明：**英特尔iGPU 用户在使用英特尔iGPU SDK前，需要根据英特尔**官方文档**提前安装好英特尔集成显卡驱动以及相关基础软件环境，安装完成后通过 `clinfo` 指令确认OpenCL能够正常识别到集成显卡信息，正确识别集显情况下clinfo指令输出参考如下：

```
root@baidu-QiitarM38-N000:~# clinfo
Number of Platforms: 1
Platform Name: Intel(R) OpenCL HD Graphics
Platform Vendor: Intel(R) Corporation
Platform Version: OpenCL 3.0
Platform Profile: FULL_PROFILE
Platform Extensions: cl_khr_byte_addressable_store cl_khr_device_uuid cl_khr_fp16 cl_khr_global_int32_base_atomics cl_khr_global_int32_extended_atomics cl_khr_licd cl_khr_local_int32_base_atomics cl_khr_local_int32_extended_atomics cl_intel_command_queue_families cl_intel_subgroups cl_intel_required_subgroup_size cl_intel_subgroups_short cl_khr_spir cl_intel_accelerator cl_intel_driver_diagnostics cl_khr_priority_hints cl_khr_throttle_hints cl_khr_create_command_queue cl_intel_subgroups_clr cl_intel_subgroups_smg cl_khr_il_program cl_intel_mem_force_host_memory cl_khr_subgroup_extended_types cl_khr_subgroup_non_uniform_vote cl_khr_subgroup_ballot cl_khr_subgroup_non_uniform_arithmetic cl_khr_subgroup_shuffle cl_khr_subgroup_shuffle_relative cl_khr_subgroup_clustered_reduce cl_intel_device_attribute_query cl_khr_suggested_local_work_size cl_intel_split_work_group_barrier cl_khr_fp64 cl_khr_subgroups cl_intel_spirv_device_side_ova_motion_estimation cl_intel_spirv_media_block_io cl_intel_spirv_subgroups cl_khr_spirv_no_integer_arry_decoration cl_intel_unified_shared_memory cl_khr_mimgp_image cl_khr_mimgp_image_writes cl_intel_platform_yuv cl_intel_motion_estimation cl_intel_device_side_ova_motion_estimation cl_intel_advanced_motion_estimation cl_khr_int64_base_atomics cl_khr_int64_extended_atomics cl_khr_image2d_from_buffer cl_khr_depth_images cl_khr_3d_image_writes cl_intel_media_block_io cl_intel_v_api_media_sharing cl_intel_sharing_format_query cl_khr_pci_bus_info
Platform host timer resolution: 1ns
Platform Extensions Function suffix: INTEL

Number of Devices: 1
Device Name: Intel(R) UHD Graphics 630 [0x9bc8]
Device Vendor: Intel(R) Corporation
Device Vendor ID: 0x8086
Device Version: OpenCL 3.0 NEO
Driver Version: 22.53.05242.13
Device OpenCL C Version: OpenCL C 1.2
Device Type: GPU
Device Profile: FULL_PROFILE
Device Available: Yes
Compiler Available: Yes
Linker Available: Yes
Max compute units: 24
Max clock frequency: 1150MHz
Device Partition: (Core)
```



### 使用序列号激活 请在官网获取序列号

**纯离线服务说明**  
发布纯离线服务，将训练完成的模型部署在本地，离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体机方案等专项适配硬件上。通过API、SDK进一步集成，灵活适应不同业务场景。

[发布前服务](#) [控制台](#)

服务器 通用小型设备 专项适配硬件

SDK API

此处发布、下载的SDK为未授权SDK，需要前往控制台[获取序列号](#)激活后才能正常使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标检测	134318-V1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无惧压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无惧压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
		ARMv7-M Linux	基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>
		ARMv8-M Linux	基础版	已发布		

SDK内bin目录下提供预编译二进制文件，可直接运行(二进制运行详细说明参考下一小节)，用于图片推理和模型http服务，在二进制参数的serial\_num(或者serial\_key)处填入序列号可自动完成联网激活(请确保硬件首次激活时能够连接公网，如果确实不具备联网条件，需要使用纯离线模式激活，请下载使用百度智能边缘控制台纳管SDK)

```

**SDK内提供的一些二进制文件，填入序列号可完成自动激活，以下二进制具体使用说明参考下一小节**
./edgekit_serving --cfg=./edgekit_serving.yml
./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}
./easyedge_serving {res_dir} {serial_key} {host} {port}

```

如果是基于源码集成，设置序列号方法如下

```
global_controller()->set_licence_key("")
```

默认情况下(联网激活或者离线激活的场景)，按照上述说明正确设置序列号即可，如果是实例数鉴权模式(请在百度智能云控制台再次确认自己的序列号是实例数鉴权模式，仅实例数鉴权需要进行下面的变量或者源码设置) 实例数鉴权环境变量设置方法

```
export EDGE_CONTROLLER_KEY_AUTH_MODE=2
export EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=30
```

实例数鉴权源码设置方法

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2)
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 300)
```

基于预编译二进制测试图片推理和http服务 测试图片推理 模型资源文件默认已经打包在开发者下载的SDK包中。

```

对于硬件使用为：Intel Movidius Myriad2 / Myriad X / IGPU on Linux x86_64 / armv7hf / aarch64，在编译或运行demo程序前执行以下命令：
source ${cpp_kit位置路径}/thirdparty/opencv/bin/setupvars.sh
或者执行
source ${cpp_kit位置路径}/thirdparty/opencv/setupvars.sh (opencv-2022.1+)
如果SDK内不包含setupvars.sh脚本，请忽略该提示

```

运行预编译图片推理二进制，依次填入模型文件路径(RES文件夹路径)、推理图片、序列号(序列号尽首次激活需要使用，激活后可不用填序列号也能运行二进制)

```

**./easyedge_image_inference {model_dir} {image_name or image_directory} {serial_num}**
LD_LIBRARY_PATH=./lib ./easyedge_image_inference ../../RES/xxx/cat.jpeg "1111-1111-1111-1111"

```

demo运行效果：





```
> ./easyedge_image_inference ../.././RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

启动http服务 bin目录下提供编译好的启动http服务二进制文件，可直接运行

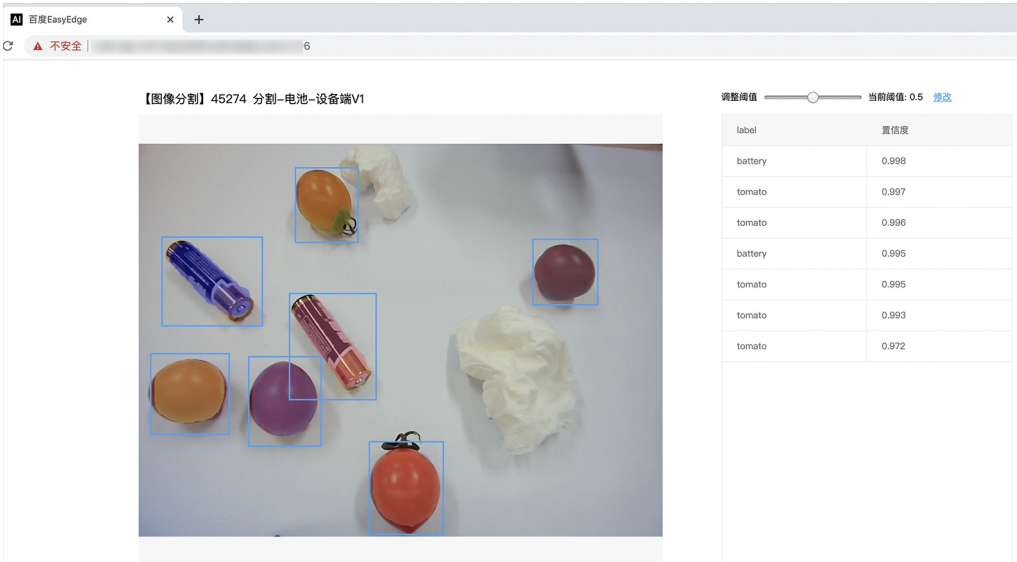
```
**推荐使用 edgekit_serving 启动模型服务**
LD_LIBRARY_PATH=./lib ./edgekit_serving --cfg=./edgekit_serving.yml

**也可以使用 easyedge_serving 启动模型服务**
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
**LD_LIBRARY_PATH=./lib ./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401**
```

后，日志中会显示

```
HTTP(or Webservice) is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试，网页右侧会展示模型推理结果



【图像分割】45274 分割-电池-设备端V1

调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

同时，可以调用HTTP接口来访问服务。

**请求http服务** 以图像预测场景为例(非语义分割模型场景，语义分割请求方式参考后面小节详细文档)，提供一张图片，请求模型服务的示例参考如下demo

```
python
```

```
c#
```

```
C++
```

```
java
```

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())

print(result)
```

关于http接口的详细介绍参考下面集成文档http服务章节的相关内容

### 集成文档

使用该方式，将运行库嵌入到开发者的程序当中。 **编译demo项目** SDK src目录下有完整的demo工程，用户可参考该工程的代码实现方式将SDK集成到自己的项目中，demo工程可直接编译运行：

```
cd src
mkdir build && cd build
cmake .. && make
./easiedge_image_inference {模型RES文件夹} {测试图片路径}
**如果是NNIE引擎，使用sudo运行**
sudo ./easiedge_image_inference {模型RES文件夹} {测试图片路径}
```

(可选) SDK包内一般自带opencv库，可忽略该步骤。如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。 SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

对于硬件使用为Intel Movidius Myriad2 / Myriad X 的，如果宿主机找不到神经计算棒Intel® Neural Compute Stick，需要执行以下命令添加USB Rules：

```
cp ${cpp_kit位置路径}/thirdparty/openvino/deployment_tools/inference_engine/external/97-myriad-usbboot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig
```

### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置运行参数
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

对于口罩检测模型，将 `EdgePredictorConfig config` 修改为 `PaddleMultiStageConfig config` 即可。

口罩检测模型请注意输入图片中人脸大小建议保持在 88到9696像素之间，可根据场景远近程度缩放图片后再传入SDK。

**SDK参数配置** SDK的参数通过 `EdgePredictorConfig::set_config` 和 `global_controller()->set_config` 配置。 `set_config` 的所有key在 `easyedge_xxxx_config.h` 中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过 `EdgePredictorConfig::set_config` 设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过 `global_controller()->set_config` 设置

以序列号为例，KEY的说明如下：

```

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

使用方法如下：

```

EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");

```

具体支持的运行参数可以参考开发工具包中的头文件的详细说明。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR, HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};
```

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域，0代表非目标区域
```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

## 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

- 接口

class `VideoDecoding` :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct `VideoConfig`

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};         // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;           // frame存储为视频文件的路径
    bool save_all{false};           // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于`/dev/video0`的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

**注意：**1.如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。2.使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

## 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```

EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");

```

## 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

## http服务

1. 开启http服务 http服务的启动参考`demo_serving.cpp`文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

## 2. http接口详细说明 http 请求方式一：无额外编码 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (图片测试, 针对图像分类、物体检测、实例分割等模型)

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

Python请求示例 (图片测试, 仅针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```

import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post('http://127.0.0.1:24401/',
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果

```

Python请求示例 (视频测试, 注意: 区别于图片预测, 需指定Content-Type; 否则会调用图片推理接口)

```

import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data).json()

```

http 请求方法二：json格式，图片传base64格式字符串 HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 图像分类网络：body中请求示例

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 物体检测和实例分割网络：Body请求示例：

```
{
  "image": "<base64数据>",
  "threshold": 0.3
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

- 语义分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情（语义分割由于模型特殊性，不支持设置threshold值，设置了也没有意义）：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部

Python请求示例 (非语义分割模型参考如下代码)



```
import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()
```

Python 请求示例 (针对语义分割模型, 同其他CV模型不同, 语义分割模型输出为灰度图)

```
import base64
import requests
def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出, 可将api返回结果保存为灰度图, 每个像素值代表该像素分类结果
if __name__ == '__main__':
    main()
```

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728,
      "mask": "...", // 图像分割模型字段
      "trackId": 0, // 目标追踪模型字段
    },
  ]
}
```

#### 其他配置

##### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



## FAQ

### 1. 如何处理一些 undefined reference / error while loading shared libraries?

如：`./easyedge_demo: error while loading shared libraries: libeasyedge.so.1: cannot open shared object file: No such file or directory` 这是因为二进制运行时ld无法找到依赖的库。如果是正确cmake && make 的程序，会自动处理好链接，一般不会出现此类问题。

遇到该问题时，请找到具体的库的位置，设置LD\_LIBRARY\_PATH。

示例一：`libverify.so.1: cannot open shared object file: No such file or directory` 链接找不到libveirfy.so文件，一般可通过 `export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/lib` 解决(实际冒号后面添加的路径以libverify.so文件所在的路径为准)

示例二：`libopencv_videoio.so.4.5: cannot open shared object file: No such file or directory` 链接找不到libopencv\_videoio.so文件，一般可通过 `export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:/thirdparty/opencv/lib` 解决(实际冒号后面添加的路径以libopencv\_videoio.so所在路径为准)

### 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

3. 如何将我的模型运行为一个http服务？目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

### 4. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

### 5. 运行SDK报错 Authorization failed

情况一：日志显示 `Http perform failed: null respond` 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示`failed to get/check device id(xxx)`或者`Device fingerprint mismatch(xxx)` 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

### 6. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

7. 运行NNIE引擎报错 `std::bad_alloc` 检查开发板可用内存，一些比较大的网络占用内存较多，推荐内存500M以上

8. 运行二进制时，提示 `libverify.so cannot open shared object file`

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 编译时报错：`file format not recognized` 可能是因为复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中，再解压缩、编译

## 端云协同服务说明

### 服务简介

EasyDL端云协同服务由EasyEdge端与边缘AI服务平台提供、基于百度智能边缘构建，能够便捷地将EasyDL定制模型的推理能力拓展至应用现场，提供临时离线、低延时的计算服务。

「云管理，端计算」的端云协同服务，具体包括：

- 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
- 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
- 联网状态下在平台管理设备运行状态、资源利用率

目前通用小型设备的应用平台支持Linux-ARM，具体使用流程请参考下方文档。

### 使用流程

#### Step 1 发布端云协同部署包

在[我的部署包](#)页面点击「发布端云协同部署包」

端云协同服务 > 我的部署包

[端云协同服务说明](#) 点击收起

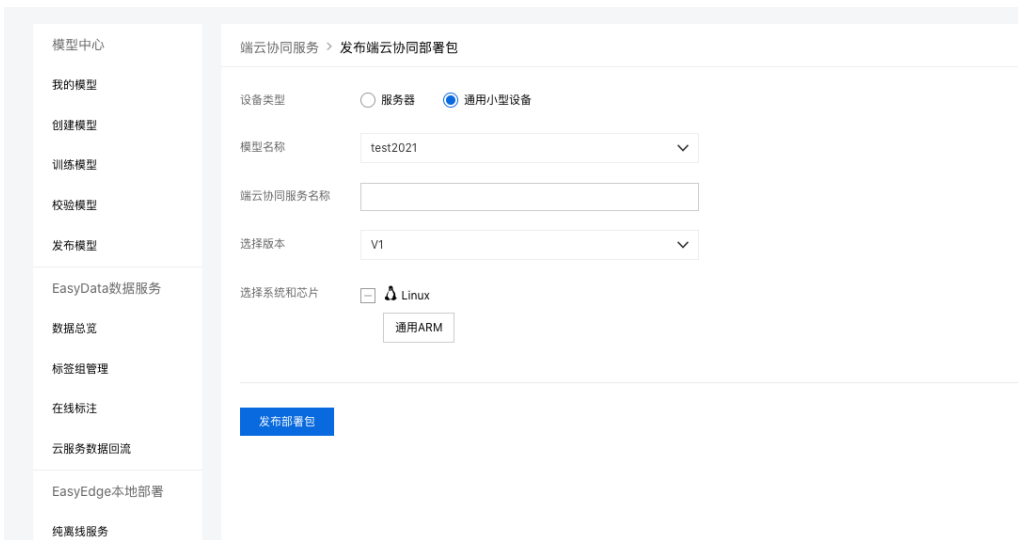
1. 在可视化界面轻松实现模型部署包在边缘设备上的集成、版本更新
2. 断网状态下模型离线计算（http服务，可调用与公有云API功能相同的接口）
3. 联网状态下在平台管理设备运行状态、资源利用率

具体使用流程如下：

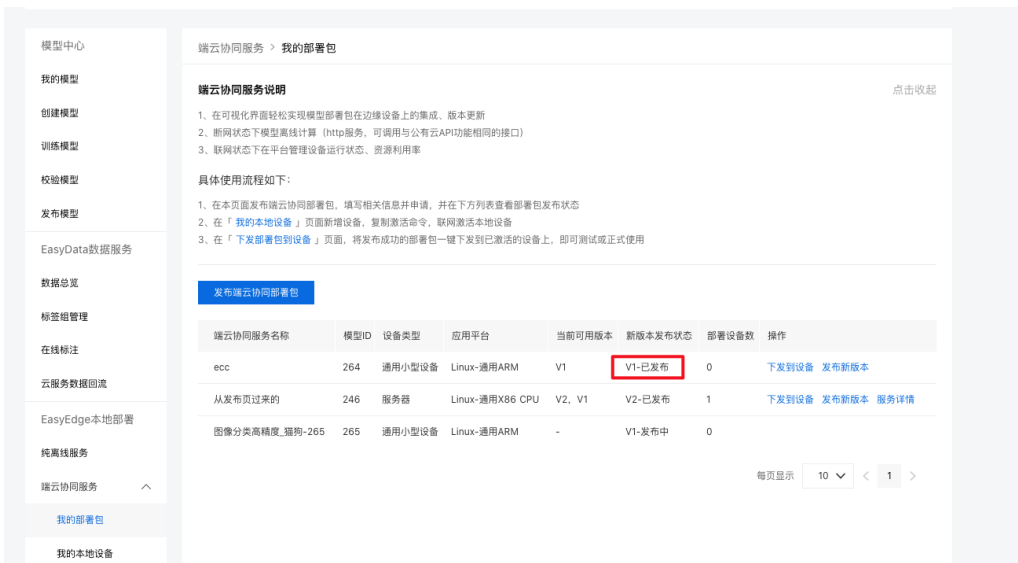
1. 在本页面发布端云协同部署包，填写相关信息并申请，并在下方列表查看部署包发布状态
2. 在「我的本地设备」页面新增设备，复制激活命令，联网激活本地设备
3. 在「下发部署包到设备」页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用

端云协同服务名称	模型ID	设备类型	应用平台	当前可用版本	新版本发布状态	部署设备数	操作
<p>暂无可用数据 请稍后再试</p>							

填写服务名称，选择模型版本并提交发布



在列表查看部署包发布状态



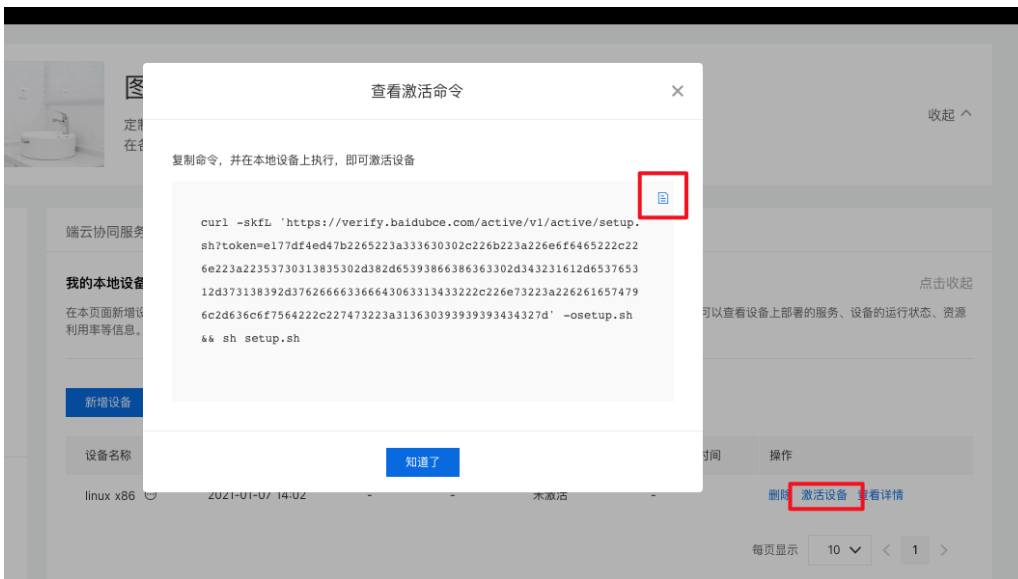
### Step 2 新增设备并激活

在**我的本地设备**页面新增设备





在列表中，点击设备对应的「激活设备」操作，复制激活命令并在本地设备上执行即可

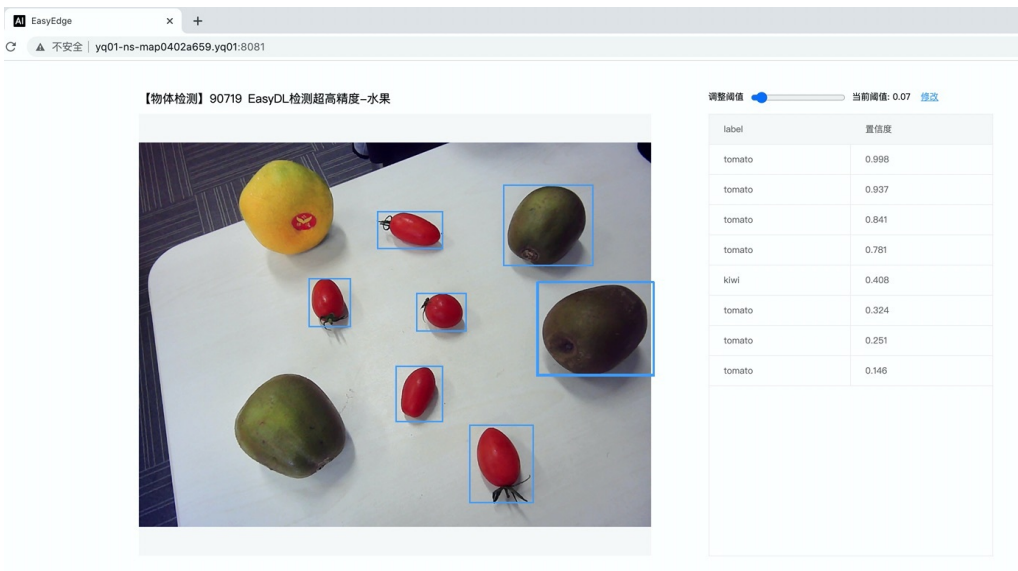


### Step 3 下发部署包到设备，在本地调用

在[下发部署包到设备](#)页面，将发布成功的部署包一键下发到已激活的设备上，即可测试或正式使用



部署包下发成功之后，会在本地启动一个HTTP推理服务。在浏览器中输入 `http://{设备ip}:{服务端口, 默认8080}`，即可预览效果：



具体接口调用说明请参考文档 [SDK - HTTP服务调用说明](#)

### 云端管理说明

### 模型部署包管理

在[我的部署包](#)页面可以进行已发布的模型部署包的管理。

### 发布及更新模型版本

点击「发布新版本」操作即可快速发布对应模型ID下的新版本。同一模型ID下已发布的模型版本均会显示在列表的「当前可用版本」中。



新版本发布成功后，即可在「下发部署包到设备」页面或当前服务的「服务详情」页面，将新版本下载到本地设备上。



## 管理模型已部署的设备

在上述的「服务详情」页面，可以查看并管理当前服务已部署的设备，包括移除设备、将服务下发到更多的设备等。

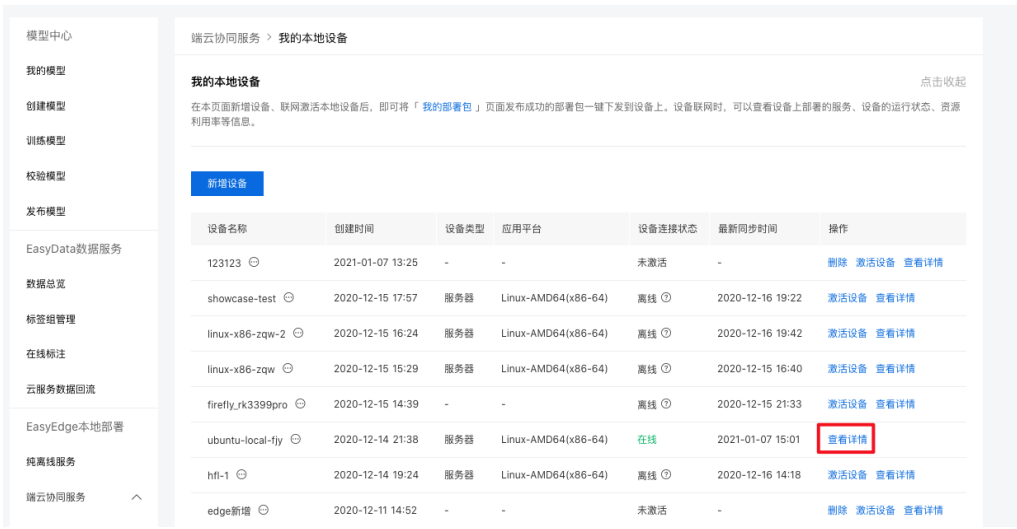


## 本地设备管理

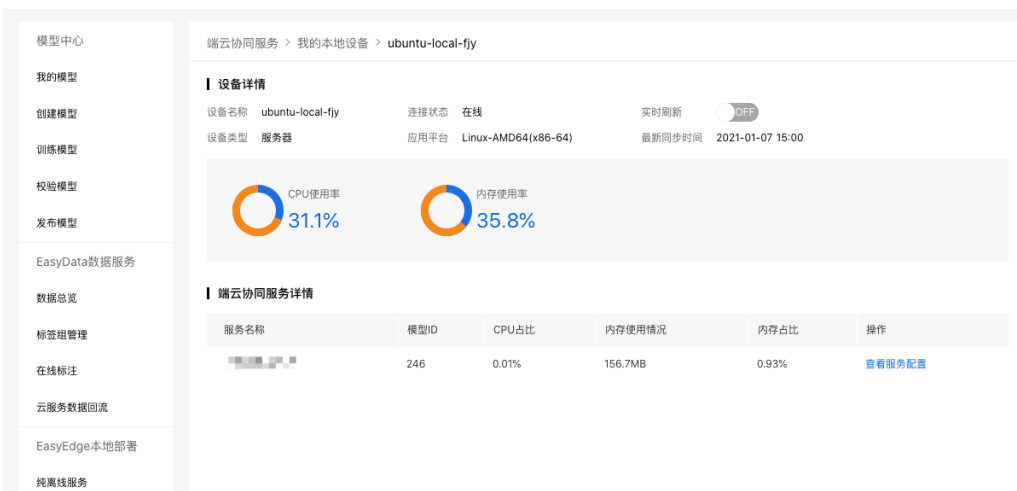
在[我的本地设备](#)页面可以进行所有本地设备的管理。

## 查看单台设备的运行状态

点击单台设备的「服务详情」，可查看设备上运行的多个服务及设备状态：



设备详情会展示当前设备的最新同步时间，以及CPU使用率、内存使用率等。服务列表则展示了当前设备上部署服务的运行情况和资源占用情况



🔗 智能边缘控制台-单节点版

🔗 EasyEdge 智能边缘控制台-单节点版 IEC

EasyEdge Intelligent Edge Console（以下简称IEC）是EasyEdge推出的边缘设备管理的本地化方案。可以运行于多种架构、多系统、多类型的终端之上。通过IEC，用户可以方便地在本地进行

- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活，服务管理
- 接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关事件
- 硬件信息的可视化查看

支持的系统+CPU架构包括：

- Windows x86\_64 (Windows 7 ~ Windows 10, 暂不支持Windows 11)
- Linux x86\_64 / arm32 / arm64

支持各类常见的AI加速芯片，包括：

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列



- 其他EasyDL/EasyEdge/BML支持的AI芯片

完整列表可参考[这里](#)

## Release Note

注意：2.0.0之后，默认以系统服务形式安装iec，无法兼容1.x版本的iec

版本号	发布时间	更新说明
2.2.0	2022-10-27	新增onvif/gb28181支持；完善端云通信逻辑
2.0.0	2022-03-22	支持连接中心节点IECC；支持以系统服务安装
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	IEC 第一版！

## 快速开始

从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#)：intel、AMD的64位x86\_84 CPU
- [Linux amd64](#)：intel、AMD的64位x86\_84 CPU
- [Linux arm](#)：树莓派等32位的ARM CPU
- [Linux arm64](#)：RK3399、飞腾等64位的ARM CPU

或者从纯离线服务管理页可下载智能边缘控制台



您也可以通过先安装多节点版本IECC，通过中心节点来自动连接安装边缘节点。

**Linux 安装** 解压缩之后，目录结构如下

```
0 EasyEdge-IEC-v2.0.0-linux-amd64 > tree .
.
├── easyedge-iec
├── easyedge-iec-setup.sh
├── etc
│   ├── easyedge-iec.service-conf.init.d
│   ├── easyedge-iec.service-conf.systemd
│   ├── easyedge-iec.service-conf.upstart
│   ├── easyedge-iec.service.yml
│   └── easyedge-iec.yml
└── readme.txt

1 directory, 8 files
```

以系统服务形式安装（推荐）以root用户运行./easyedge-iec-setup.sh install 即可

```
[setup]: sudo could not be found
[setup]: Start to install IEC...
[setup]: + bash -c "cp easyedge-iec /usr/sbin/easyedge-iec"
[setup]: + bash -c "chmod +x /usr/sbin/easyedge-iec"
[setup]: + bash -c "cp etc/easyedge-iec.service.yml /etc/easyedge-iec/easyedge-iec.yml"
[setup]: + bash -c "cp etc/easyedge-iec.service-conf.init.d /etc/init.d/easyedge-iec"
[setup]: + bash -c "chmod +x /etc/init.d/easyedge-iec"
[setup]: Install IEC success!
[setup]: + bash -c "service easyedge-iec start"
Starting easyedge-iec: success
[setup]: Start to check IEC status...
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: IEC status: OK!
[easyedge-iec]: default configure file: /etc/easyedge-iec/easyedge-iec.yml
[easyedge-iec]: default log file: /var/log/easyedge-iec/easyedge-iec.log
[easyedge-iec]: service usage: service easyedge-iec { start | stop }
[setup]: Done!
```

- 日志：`/var/log/easyedge-iec/easyedge-iec.log`
- 系统配置：`/etc/easyedge-iec/easyedge-iec.yml`
- 服务启动/停止：`service easyedge-iec { start | stop }` (不同操作系统内可能不同，具体命令参考安装日志)

**自定义安装 (不推荐)** 自定义安装方法仅限于 安装脚本无法识别的情况。

- 拷贝 `./EasyEdge-IEC-v2.0.0/` 整个目录至自定义文件夹，如 `/opt/EasyEdge-IEC`
- 进入到 `/opt/EasyEdge-IEC`
- 通过 `nohup` 等方法运行 `./easyedge-iec-linux-{您的系统架构}` `amd64: intel、AMD的64位x86_64 CPU` `arm: 树莓派等32位的ARM CPU *` `arm64: RK3399、飞腾等64位的ARM CPU`
- 日志：`./log/easyedge-iec.log`
- 系统配置：`./easyedge-iec.yml`

## Windows 安装

解压缩之后，安装目录如下所示：

```
0 tmp2 > tree EasyEdge-IEC-v2.0.0-windows-amd64
EasyEdge-IEC-v2.0.0-windows-amd64
|--- easyedge-iec.exe
|--- easyedge-iec-setup.bat
|--- etc
|   |--- easyedge-iec.yml
|   |--- readme.txt
|
1 directory, 4 files
```

打开命令行 (非powershell) 运行 `easyedge-iec-setup.bat install`。

如果遇到hang住的情况，可修改命令行配置 启动之后，打开浏览器，访问 `http://{设备ip}:8702/easyedge/iec` 即可：

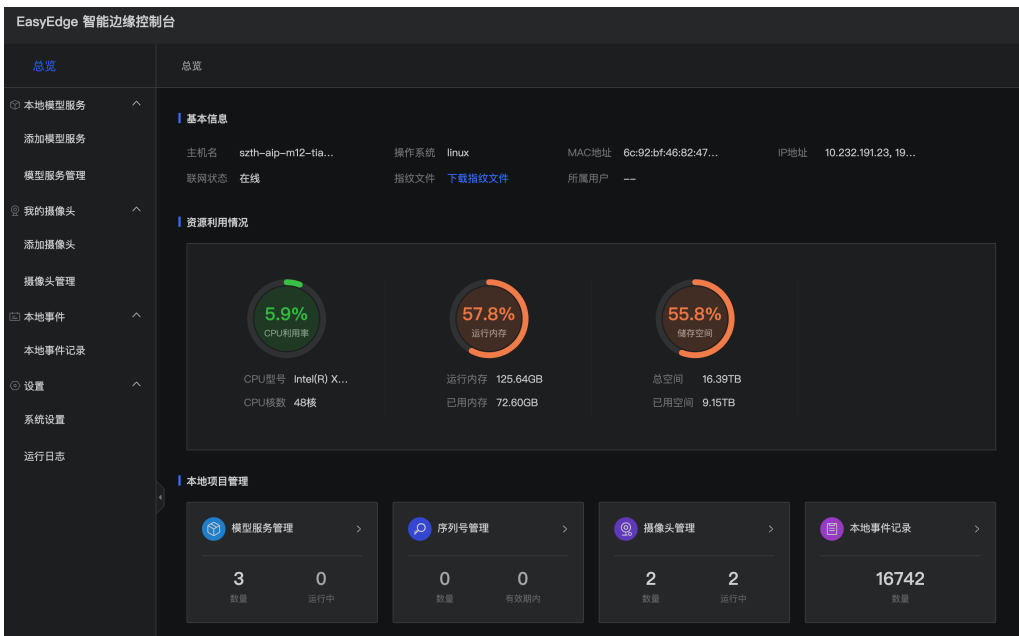




启动之后，打开浏览器，访问 `http://{设备ip}:8702/easyedge` 即可：

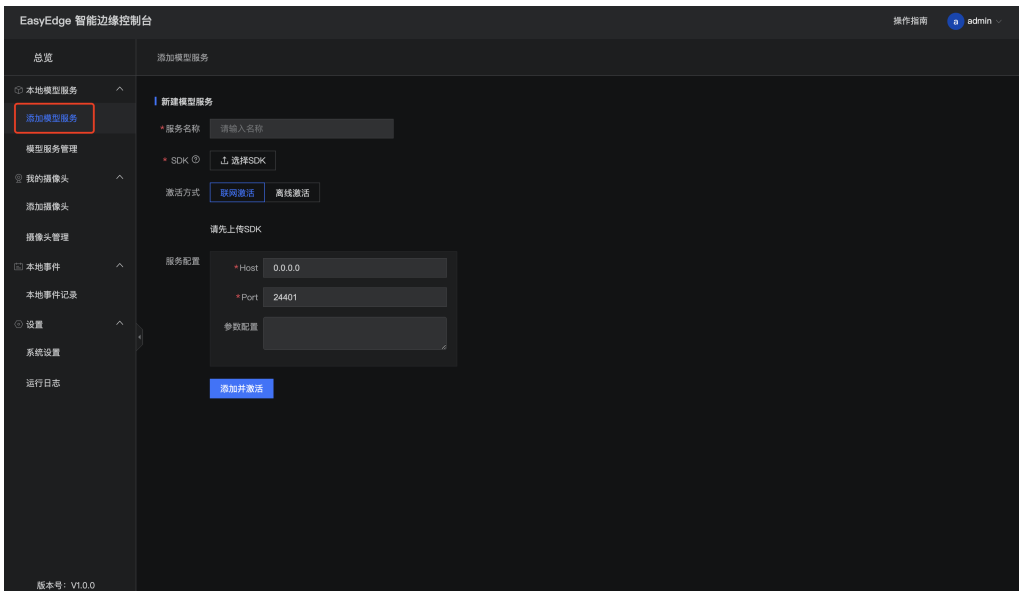


默认用户名密码为 admin / easyedge



## 功能使用说明

①**添加模型服务** 首先，点击导航栏的「本地模型服务」-「添加模型服务」。在页面中定义服务名称后，将已经下载好的Linux/Windows版本的SDK与IEC关联。关联完后可按两种激活方式，激活使用SDK。



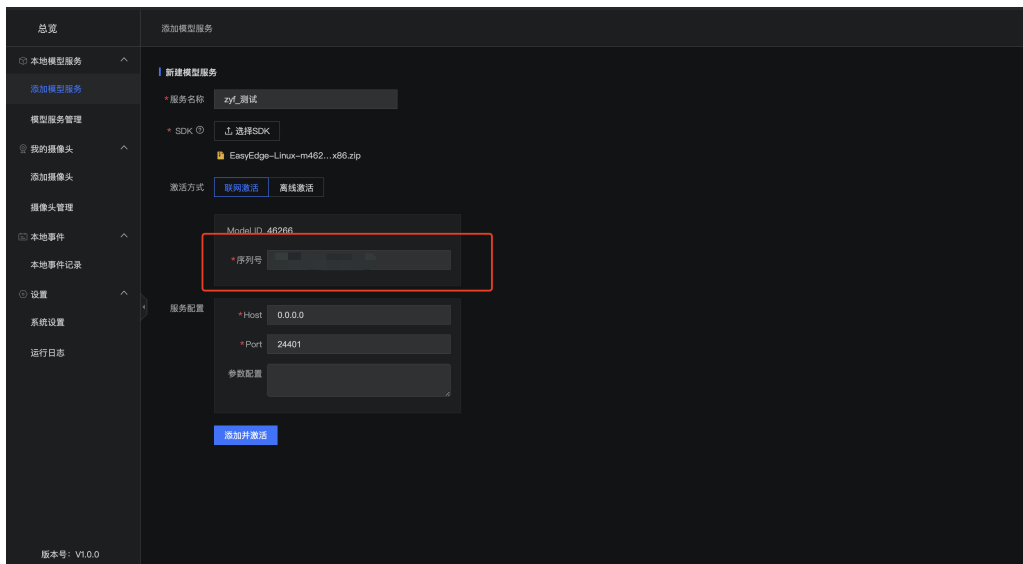
部分SDK需要提前安装系统依赖，如TRT等，具体请参考EasyDL/BML/EasyEdge SDK使用文件中的环境依赖安装说明

## 联网激活

1. 在关联SDK完成后，需要在百度智能云控制台对应部署方式管理页中新增测试序列号或购买正式序列号。（图中以服务器版SDK为例）



2. 再在IEC中填入所申请的序列号



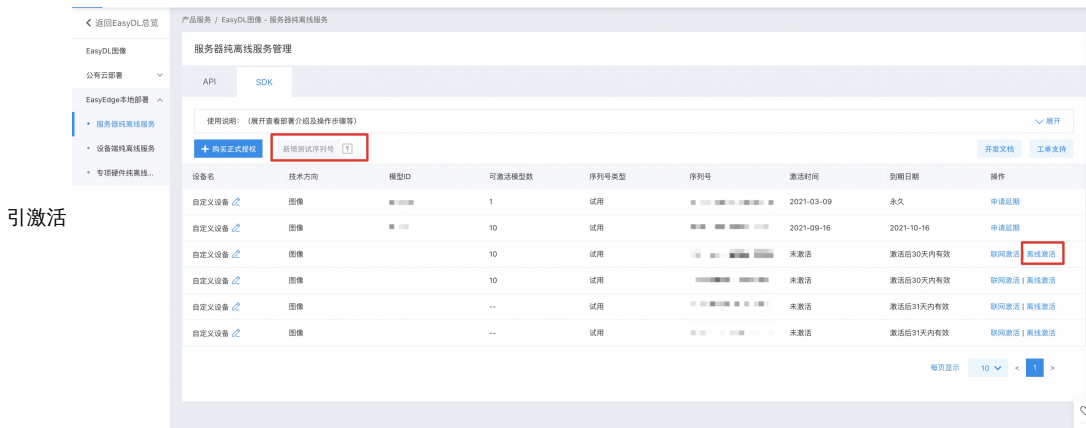
3. 配置服务，在服务端口不冲突占用的情况下，使用默认即可
4. 添加并激活

### 离线激活

1. 在IEC总览页面下载「指纹文件」

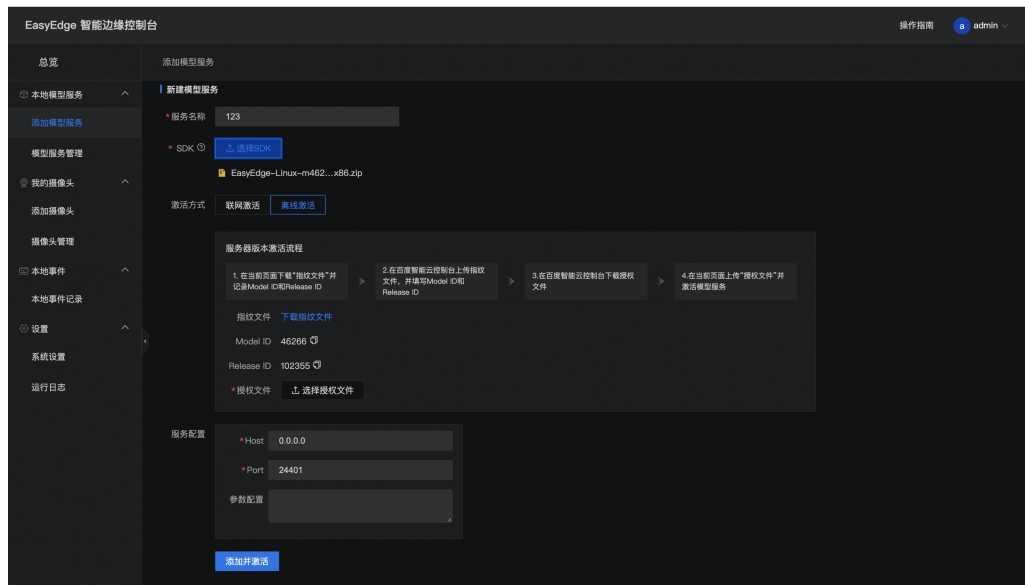


2. 在百度智能云的**控制台**中找到SDK对应的管理列表，图中以服务器SDK为例。申请序列号后，点击对应序列号尾部的「离线激活」操作，按指



### 引激活

3. 在IEC的添加模型服务页面，上传下载好的授权文件，完成激活

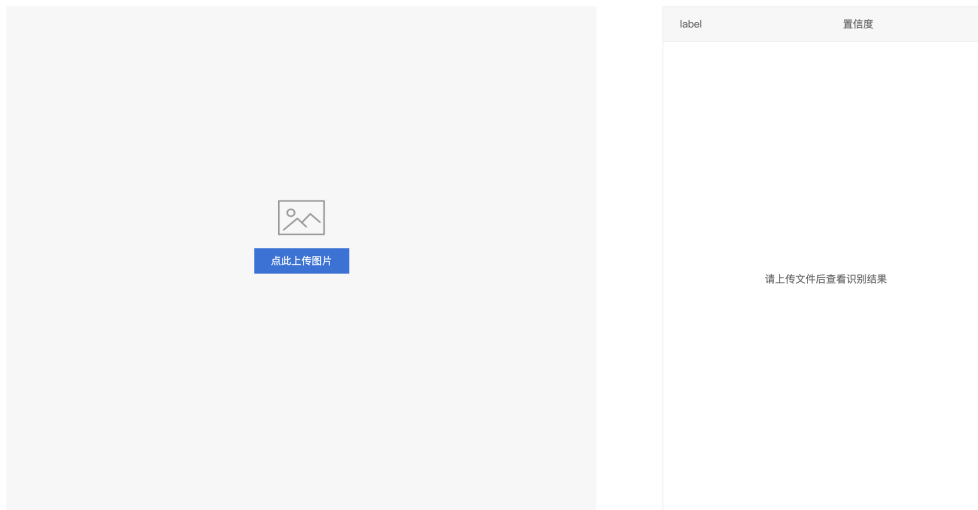


激活完成后即可在「模型服务管理」列表中启动服务，使用后续的操作栏功能。

### 体验本地demo

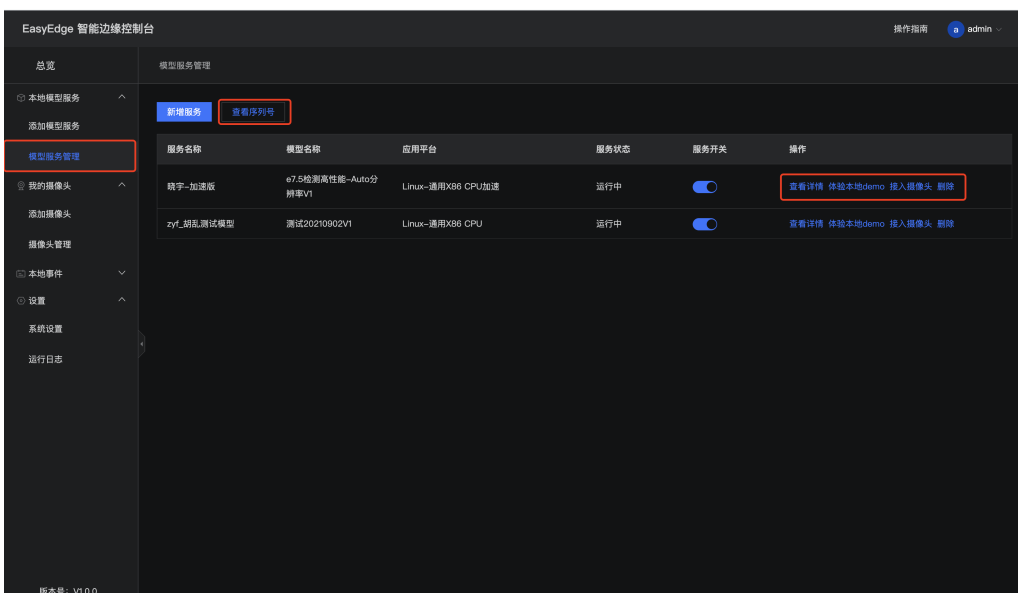
点击「本地demo体验」即可在立即上传图片进行预测

【物体检测】97741 e7.5检测高性能-Auto分辨率V1



### 接入摄像头

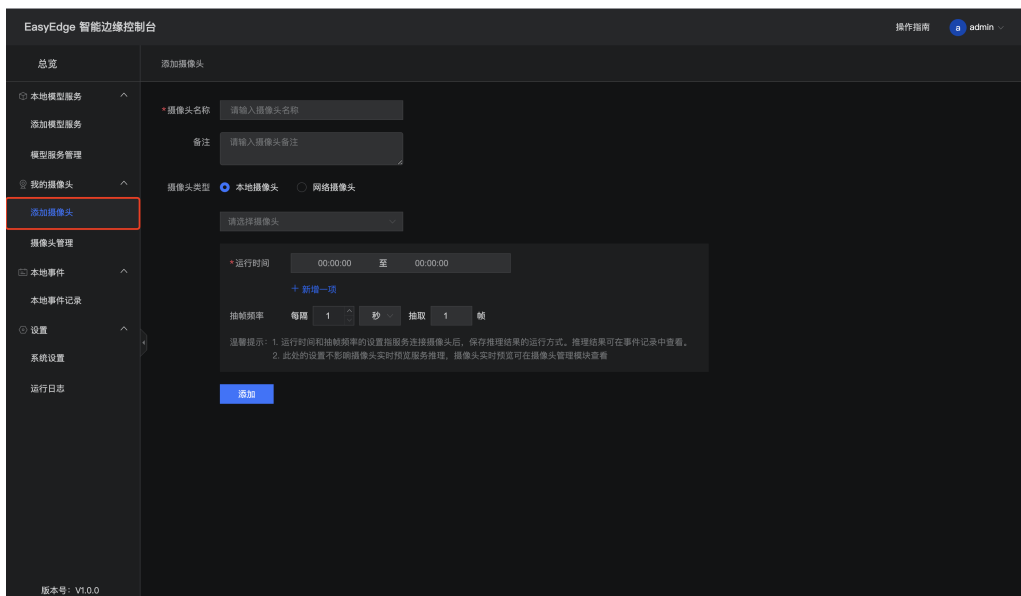
使用接入摄像头功能首先需要添加摄像头，请参考第②步，完成后按照第③步操作 注：服务启动后也可参考「模型发布」模块的技术文档进行开发使用，本文档主要介绍IEC使用功能



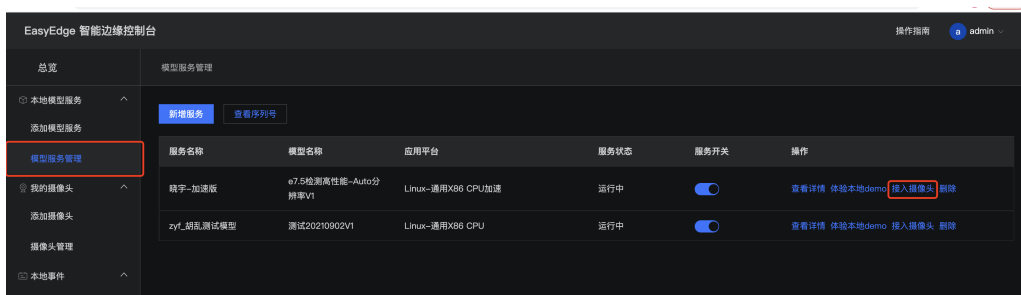
激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

②添加摄像头 导航栏点击「我的摄像头」-「添加摄像头」，定义摄像头名称、备注后即可添加摄像头。支持本地摄像头和网络摄像头。摄像头添加成功后即可设置摄像头的运行时间和频率



③摄像头接入模型服务预测 点击「本地模型服务」-「模型服务管理」中，所需接入预测的服务的「接入摄像头」



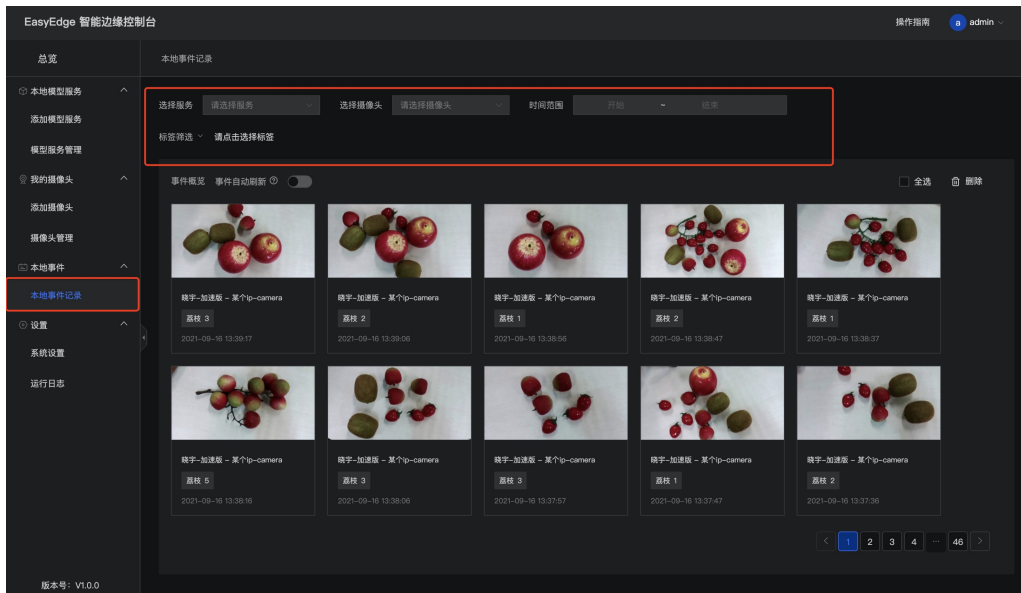
在弹出的弹窗中选择第②

步中添加的摄像头，此时点击确认即可在「摄像头管理」中的实时预览功能中查看摄像头预测结果，识别结果默认不保存。如需保存识别结果，可设置对应的「本地事件触发条件」，根据标签和置信度，将识别结果保存至本地事件记录当中。设置多个标签条件时，IEC会以“或”的逻辑来将

所有满足条件的识别结果保存



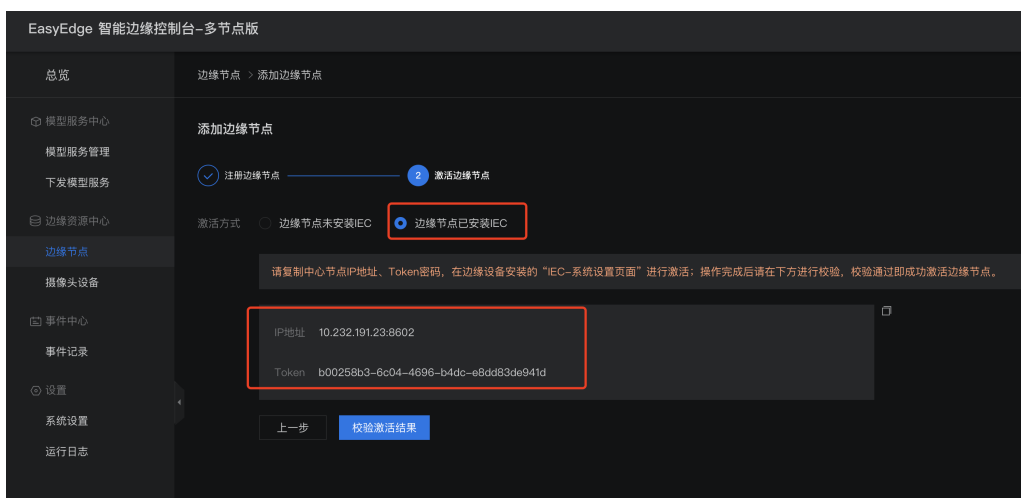
④本地事件 点击导航栏「本地事件记录」，可通过服务名称、摄像头名称、事件记录的时间、标签及置信度来筛选识别结果查看，多个标签及置信度同样也是“或”的逻辑记录。如有想要删除的事件数据可选择后删除，全选为本页全选。



#### ⑤ 连接到智能边缘控制台-多节点版 (IECC)

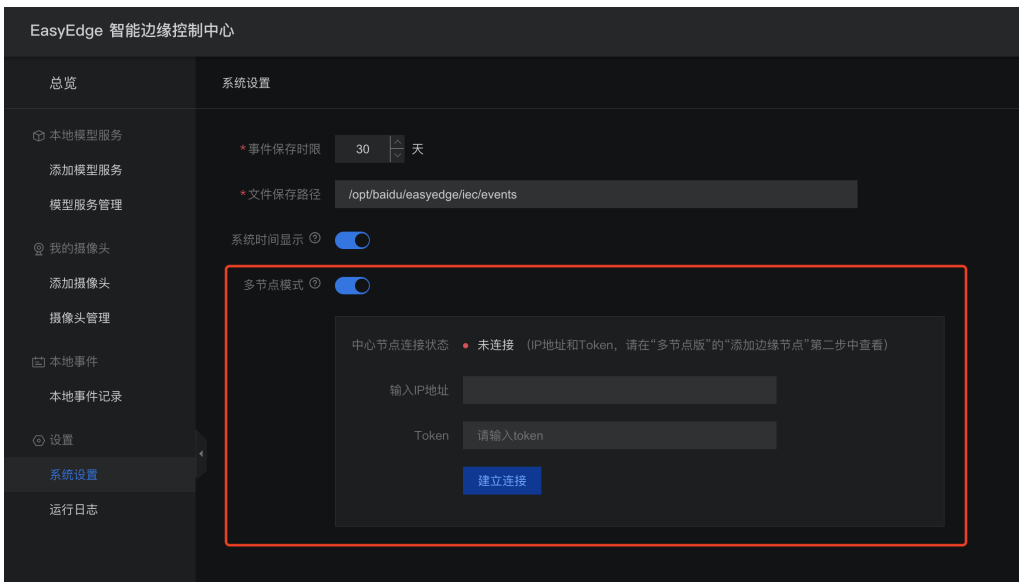
与中心节点连接之后，边缘节点主程序版本会自动随控制中心版本升级。（>2.0.0）

- Step 1 在IECC中添加边缘节点，选择「边缘节点已安装IEC」，并记录IP地址与Token



- Step 2 在IEC的系统设置中打开多节点模式，并填入刚才记录的IP地址与Token，点击建立连接





- 连接完成后即可在中心节点IECC去监控/管理/应用在边缘节点上的IEC

### 配置项\*

配置文件etc/easyedge-iec.yml中有关于IEC的各项配置说明，一般无需修改，请确保理解配置项含义之后，再做修改。

```
##### IEC系统配置
##### ----- 高级配置一般无需修改 -----
##### !!!注意!!! 请确保理解配置项含义后再做修改
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
# role: edge
# 硬件利用率刷新时间间隔： 过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# 事件监测触发扫描周期
eventTriggerIntervalSecond: 10
# IEC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: default
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge)
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
toFile: yes
loggingFile: /var/log/easyedge-iec/easyedge-iec.log
# 0:info; -1:debug; -2:verbose
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8702
listenHost: 0.0.0.0

sdk:
# GPU SDK所使用的cuda版本：9 / 10 / 10.2 / 11.0 / 11.1。请安装完cuda之后，这设置正确的版本号。
cudaVersion: 10.2
# AI服务启动时，额外配置的 LD_LIBRARY_PATH(linux) 或者 PATH(windows)
libPath: ./
# AI服务启动时，额外配置的其他环境变量。
ENVs:
EDGE_CONTROLLER_KEY_LOG_BRAND: EasyEdge
##### EDGE_CONTROLLER_KEY_XXX: XXXX
```

```

commu:
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2

##### 数据库相关配置
db:
sqliteDbFile: /var/lib/easyedge-iec/easyedge-iec.db
hubDbFile: /var/lib/easyedge-iec/easyedge-iec.hub.db
eventDbFile: /var/lib/easyedge-iec/easyedge-event.db
fileServerDbFile: /var/lib/easyedge-iec/easyedge-fileserver.hub.db
nodeMonitorDbFile: /var/lib/easyedge-iec/easyedge-nodemonitor.hub.db

##### 推流相关配置
mediaserver:
flvPort: 8715
rtmpPort: 8716

##### 视频流相关配置
edgestream:
logLevel: -1
listenHost: 127.0.0.1
listenPort: 8710
# 摄像头预览: 识别结果绘制延迟消失
renderExtendFrames: 10
# 预测队列大小: 如果设置为60, 当摄像头fps=30时, 视频延迟约为2秒。降低inferenceQueueSize可以降低预览延迟, 但是根据硬件的计算情况, 可能导致模型推理速度跟不上, 没有识别结果, 不建议设置太低
inferenceQueueSize: 60
videoEncodeBitRate: 400000
# 视频采样 & 视频实时预览分辨率设置
# 0: auto, 1: 1080p, 2: 720p, 3: 480p, 4: 360p, 5: 240p
resolution: 0
# 内置多媒体服务配置
# port设为0表示关闭
mediaServerHost: 127.0.0.1
mediaServerFlvPort: 8713
mediaServerRtmpPort: 8714
mediaServerRtspPort: 0

```

**FAQ 启动服务后, 进程中出现两个easyedge-iec进程** 这是正常现象, IEC通过守护进程的方式来完成更新等操作。

**启动服务时, 显示端口被占用port already been used** 通过修改 easyedge-iec.yml文件的配置后, 再重新启动服务。

**安装服务时, 报错permission denied** 请以管理员身份运行安装程序。

**中心节点重启后, 边缘节点IEC一直离线** 中心节点短时间的离线, 边缘节点会自动重连。如果中心节点已经恢复在线, 边缘节点长时间未自动连接上, 可通过边缘节点iec的方法来重新连接 (右上角 admin - 重启系统)

**IEC 是否有Android / iOS 版本** 我们将会在未来发布对Android操作系统的支持

**添加SDK时, 报错 SDK不支持该硬件。SDK not supported by this device** 一般是因为使用的SDK跟硬件不匹配, 如 GPU的SDK, 硬件没有GPU卡。对于Jetson, 也可能是Jetpack版本不支持, 可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表 (cpp文件中的文件名来查看) 来匹配。

🔗 智能边缘控制台-多节点版

🔗 EasyEdge 智能边缘控制台——多节点版

## 整体介绍

智能边缘控制台 - 多节点版 (EasyEdge Intelligent EdgeConsole Center 以下简称IECC), 是EasyEdge推出的边缘资源管理、服务应用与管理一站式本地化方案。

通过IECC, 用户可以方便地在中心节点管理子节点:

- 边缘硬件资源的管理与监控
- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活, 服务管理

- 视频流解析，接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关视频流推理事件

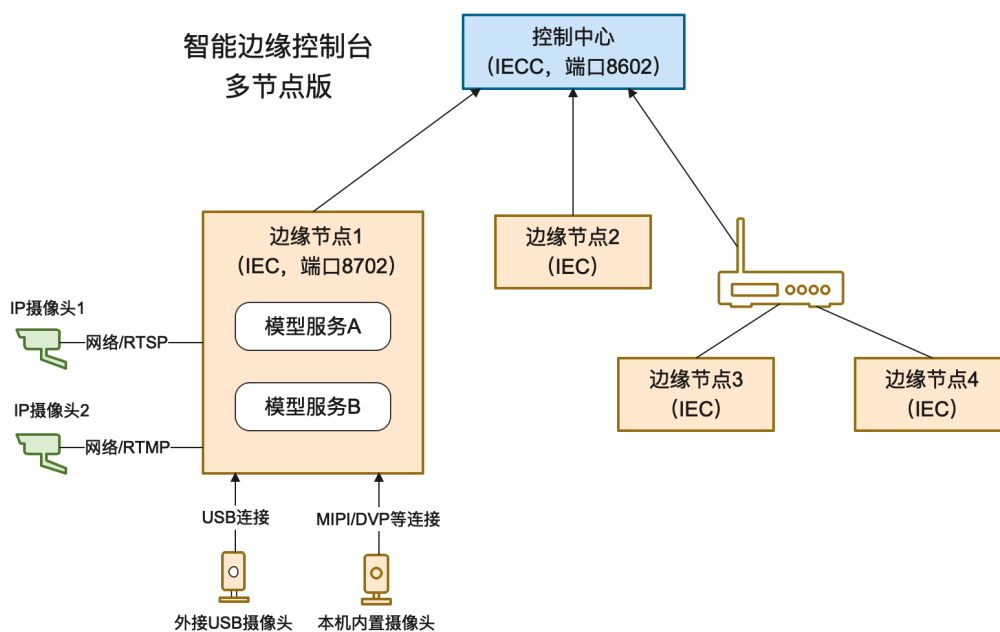
支持的系统+CPU架构包括：

- Windows x86\_64 (Windows 7 ~ Windows 10, 暂不支持Windows 11)
- Linux x86\_64 / arm32 / arm64

支持各类常见的AI加速芯片，包括：

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列
- 其他EasyDL/EasyEdge/BML支持的AI芯片

连接说明 以下为 中心节点（控制中心）,边缘节点/子节点,摄像头的连接示意：



其中：

- 控制中心需要有固定IP，而边缘节点可以处于多级子网之下，只需IEC能够主动访问到控制中心节点即可
- 模型服务均运行于各边缘节点之上
- 摄像头均与边缘节点相连

#### Release Note

版本号	发布时间	更新说明
2.2.0	2022-10-27	边缘节点新增Android支持；新增onvif/gb28181支持；优化端云通信通道安全
2.0.0	2022-03-25	多节点版上线！
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	智能边缘控制台 - 单节点版 IEC 第一版！

安装 从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#)：intel、AMD的64位x86\_64 CPU
- [Linux amd64](#)：intel、AMD的64位x86\_64 CPU

- [Linux arm](#) : 树莓派等32位的ARM CPU
- [Linux arm64](#) : RK3399、飞腾等64位的ARM CPU

或者从纯离线服务管理页可下载智能边缘控制台



以Linux为例，解压缩后目录结构如下所示：

```
./EasyEdge-IECC-v{版本号}/
|-- easyedge-iecc
|-- easyedge-iecc-setup.sh
|-- etc/
|-- etc/easyedge-iecc.yml
|-- readme.txt
```

## Linux 系统

### 通过系统服务形式安装（推荐）

以管理员运行 `bash easyedge-iecc-setup.sh install` 即可。

```
0 EasyEdge-IEC-v2.0.0 > bash ./easyedge-iecc-setup.sh install
[setup]: sudo could not be found
[setup]: Start to install IECC...
[setup]: + bash -c "cp easyedge-iecc-linux-amd64 /usr/sbin/easyedge-iecc"
[setup]: + bash -c "chmod +x /usr/sbin/easyedge-iecc"
[setup]: + bash -c "cp easyedge-iecc-* /var/lib/easyedge-iecc/fs/tmp"
[setup]: + bash -c "cp etc/easyedge-iecc.service.yml /etc/easyedge-iecc/easyedge-iecc.yml"
[setup]: + bash -c "cp etc/easyedge-iecc.service-conf.init.d /etc/init.d/easyedge-iecc"
[setup]: + bash -c "chmod +x /etc/init.d/easyedge-iecc"
[setup]: Install IECC success!
[setup]: + bash -c "service easyedge-iecc start"
Starting easyedge-iecc: success
[setup]: Start to check IECC status...
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: IECC status: OK!
[easyedge-iecc]: default configure file: /etc/easyedge-iecc/easyedge-iecc.yml
[easyedge-iecc]: default log file: /var/log/easyedge-iecc/easyedge-iecc.log
[easyedge-iecc]: service usage: service easyedge-iecc { start | stop }
[setup]: Done!
```

出现success字样，表示安装成功。

- 日志：`/var/log/easyedge-iecc/easyedge-iecc.log`
- 系统配置：`/etc/easyedge-iecc/easyedge-iecc.yml`
- 服务启动/停止：`service easyedge-iecc { start | stop }` (不同操作系统内可能不同，具体命令参考安装日志)
- 配置服务自启动：可根据不同操作系统参考[这里](#)进行对应配置

可通过 `bash easyedge-iecc-setup.sh uninstall` 来卸载，以及 `bash easyedge-iecc-setup.sh upgrade` 来升级为当前安装包的版本

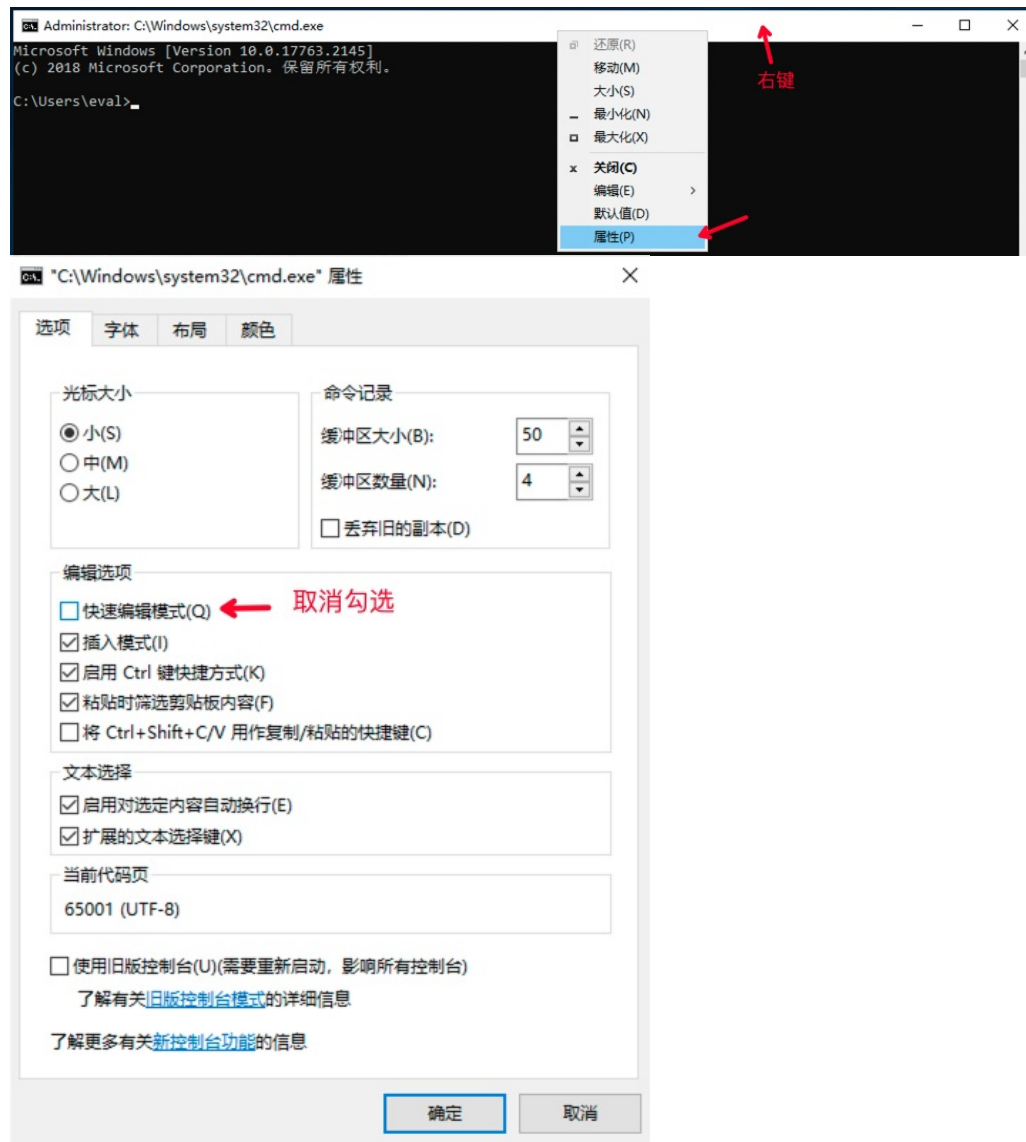
### 自定义安装（不推荐）

自定义安装仅限于 安装脚本无法识别您的操作系统的情况。

- 拷贝 `./EasyEdge-IEC-v2.0.0/` 整个目录至自定义文件夹，如 `/opt/EasyEdge-IEC`
- 进入到 `/opt/EasyEdge-IEC`
- 通过 `nohup` 等方法运行 `./easyedge-iec-linux-{您的系统架构} --com.role=hub amd64: intel、AMD的64位x86_84 CPU arm : 树莓派等32位的 ARM CPU * arm64 : RK3399、飞腾等64位的ARM CPU`
- 日志：`./log/easyedge-iecc.log`
- 系统配置：`./easyedge-iecc.yml`

Windows 系统 打开命令行（非powershell）运行 `easyedge-iecc-setup.bat install`。

注：如果遇到hang住的情况，可修改命令行配置



验证安装：启动之后，打开浏览器，访问 `http://{设备ip}:8602/easyedge` 即可：



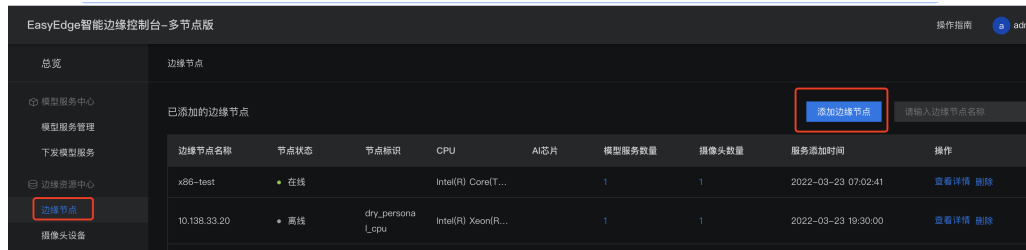
**更新服务：** 关闭服务，下载最新的安装包，重新执行安装流程即可。

注：1. 中心节点更新到新版之后，已连接的边缘节点会自动跟随中心节点，自我升级到同样的版本。

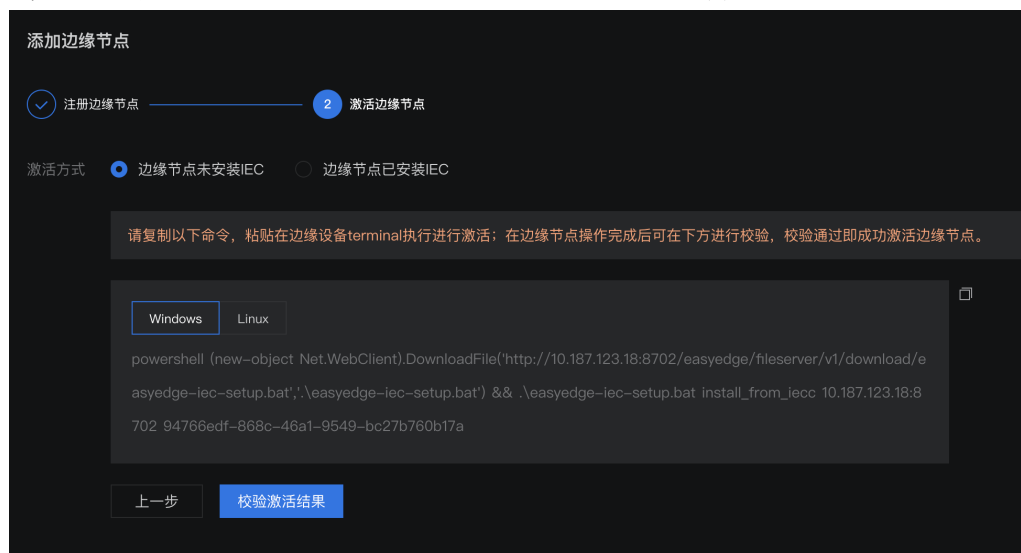
2. 报错: Text file busy. 一般是因为服务没有停止。

### 使用流程 Step 1 注册并激活边缘节点

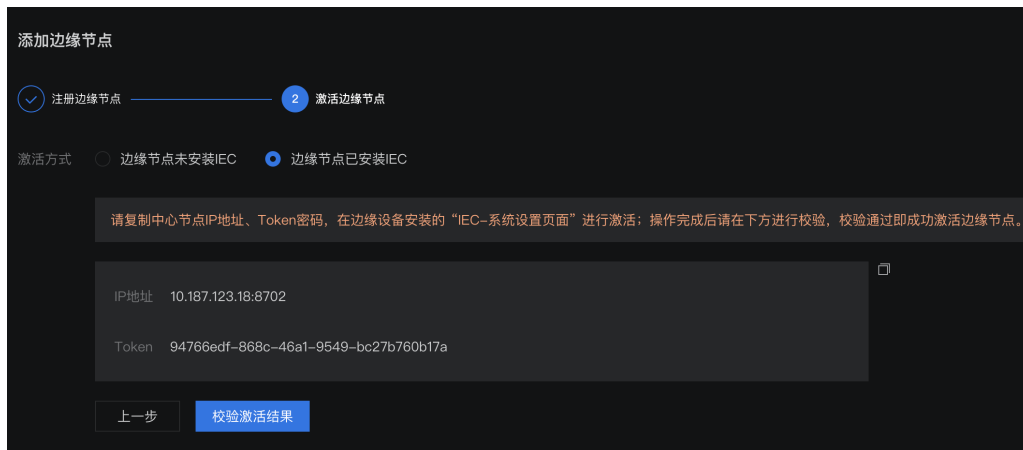
- 在IECC导航栏中点击边缘节点，点击页面中的添加边缘节点按钮



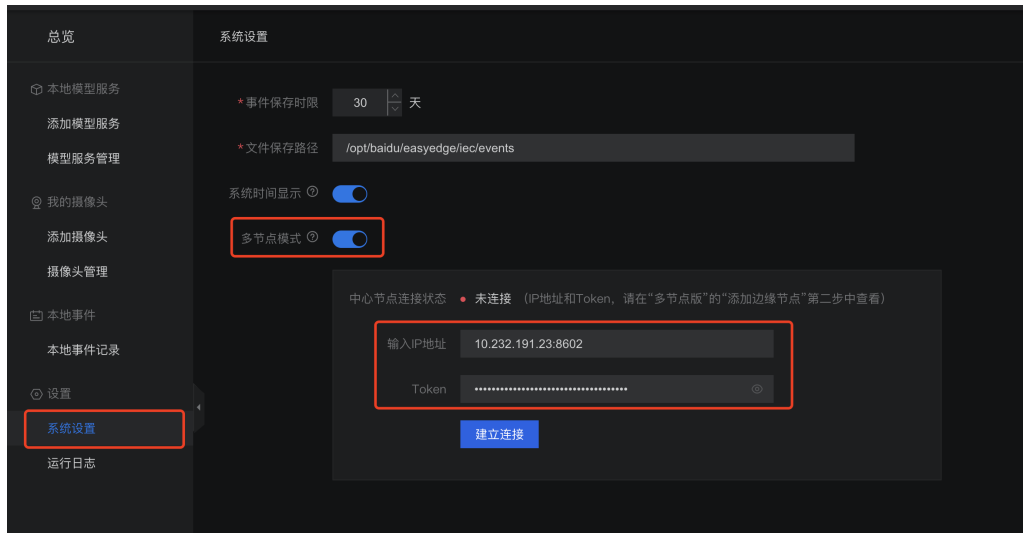
- 注册边缘节点，填写基本信息
- 激活边缘节点，根据边缘节点上是否安装智能边缘控制台-单节点版（IEC）分两种激活方式
  - 边缘节点未安装IEC：复制提供的命令，在边缘节点的终端中输入执行（命令会自动在当前目录，下载单节点版IEC并注册到控制中心）。终端命令执行完成后，在下方校验激活结果，如结果通过即可完成边缘节点的激活



- 边缘节点已安装IEC：记录页面中提供的IP地址和Token



- 在边缘节点的IEC-系统设置中，打开多节点模式开关，将刚才记录的IP地址和Token填入其中，建立连接

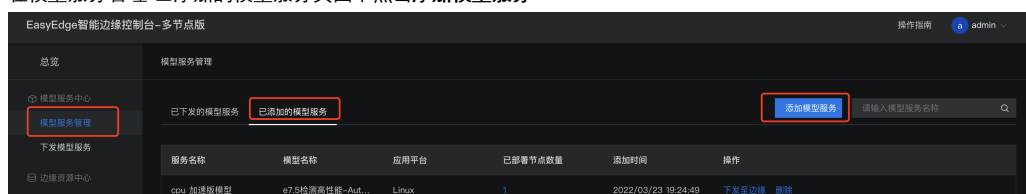


- 成功激活后可在边缘节点页面中看到一行状态为在线的记录



## Step 2 上传并下发模型服务

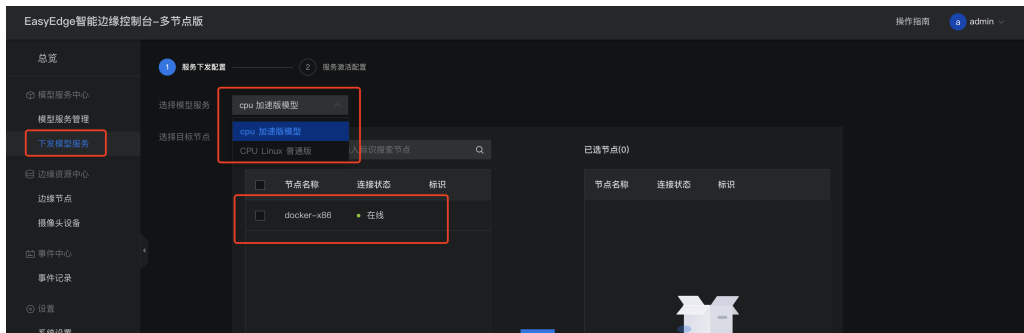
- 在模型服务管理-已添加的模型服务页面中点击添加模型服务



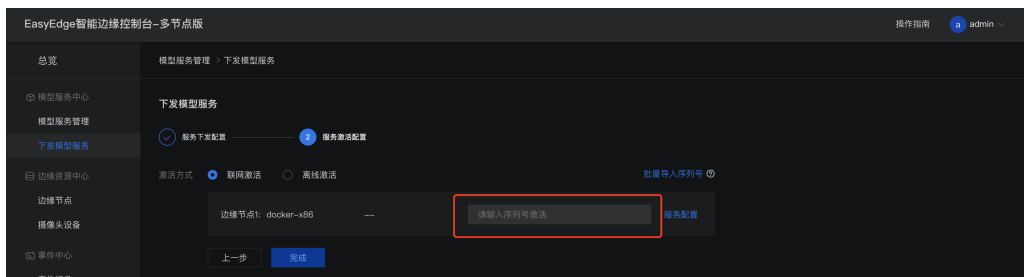
- 上传来自于EasyDL/BML的SDK，目前仅支持Windows/Linux的SDK



- 添加成功后可在已添加的模型服务页面查看添加的模型服务SDK
- 在模型服务SDK上传成功以及边缘节点也添加激活过后，即可将模型服务下发至边缘。点击导航栏-下发模型服务，选择已添加的模型服务，选择下发的目标节点（支持多节点批量下发）进行模型服务下发

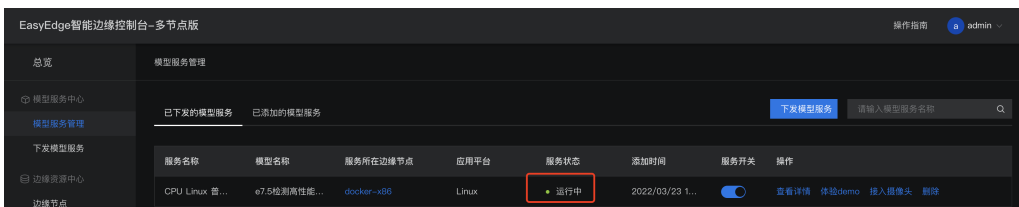


- 确定下发配置后，填入模型服务在边缘节点联网激活运行的序列号（支持批量导入）即可完成模型服务下发，序列号可在[智能云控制台](#)获取。离线激活的过程可参考IECC中的具体指引



- 完成上述流程后即可在模型服务管理-已下发的模型服务列表中查看记录，并进行下一步应用功能体验

注：完成此步骤后即可在边缘节点进行二次集成已下发的模型服务，具体的集成方式可在文档-某图像任务类型-模型发布中查找对应的SDK开发文档进行集成开发



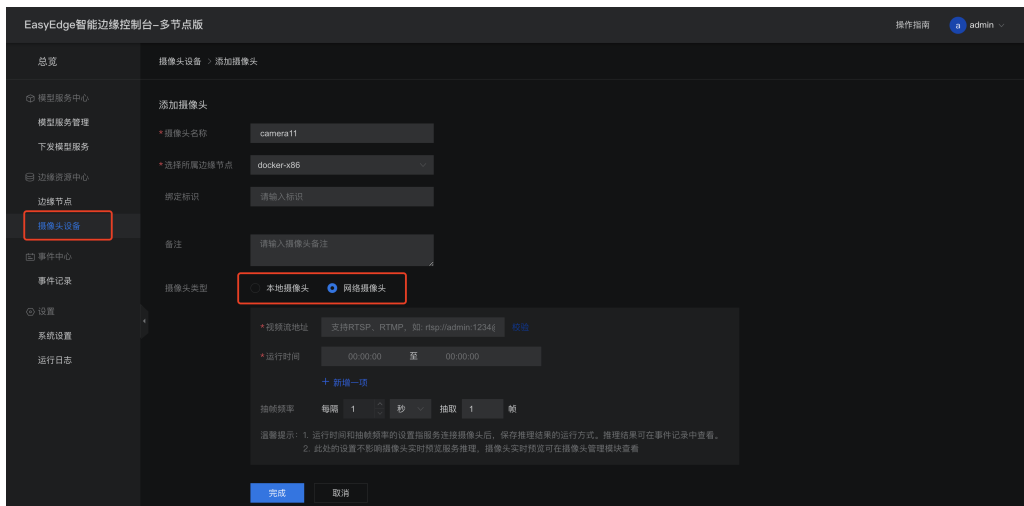
下发时可以通过高级配置设置服务运行的host和port。若不设置，默认host为0.0.0.0，port为系统随机分配的可用端口

### Step 3 配置摄像头

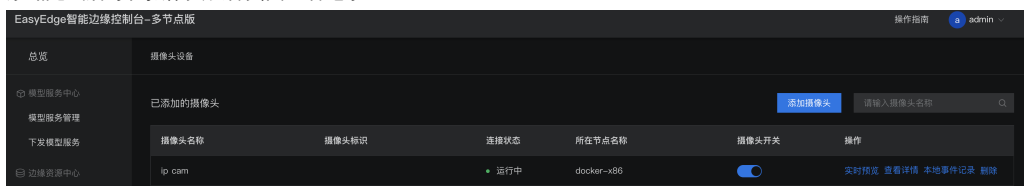
Step 3 - 5 描述的是如何使用IECC可视化进行视频流式推理与应用，对此有需求的用户建议详细查看后续步骤内容。如仅需对下发的模型服务进行二次集成的用户无需进行后续操作，参考SDK对应的开发文档进行集成即可



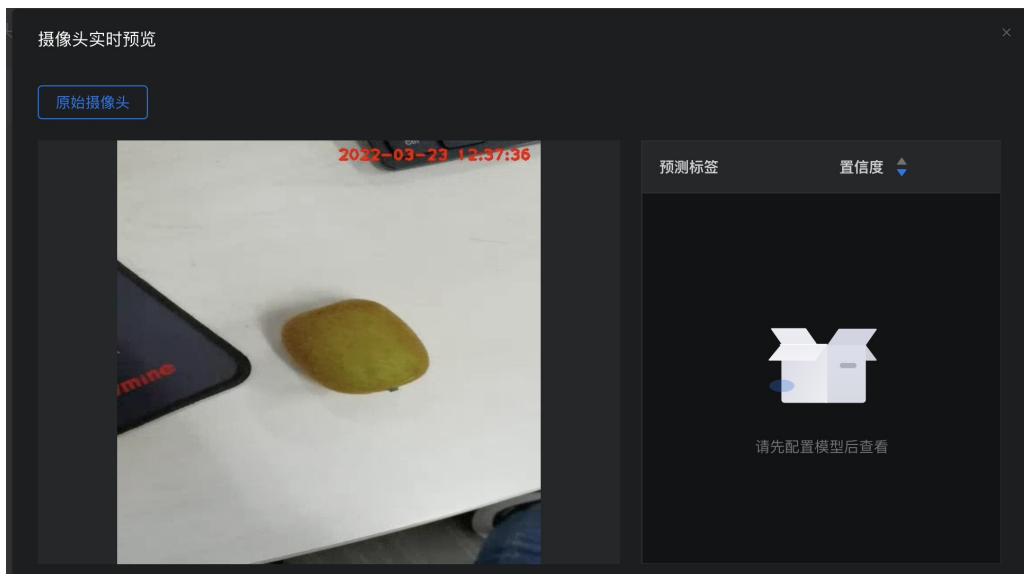
- 首先需要确定边缘节点已经接入物理摄像头，可通过USB接口接入，也可通过RTSP/RTMP流式协议接入。在摄像头设备页面点击添加摄像头按钮，填写对应的信息添加摄像头。支持设置摄像头的运行时间以及摄像头的抽帧频率



- 添加完成后可在摄像头设备页面查看记录

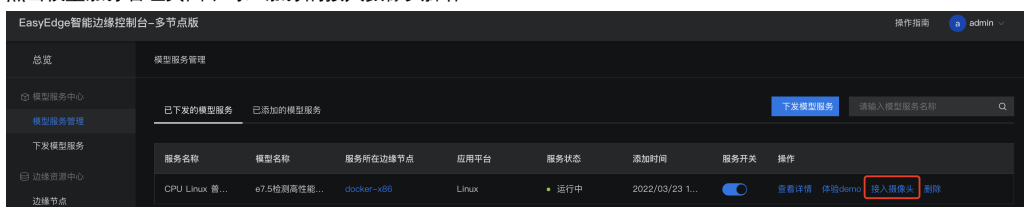


- 点击预览可查看摄像头预览画面

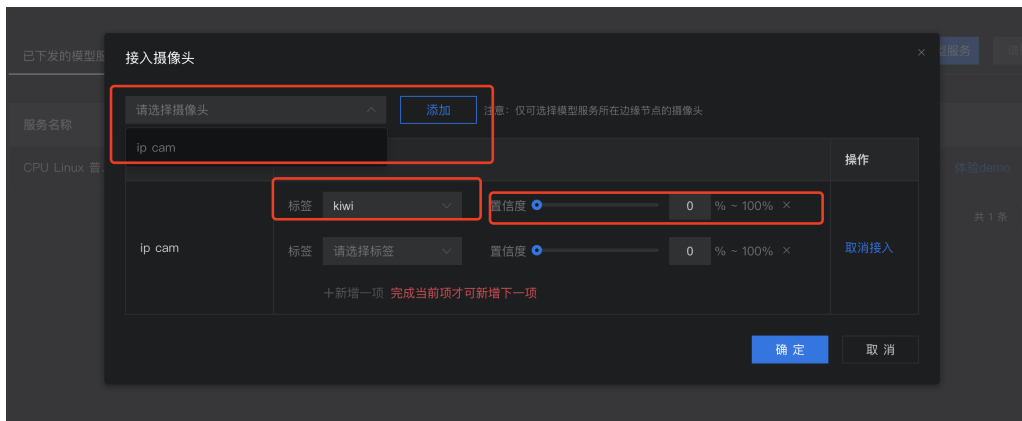


#### Step 4 模型服务接入视频流预测

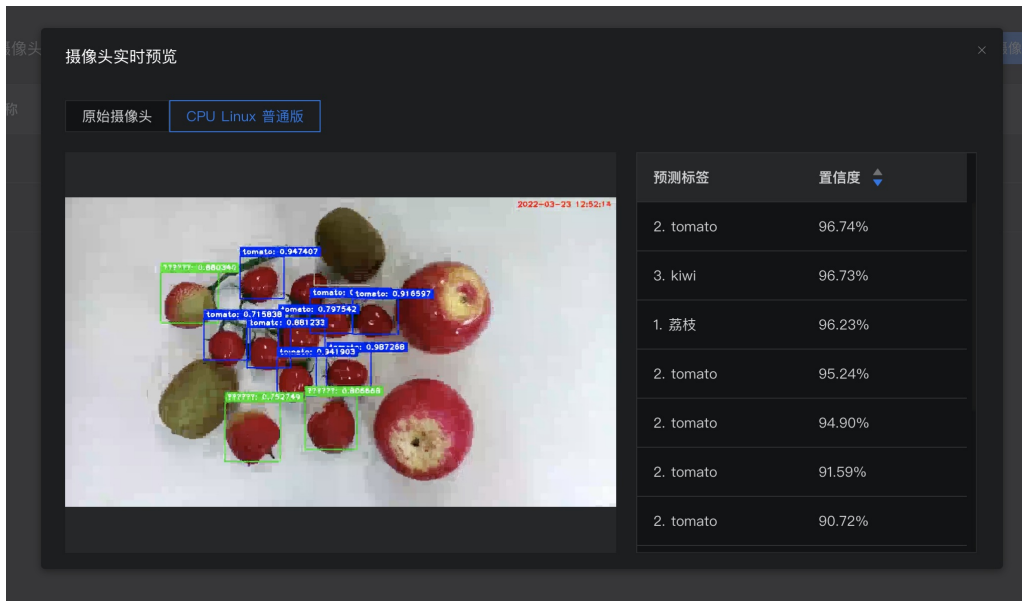
- 模型服务可接入摄像头直接进行预测，并可同时设置告警规则，出发告警条件的结果将会以事件的形式保存至IECC中。点击模型服务管理页面中对应服务的接入摄像头操作



- 将已添加至IECC的摄像头与模型服务关联，并在下方设置对应的事件告警条件。告警规则通过标签阈值的方式来建立，例如设置“猕猴桃”标签阈值80%-100%，则大于80%置信度的“猕猴桃”识别结果将会保存至事件记录中

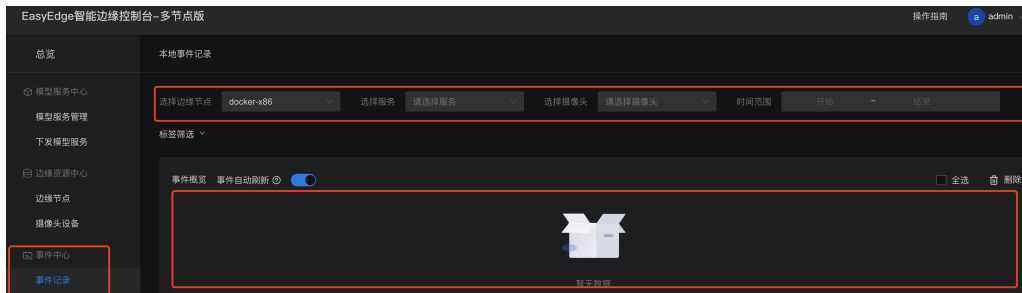


- 也可在摄像头设备页面-实时预览中查看实时的模型服务预测结果



### Step 5 视频事件告警

- 可在事件中心-事件记录中查看满足时间告警条件的图片记录



高级配置说明 在系统设置 - 高级，可以修改控制中心的高级系统配置

```

##### IECC系统配置
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
role: hub
# 硬件利用率刷新时间间隔：过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# IECC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: default
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge)
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
ToFile: yes
loggingFile: /var/log/easyedge-iecc/easyedge-iecc.log
# 0:info; -1:debug; -2:verbose
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8602
listenHost: 0.0.0.0

commu:
mqServer:
  host: 0.0.0.0
  port: 8632
  HTTPPort: 8620
  maxPayload: 8388608
  pingIntervalSecond: 30
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2
nodeRefreshIntervalSecond: 30

##### ----- 以下高级配置一般无需修改 -----
##### !!!注意!!! 请确保理解配置项含义后再做修改
##### 数据库相关配置
db:
  sqliteDbFile: /var/lib/easyedge-iecc/easyedge-iecc.db
  hubDbFile: /var/lib/easyedge-iecc/easyedge-iecc.hub.db
  eventDbFile: /var/lib/easyedge-iecc/easyedge-event.db
  fileServerDbFile: /var/lib/easyedge-iecc/easyedge-fileserver.hub.db
  nodeMonitorDbFile: /var/lib/easyedge-iecc/easyedge-nodemonitor.hub.db

##### 推流相关配置
mediaserver:
  flvPort: 8613
  rtmpPort: 8614

##### 文件服务器相关配置
fileserver:
  root: /var/lib/easyedge-iecc/fs

```

## FAQ

启动服务后，进程中出现两个 `easyedge-iec` 进程 这是正常现象，IEC通过守护进程的方式来完成更新等操作。

启动服务时，显示端口被占用 `port already been used` 通过修改 `easyedge-iecc.yml` 文件的配置后，再重新启动服务。

安装服务时，报错 `permission denied` 请以管理员身份运行安装程序。

添加SDK时,报错 SDK不支持该硬件。SDK not supported by this device 一般是因为使用的SDK跟硬件不匹配,如 GPU的SDK,硬件没有GPU卡。对于Jetson,也可能是Jetpack版本不支持,可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表 (cpp文件中的文件名来查看) 来匹配。

## 常见问题

### 数据相关问题

需要上传多少张图片才能训练出效果较好的模型？

- 每种要识别的物体在所有图片中出现的数量需要大于50。如果某些要区分的物体具有相似性,需要增加更多图片。

上传图片的总量有限制吗？

- 每个账号下所有数据集的图片总数不能超过10万张。

### 训练相关问题

数据处理失败或者状态异常怎么办？

- 如是是图像分类模型上传处理失败,请先检查已上传的分类命名是否正确,是否存在中文命名、或者增加了空格;然后检查下数据图片量是否超过上限(10万张);再检查图片中是否有损坏。如果自查没有发现问题,请在百度智能云控制台内[提交工单](#)反馈

模型训练失败怎么办？

- 如果遇到模型训练失败的情况,请在百度智能云控制台内[提交工单](#)反馈

已经上线的模型还可以继续优化吗？

- 已经上线的模型依然可以持续优化,操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练,然后在模型列表中更新线上服务,完成模型的优化

点击我的模型列表——找到新训练好的模型版本——点击申请发布

应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V2	训练完成	未申请	未发布	top1准确率87.61% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">校验</a> <a href="#">训练</a>
离线SDK	V1	训练完成	未申请	未发布	top1准确率85.84% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">训练</a>

每页显示 12 < 1 >

在出来的弹窗中点击确定



### 模型效果相关问题

图像分割模型如何正确标注？

- 所有图片中出现的目标物体都需要被标出(标注可以重叠)
- 标注应包含整个物体,且尽可能不要包含多余的背景
- 如果图片中存在很多相同标签的目标物体,可以使用右侧的锁定按钮。锁定标签后,只需要在左侧标注目标物体即可,不用再重复选择标签

如何通过「完整评估结果」里的错误示例优化模型？

- 错误示例中，左侧是正确的结果，右侧是模型的识别结果
- 观察模型识别有误的图片有哪些共同点，并有针对性地补充训练数据。比如：当图片比较亮的时候模型都能识别正确，但比较暗的时候模型就识别错了。这时就需要补充比较暗的图片作为训练数据

#### 我的数据有限，如何优化效果？

- 先申请发布模型，并备注说明希望通过[云服务数据管理](#)功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

#### 实际调用服务时模型效果变差？

- 训练图片和实际场景要识别的图片拍摄环境应一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片
- 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 如果使用的是云服务，可以开通[云服务数据管理](#)功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

如果训练数据已经达到以上要求，且单个分类/标签的图片量超过200张以上，效果仍然不佳，请在百度智能云控制台内[提交工单](#)反馈

### 🔗 智能标注相关问题

智能标注功能目前已对图像分割模型开放，[了解功能详情](#)

#### “一键标注”和“立即训练”要如何选择？

- 当系统推荐“立即训练”，且系统预标注确实已非常精准时，可以不用标注剩余数据，直接开始模型训练。此时，仅用当前已标注图片训练的模型，与标注所有数据后训练的模型相比，效果几乎等同
- 如果系统预标注还有些不精准，可以启动一键标注，人工确认系统的标注后，再开始训练

#### 选择了“立即训练”之后是否还可以“一键标注”？

- 选择“立即训练”之后，系统默认为您结束此次智能标注
- 再次启动智能标注后，您可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

#### 智能标注结束后，又往数据集上传了新图片，是否可以直接“一键标注”新图片？

- 如果您创建了新的标签、或新上传的图片场景和之前的图片场景差异较大，建议不要使用一键标注，而是从头开始智能标注（即再次筛选关键图片）
- 如果不是以上情况，再次启动智能标注后，可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

#### 智能标注中可以增删标签吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签
- 如果确实需要增删标签，可以先结束智能标注

#### 智能标注中可以增删图片吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片
- 如果确实需要增删图片，可以先结束智能标注

#### 智能标注中可以修改已标注图片的标注吗？

- 可以。但为了保证智能标注的效果，建议不要大量改动
- 如果确实需要修改大量标注，建议先结束智能标注

为什么我已经人工标注了很多图片，但系统预标注依然不准？

- 系统预标注的结果会受以下因素影响：
  - 智能标注期间，对“已标注”图片的标签进行大量改动
  - 曾结束智能标注，并对标签、图片进行增删
- 如果您没有进行以上操作，系统标注结果依然不理想，请在百度智能云控制台内[提交工单](#)反馈

#### 多个数据集是否可以同时启动智能标注？

- 目前每个账号同一时间仅支持对一个数据集启动智能标注

#### 共享中的数据集是否可以启动智能标注？

- 暂不支持。智能标注中的数据集也暂不支持共享，如有疑问，请在百度智能云控制台内[提交工单](#)反馈

#### 智能标注失败了怎么办？

- 可以先尝试稍后重新启动
- 若再次遇到问题，请在百度智能云控制台内[提交工单](#)反馈

### 🔗 模型上线相关问题

#### 希望加急上线怎么处理？

- 请在百度智能云控制台内[提交工单](#)反馈

#### 每个账号可以上线几个模型？是否可以删除已上线的模型？

- 每个账号最多申请发布十个模型，已上线模型无法删除

#### 申请发布模型审核不通过都是什么原因？

- 可能原因有，1、经过电话沟通当前模型存在一些问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝。2、电话未接通且模型效果较差，会直接拒绝。如果需要申诉，请在百度智能云控制台内[提交工单](#)反馈

## EasyDL 文本使用说明

### EasyDL文本介绍

#### 🔗 概述

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

EasyDL平台的自然语言处理模型定制能力，基于文心-NLP大模型领先的语义理解技术，为企业/开发者提供一整套NLP定制与应用能力。

当前EasyDL平台提供了8种模型定制能力：

- 文本分类-单标签：定制分类标签实现文本内容的自动分类，每个文本仅属于一种标签类型
- 文本分类-多标签：定制分类标签实现文本内容的自动分类，每个文本可同时属于多个分类标签
- 情感倾向分析：定制情感倾向分析模型，可实现文本按情感的正向（positive）和负向（negative）做自动分类
- 短文本相似度：定制短文本相似度模型，是基于深度学习技术，可实现对两个文本进行相似度的比较计算
- 文本实体抽取：定制文本实体抽取模型，实现对文本进行内容抽取，并识别为自定义的实体类别
- 文本实体关系抽取：定制实体关系抽取模型，是指从文本中抽取预定义的实体类型及实体间的关系类型，得到包含语义信息的实体关系三元组，每个实体关系三元组由两个实体及其关系构成
- 评论观点抽取：定制评论观点抽取模型，实现从文本中抽取评价片段、评价维度、评价观点，并判断评价情感倾向
- 大模型创作：定制文本创作模型，基于ERNIE 3.0大模型实现对输入文本内容进行创作和续写

#### 🔗 产品优势

#### 🔗 可视化操作

无需机器学习专业知识，通过模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型。

## 🔗 操作步骤

### \*\*Step 1 创建模型\*\*

确定模型名称，记录希望模型实现的功能。

### \*\*Step 2 上传并标注数据\*\*

不同类型的任务对应的数据格式不一致，您可以上传未标注数据并使用平台提供的标注工具进行标注。或直接上传各任务的标注数据。

### \*\*Step 3 训练模型并校验效果\*\*

选择部署方式与算法，用上传的数据一键训练模型。

模型训练完成后，可在线校验模型效果。

### \*\*Step 4 发布模型\*\*

根据训练时选择的部署方式，将模型以云端API、设备端私有API等多种方式发布使用

更详细的操作指导，请参考各类模型的技术文档

## 🔗 高精度效果

EasyDL文本任务内置[文心大模型](#)，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

[文心大模型](#)是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 🔗 丰富的部署方案

训练完成后，可将模型部署在公有云服务器、私有服务器。

部署方式	支持的硬件	支持的系统	技术文档
公有云API	无需关注硬件，发布到公有云	不限制	<a href="#">文本分类-单标签</a> 、 <a href="#">文本分类-多标签</a> 、 <a href="#">情感倾向分析</a> 、 <a href="#">短文本相似度</a> 、 <a href="#">文本实体抽取</a> 、 <a href="#">文本实体关系抽取</a> 、 <a href="#">评论观点抽取</a> 、 <a href="#">大模型创作</a>
EasyEdge本地-私有服务器部署[私有API]	x86-64 CPU	Linux	<a href="#">文本分类-单标签</a> 、 <a href="#">文本分类-多标签</a> 、 <a href="#">情感倾向分析</a> 、 <a href="#">短文本相似度</a> 、 <a href="#">文本实体抽取</a> 、 <a href="#">文本实体关系抽取</a> 、 <a href="#">评论观点抽取</a>
EasyEdge本地-私有服务器部署[私有API]	Nvidia GPU	Linux	<a href="#">文本分类-单标签</a> 、 <a href="#">文本分类-多标签</a> 、 <a href="#">情感倾向分析</a> 、 <a href="#">短文本相似度</a> 、 <a href="#">文本实体抽取</a> 、 <a href="#">文本实体关系抽取</a> 、 <a href="#">评论观点抽取</a>

## 🔗 公有云API

已全面支持[文本分类（单标签、多标签）](#)、[情感倾向分析](#)、[短文本相似度](#)、[文本实体抽取](#)、[文本实体关系抽取](#)、[评论观点抽取](#)、[大模型创作](#)。

训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统整合。

具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求。

## 🔗 私有服务器部署

已全面支持[文本分类（单标签、多标签）](#)、[情感倾向分析](#)、[短文本相似度](#)、[文本实体抽取](#)、[文本实体关系抽取](#)、[评论观点抽取](#)。

EasyDL文本任务使用EasyEdge本地部署服务，支持[私有服务器部署API的本地化部署](#)。私有API的特点如下：

- 将训练完成的模型部署在私有CPU/GPU服务器上，可在内网/无网环境下使用模型，确保数据隐私
- 将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口
- 可纯离线完成部署，服务调用便捷

私有API支持的Linux发行版本如下：

- Ubuntu: 14、16、18



- Centos : 7.0及以上
- redhat : 7.2以上
- suse 12

## 文本分类-单标签

### 整体介绍

#### 简介

Hi, 您好, 欢迎使用百度EasyDL定制化训练和服务平台。

定制文本分类的模型, 是基于自建分类体系的机器学习方法, 可实现文本按内容类型做自动分类。平台目前提供的文本分类模型包括: 文本分类(单标签)和文本分类(多标签)两种模型类型, 请您根据自己的业务场景来选择合适的模型。本文介绍的是关于**文本分类(单标签)**的模型介绍。

文本分类(单标签)场景: 如您对网络文章进行舆情分析, 判断舆情是正向评价还是负向评价, 即每条文本**仅有一个**分类标准, 此问题属于单标签的文本分类场景;

文本分类(多标签)场景: 如您对网络文章进行板块划分, 即每条文本有**两个及以上**分类标准, 文章可能属于娱乐、国际、生活等多个标签, 则可使用多标签的文本分类模型;

更多详情访问: [EasyDL自然语言处理方向](#)

#### 应用场景

- 1、投诉信息分类: 训练客服投诉信息的自动分类, 将每个用户投诉的内容进行分类管理, 节省大量客服人力
- 2、媒体文章分类: 训练网络媒体文章的自动分类, 进而实现各类文章的自动分类
- 3、文本审核: 定制训练文本审核的模型, 如训练文本中是否含有违规/偏激性质的描述
- 4、其他: 尽情脑洞大开, 训练你希望实现的文本分类(单标签)模型

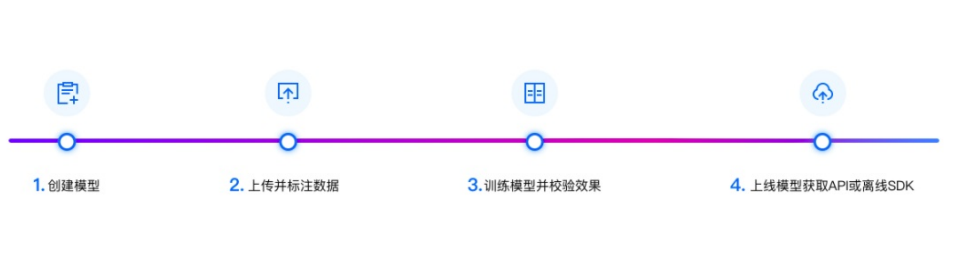
#### 技术特色

文本分类模型内置**文心大模型**, 将大数据预训练与多源丰富知识相结合, 通过持续学习技术, 不断吸收海量文本数据中词汇、结构、语义等方面的新知识, 实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型, 是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型, 也包含了面向重点领域和重点任务的大模型, 还提供丰富的工具与平台, 支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色, 文心能够同时从大规模知识和海量多元数据中持续学习, 如同站在巨人的肩膀上, 训练效率和理解准确率都得到大幅提升, 并具备了更好的可解释性。

#### 使用流程

训练模型的基本流程如下图所示, 全程可视化简易操作, 在数据已经准备好的情况下, 最快15分钟即可获得定制模型。



## 数据准备

### 创建数据集并导入

#### 创建数据集

在训练模型之前, 需要创建数据集。需输入数据集名称、选择相应的标注模版、选择数据去重策略, 即可创建一个空数据集。





**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

如果待导入数据集是中文简体/繁体，请选择『短文本单标签』；如果待导入数据集是非中文的其他语言，请选择『多语种文本单标签』，[点击可查看](#)支持的全部语言种类。

**导入数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。

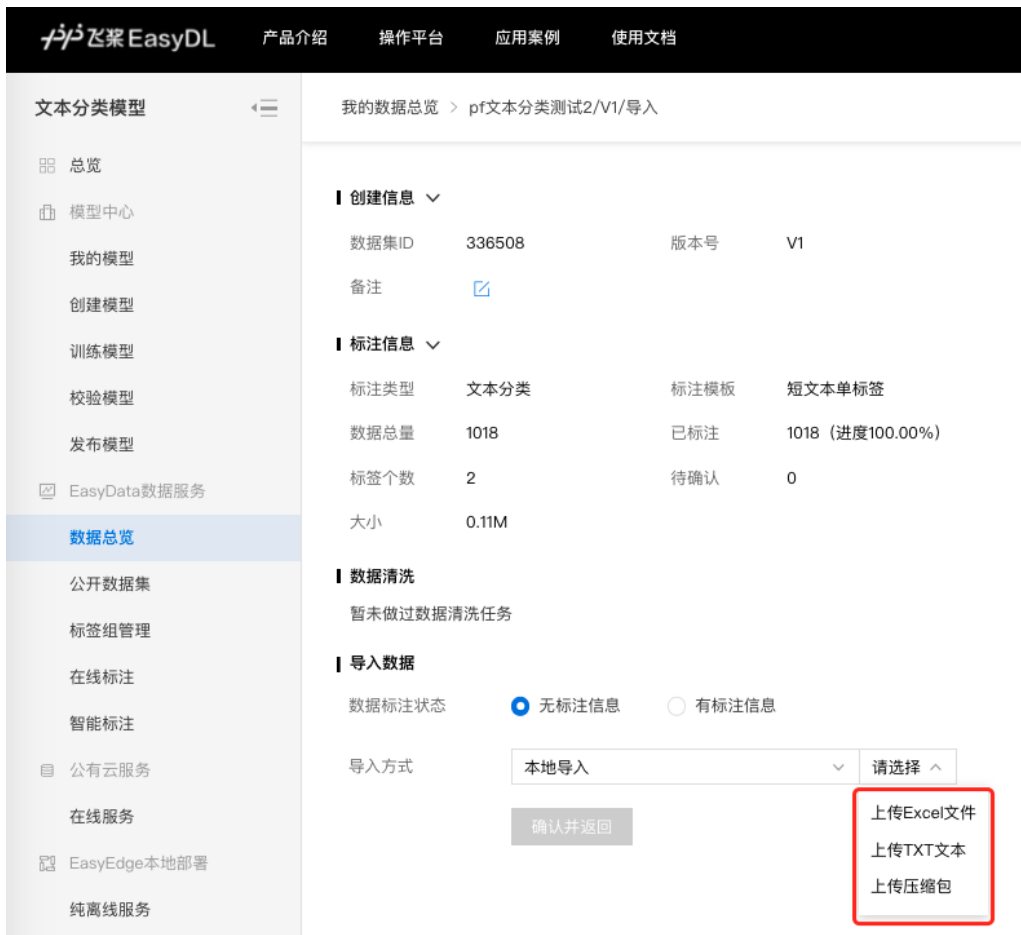


您可以使用4种方案上传文

本分类的数据，分别为：

- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

**本地导入** 您可以通过以下三种方式进行本地数据的导入：



- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传 **以压缩包方式上传**

如果您想上传的数据为压缩包，请根据您的数据是否已标注，按照以下格式要求完成数据上传。

#### 无标注数据

- 压缩包内包含上传的所有文本数据，每一个文本文件将作为一个样本上传，详见[示例压缩包](#)
- 压缩包格式为.zip格式，压缩包内文件类型支持txt，文件编码仅支持UTF-8

#### 有标注数据

- 压缩包格式为.zip格式，同时压缩包大小在5GB以内，文本编码仅支持UTF-8，每个文本文件最长不能超过4096个字符
- 标注文件中标签由数字、中英文、中/下划线组成，长度上限256字符。

有标注数据可以使用以下两种格式组织压缩包的内容：

- 以文件夹命名样本的标签：压缩包内按照文本类别数量分为多个文件夹，以文件夹的名称作为文本类别标签，文件夹下的所有txt文件作为样本，详细请见[示例压缩包](#)
- 用json文件标记分类：压缩包内仅支持单个文本文件（txt）及同名的json格式标注文件的上传，可传多组样本，详细请见[示例压缩包](#)

#### 以TXT文本文件上传

- 无标注数据文本文件内数据格式要求为“文本内容\n”（即每行一个未标注样本，使用回车换行），详见[数据样例](#)。有标注数据中文本文件内数据格式要求为“文本内容\t标签\n”（即每行一个标注样本，使用tab键将文本内容与标签分开，使用回车换行），详见[数据样例](#)。每一行表示一组数据，每组数据的字符数建议不超过4096个字符，超出将被截断；训练的字符数不超过512个字符，超出的字符可正常保存，但不参与训练。
- 文本文件类型支持txt，编码仅支持UTF-8，单次上传限制100个文本文件，最多可上传100万个文本文件

#### 以Excel文件上传

- Excel文件内数据格式要求为：每行是一个样本详见[数据样例](#)，如果您上传的为有标注数据，则每行的样本包含两列，第一列为数据文本内容，第二列为文本对应标签，详见[数据样例](#)；如果您上传的为无标注数据，则每行样本仅包含第一列数据文本内容，每个数据样本文本内容的字符数建议不超过4096个，超出将被截断。
- 文件类型支持xlsx格式，单次上传文件个数上限为100个
- 请确保您上传的样本在sheet1中，注意，首行作为表头将被系统忽略

**BOS目录导入** 需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

**分享链接导入** 需输入链接地址。分享链接导入的要求如下：

- 仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

**平台已有数据集**

- 导入无标注数据时，选择需要导入的数据集名称，可导入其不带标注的全部数据，或未标注的数据
- 导入已标注数据时，选择需要导入的数据集名称，可导入其某个或全部标签下的数据

**准备数据集的技巧** 文本分类任务中，可参考以下准备数据集的技巧：**设计分类**

首先想好分类如何设计，每个分类为你希望识别出的一种结果，如要识别新闻的内容类型，则可以以“科技”、“体育”、“农业”等分别作为一个分类标准；如果审核场景中通过文本判断是否出现广告，可以设计为两类设计为“正常”、“不正常”两类，或者“正常”、“异常原因一”、“异常原因二”、“异常原因三”等多类。

**注意：**目前单个模型的上限为1000类，暂不支持扩容

**数据量**

基于设计好的分类准备文本数据，每个分类建议至少需要准备50个文本文件以上，如果想要较好的效果，建议文件1000个起，如果某些分类的文本具有相似性，需要增加更多文本。

文本的基本格式要求：目前文本文件类型支持txt，文本文件大小限制长度最大4096，格式为UTF-8字符。一个模型的文本总量限制10万个文本文件。

**数据分布**

- 训练集文本需要和实际场景要识别的文本环境一致
- 考虑实际应用场景的种种可能性，每个分类的文本需要覆盖实际场景里面存在的可能性，训练集若能覆盖的场景越多，模型的泛化能力则越强

**可能的疑问**


- 如果训练文本数据无法全部覆盖实际场景要识别的文本，怎么办？

答：训练的模型算法会有一定的泛化能力，尽可能覆盖即可。

- 多语种模型支持全球94种语言：

南非语, 阿姆哈拉语, 阿拉伯语, 阿萨姆语, 阿塞拜疆语, 白俄罗斯语, 保加利亚语, 孟加拉语, 孟加拉语(拉丁化), 布列塔尼亚语, 波斯尼亚语, 加泰隆语, 捷克语, 威尔士语, 丹麦语, 德语, 希腊语, 英语, 世界语, 西班牙语, 爱沙尼亚语, 巴斯克语, 波斯语, 芬兰语, 法语, 弗里斯兰语, 爱尔兰语, 苏格兰盖尔语, 加利西亚语, 古吉拉特语, 希伯来语, 印地语, 印地语(拉丁化), 克罗地亚语, 匈牙利语, 亚美尼亚语, 印尼语, 冰岛语, 意大利语, 日语, 爪哇语, 格鲁吉亚语, 哈萨克语, 高棉语, 康纳达语, 韩语, 库尔德语, 柯尔克孜语, 拉丁语, 老挝语, 立陶宛语, 拉脱维亚语, 马拉加语, 马其顿语, 马拉雅拉姆语, 蒙古语, 马拉提语, 马来语, 缅甸语, 尼泊尔语, 荷兰语, 挪威语, 奥里亚语, 旁遮普语, 巴利语, 普什图语, 葡萄牙语, 罗马尼亚语, 俄语, 梵语, 信德语, 僧伽罗语, 斯洛伐克语, 斯洛文尼亚语, 索马里语, 阿尔巴尼亚语, 塞尔维亚语, 巽他语, 瑞典语, 斯瓦希里语, 泰米尔语, 泰米尔语(拉丁化), 泰卢固语, 泰卢固语(拉丁化), 泰语, 他加禄语, 土耳其语, 维吾尔语, 乌克兰语, 乌尔都语, 乌尔都语(拉丁化), 乌兹别克语, 越南语, 意第绪语。

如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单](#)反馈

 **在线标注**

**在线标注**

**\*\*Step 1 进入标注页面\*\*** 上传未标注的数据后，可以通过以下两个方式进入标注页面：

- 在「数据总览」页面，该数据集对应的操作列下，点击「标注」，即可进入标注页面

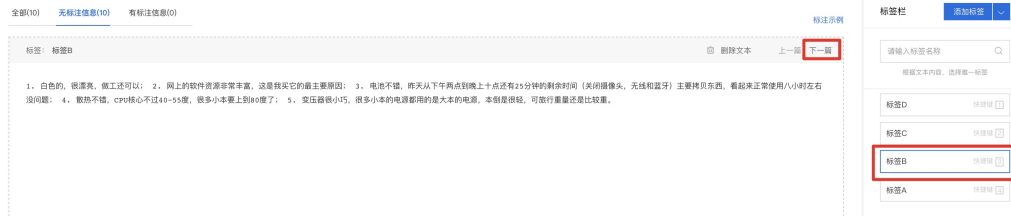
- 在「在线标注」页面，选择该数据集，即可进入标注页面

**\*\*Step 2 进行文本标注\*\*** 针对尚未进行标注的数据，通过 ([ 下一篇 ] 按钮) 方式进行标注：

- 在右边标签栏中添加标签
- 针对文本内容，选择其对应的标签
- 点击下一篇，此篇文本的内容即可进行自动保存，且您将开始对下一篇文本进行标注

针对已进行标注的数据，通过 ([ 下一篇 ] 按钮) 方式进行标注修改：

- 进入需修改标签的文本的标注页面，选择右边标签栏中的标签
- 点击下一篇，对此篇文本标签的修改即可进行自动保存



标注技巧: 每个标签的已标注文本需大于100 条，且“无标注数据”大于0条，即可启动**智能标注**功能，提高标注效率

**\*\*Step 3 查看标注信息\*\*** 通过 ([ 查看 ] 按钮) 方式查看已标注的文本信息：

- 在「数据总览」页面，该数据集对应的操作列下，点击「查看」，进入查看标注页面后，点击「有标注信息」
- 通过选择左侧标签中的不同标签名称，即可查看不同标签下的文本数据



🔗 数据去重

**重复样本的定义**

一个样本包括文本内容和标签。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。

例如：

文本内容	标签
今天北京的空气不错	weather
今天北京的空气不错	weather
今天北京的空气不错	local

上表三个样本均为重复样本，后两个样本虽然标签不一，但文本内容一致，也为重复样本。根据文本出现的顺序，最后一次的重复样本将代替之前的重复样本。

小Tips：“如何利用好重复样本”如果您的数据存在样本种类不均衡的现象，您可以通过将重复样本数量小的那一类，使其样本数量增加到与数据量大的那一类样本数量相近，以提高模型训练的效果，这种方法也称为“上采样”。

## 平台去重策略

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

- 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
- 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
- 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

## API上传

本文档主要说明当您线下已有大量的已经完成分类整理的文本数据，如何通过调用API完成文本数据的便捷上传和管理。

EasyDL数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

## 数据集创建API

### 接口描述

该接口可用于创建数据集。

### 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/create

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

### 查看数据集列表API

#### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/list

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态, 包括shared、smart和空值, 分别表示共享中、智能标注中、非特殊状态

### 查看分类 (标签) 列表API

#### 接口描述

该接口可用于查看分类 (标签)。返回分类 (标签) 的名称、包含数据量等信息。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法 : POST

请求URL : <https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下 :

参数	值
Content-Type	application/json

Body中放置请求参数, 参数详情如下 :

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型, 可包括: IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应: 图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号, 默认0
num	否	number	数量, 默认20, 最多100

若查看声音分类的全部分类, 在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：`POST`

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数



字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
append Label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类10000个汉字</b>
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称（由数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

##### 接口描述

该接口可用于删除数据集。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID

若删除声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 分类（标签）删除API

##### 接口描述

该接口可用于删除分类（标签）。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

若删除声音分类的子类，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用, 请再次请求, 如果持续出现此类错误, 请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在, 请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数, 请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法, 请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 🔗 创建数据集并导入

### 创建数据集

在训练模型之前, 需要创建数据集。需输入数据集名称、选择相应的标注模版、选择数据去重策略, 即可创建一个空数据集。

The screenshot shows the EasyDL web interface. The top navigation bar includes '产品介绍', '操作平台', '应用案例', and '使用文档'. The left sidebar contains a menu with '文本分类模型' selected, and sub-items like '总览', '模型中心', '我的模型', '创建模型', '训练模型', '校验模型', '发布模型', 'EasyData数据服务', '数据总览', and '公开数据集'. The main content area is titled '我的数据总览' and features a blue '创建数据集' button highlighted with a red box. Below the button, there is a table listing datasets:

版本	数据集ID	数据量	最近导入状态	标注类型
V1	336508	1018	● 已完成	文本分类

At the top of the main content area, there is a notification: 'EasyData智能数据服务平台已上线, 使用EasyData可享受包括多人标注、数据清洗、数据采集等完整数据服务 [立即前往](#)'.

**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

如果待导入数据集是中文简体/繁体，请选择『短文本单标签』；如果待导入数据集是非中文的其他语言，请选择『多语种文本单标签』，[点击可查看支持的全部语言种类](#)。

**导入数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。

飞桨 EasyDL 产品介绍 操作平台 应用案例 使用文档

我的数据总览 > pf文本分类测试2/V1/导入

**创建信息**

数据集ID	336508	版本号	V1
备注	<a href="#">🔗</a>		

**标注信息**

标注类型	文本分类	标注模板	短文本单标签
数据总量	1018	已标注	1018 (进度100.00%)
标签个数	2	待确认	0
大小	0.11M		

**数据清洗**

暂未做过数据清洗任务

**导入数据**

数据标注状态  无标注信息  有标注信息

导入方式

请选择

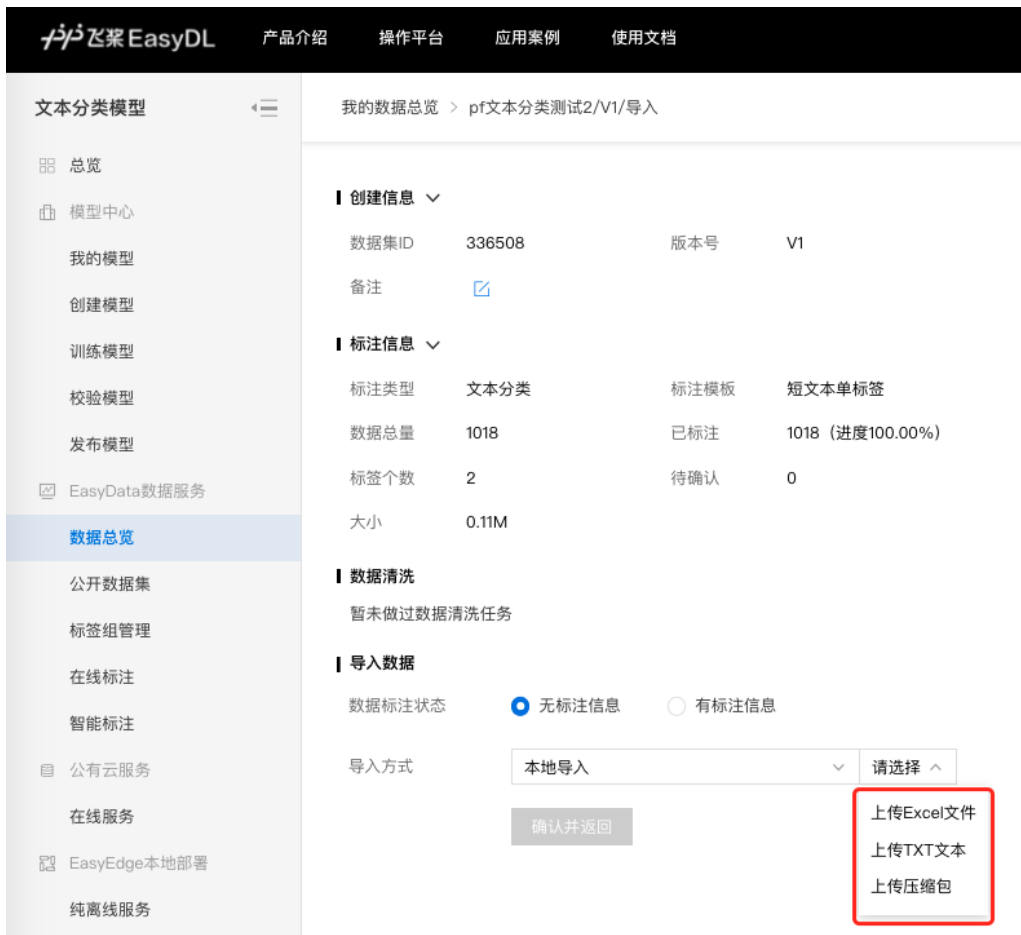
- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

您可以使用4种方案上传文

本分类的数据，分别为：

- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

**本地导入** 您可以通过以下三种方式进行本地数据的导入：



- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传 **以压缩包方式上传**

如果您想上传的数据为压缩包，请根据您的数据是否已标注，按照以下格式要求完成数据上传。

#### 无标注数据

- 压缩包内包含上传的所有文本数据，每一个文本文件将作为一个样本上传，详见[示例压缩包](#)
- 压缩包格式为.zip格式，压缩包内文件类型支持txt，文件编码仅支持UTF-8

#### 有标注数据

- 压缩包格式为.zip格式，同时压缩包大小在5GB以内，文本编码仅支持UTF-8，每个文本文件最长不能超过4096个字符
- 标注文件中标签由数字、中英文、中/下划线组成，长度上限256字符。

有标注数据可以使用以下两种格式组织压缩包的内容：

- 以文件夹命名样本的标签：压缩包内按照文本类别数量分为多个文件夹，以文件夹的名称作为文本类别标签，文件夹下的所有txt文件作为样本，详细请见[示例压缩包](#)
- 用json文件标记分类：压缩包内仅支持单个文本文件（txt）及同名的json格式标注文件的上传，可传多组样本，详细请见[示例压缩包](#)

#### 以TXT文本文件上传

- 无标注数据文本文件内数据格式要求为“文本内容\n”（即每行一个未标注样本，使用回车换行），详见[数据样例](#)。有标注数据中文本文件内数据格式要求为“文本内容\t标签\n”（即每行一个标注样本，使用tab键将文本内容与标签分开，使用回车换行），详见[数据样例](#)。每一行表示一组数据，每组数据的字符数建议不超过4096个字符，超出将被截断；训练的字符数不超过512个字符，超出的字符可正常保存，但不参与训练。
- 文本文件类型支持txt，编码仅支持UTF-8，单次上传限制100个文本文件，最多可上传100万个文本文件

#### 以Excel文件上传

- Excel文件内数据格式要求为：每行是一个样本详见[数据样例](#)，如果您上传的为有标注数据，则每行的样本包含两列，第一列为数据文本内容，第二列为文本对应标签，详见[数据样例](#)；如果您上传的为无标注数据，则每行样本仅包含第一列数据文本内容，每个数据样本文本内容的字符数建议不超过4096个，超出将被截断。
- 文件类型支持xlsx格式，单次上传文件个数上限为100个
- 请确保您上传的样本在sheet1中，注意，首行作为表头将被系统忽略

**BOS目录导入** 需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

**分享链接导入** 需输入链接地址。分享链接导入的要求如下：

- 仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

**平台已有数据集**

- 导入无标注数据时，选择需要导入的数据集名称，可导入其不带标注的全部数据，或未标注的数据
- 导入已标注数据时，选择需要导入的数据集名称，可导入其某个或全部标签下的数据

**准备数据集的技巧** 文本分类任务中，可参考以下准备数据集的技巧：**设计分类**

首先想好分类如何设计，每个分类为你希望识别出的一种结果，如要识别新闻的内容类型，则可以以“科技”、“体育”、“农业”等分别作为一个分类标准；如果审核场景中通过文本判断是否出现广告，可以设计为两类设计为“正常”、“不正常”两类，或者“正常”、“异常原因一”、“异常原因二”、“异常原因三”等多类。

**注意：**目前单个模型的上限为1000类，暂不支持扩容

**数据量**

基于设计好的分类准备文本数据，每个分类建议至少需要准备50个文本文件以上，如果想要较好的效果，建议文件1000个起，如果某些分类的文本具有相似性，需要增加更多文本。

文本的基本格式要求：目前文本文件类型支持txt，文本文件大小限制长度最大4096，格式为UTF-8字符。一个模型的文本总量限制10万个文本文件。

**数据分布**

- 训练集文本需要和实际场景要识别的文本环境一致
- 考虑实际应用场景的种种可能性，每个分类的文本需要覆盖实际场景里面存在的可能性，训练集若能覆盖的场景越多，模型的泛化能力则越强

**可能的疑问**


- 如果训练文本数据无法全部覆盖实际场景要识别的文本，怎么办？

答：训练的模型算法会有一定的泛化能力，尽可能覆盖即可。

- 多语种模型支持全球94种语言：

南非语, 阿姆哈拉语, 阿拉伯语, 阿萨姆语, 阿塞拜疆语, 白俄罗斯语, 保加利亚语, 孟加拉语, 孟加拉语(拉丁化), 布列塔尼语, 波斯尼亚语, 加泰隆语, 捷克语, 威尔士语, 丹麦语, 德语, 希腊语, 英语, 世界语, 西班牙语, 爱沙尼亚语, 巴斯克语, 波斯语, 芬兰语, 法语, 弗里斯兰语, 爱尔兰语, 苏格兰盖尔语, 加利西亚语, 古吉拉特语, 希伯来语, 印地语, 印地语(拉丁化), 克罗地亚语, 匈牙利语, 亚美尼亚语, 印尼语, 冰岛语, 意大利语, 日语, 爪哇语, 格鲁吉亚语, 哈萨克语, 高棉语, 康纳达语, 韩语, 库尔德语, 柯尔克孜语, 拉丁语, 老挝语, 立陶宛语, 拉脱维亚语, 马拉加语, 马其顿语, 马拉雅拉姆语, 蒙古语, 马拉提语, 马来语, 缅甸语, 尼泊尔语, 荷兰语, 挪威语, 奥里亚语, 旁遮普语, 巴利语, 普什图语, 葡萄牙语, 罗马尼亚语, 俄语, 梵语, 信德语, 僧伽罗语, 斯洛伐克语, 斯洛文尼亚语, 索马里语, 阿尔巴尼亚语, 塞尔维亚语, 巽他语, 瑞典语, 斯瓦希里语, 泰米尔语, 泰米尔语(拉丁化), 泰卢固语, 泰卢固语(拉丁化), 泰语, 他加禄语, 土耳其语, 维吾尔语, 乌克兰语, 乌尔都语, 乌尔都语(拉丁化), 乌兹别克语, 越南语, 意第绪语。

如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单](#)反馈

 **数据去重**

**重复样本的定义**

一个样本包括文本内容和标签。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。

例如：

文本内容	标签
今天北京的空气不错	weather
今天北京的空气不错	weather
今天北京的空气不错	local

上表三个样本均为重复样本，后两个样本虽然标签不一，但文本内容一致，也为重复样本。根据文本出现的顺序，最后一次重复样本将代替之前的重复样本。

小Tips：“如何利用好重复样本”如果您的数据存在样本种类不均衡的现象，您可以通过将重复样本数量小的那一类，使其样本数量增加到与数量大的那一类样本数量相近，以提高模型训练的效果，这种方法也称为“上采样”。

## 平台去重策略

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

- 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
- 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
- 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

## 模型训练

### 创建模型

#### 步骤 Step 1 创建模型

在【模型中心】或者【模型中心-我的模型】点击创建模型；数据集是中文数据集，任务场景请选择『短文分类任务』；数据集是非中文数据集，任务场景请选择『多语种文本分类任务』，模型共支持全球94种语言，[点击可查阅](#)

Step 2 填写基本信息 选择模型类型、提交模型名称、模型描述、联系方式即可创建模型。

Step 3 查看已创建的模型 模型创建成功后，可以在【我的模型】中看到刚刚创建的模型，操作示例见下图。

1. 创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
2. 目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。
3. 如果您是 enterprise 用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

- 支持语言清单：

南非语, 阿姆哈拉语, 阿拉伯语, 阿萨姆语, 阿塞拜疆语, 白俄罗斯语, 保加利亚语, 孟加拉语, 孟加拉语(拉丁化), 布列塔尼语, 波斯尼亚语, 加泰隆语,



捷克语, 威尔士语, 丹麦语, 德语, 希腊语, 英语, 世界语, 西班牙语, 爱沙尼亚语, 巴斯克语, 波斯语, 芬兰语, 法语, 弗里斯兰语, 爱尔兰语, 苏格兰盖尔语, 加利西亚语, 古吉拉特语, 希伯来语, 印地语, 印地语(拉丁化), 克罗地亚语, 匈牙利语, 亚美尼亚语, 印尼语, 冰岛语, 意大利语, 日语, 爪哇语, 格鲁吉亚语, 哈萨克语, 高棉语, 康纳达语, 韩语, 库尔德语, 柯尔克孜语, 拉丁语, 老挝语, 立陶宛语, 拉脱维亚语, 马拉加斯语, 马其顿语, 马拉雅拉姆语, 蒙古语, 马拉提语, 马来语, 缅甸语, 尼泊尔语, 荷兰语, 挪威语, 奥里亚语, 旁遮普语, 巴利语, 普什图语, 葡萄牙语, 罗马尼亚语, 俄语, 梵语, 信德语, 僧伽罗语, 斯洛伐克语, 斯洛文尼亚语, 索马里语, 阿尔巴尼亚语, 塞尔维亚语, 巽他语, 瑞典语, 斯瓦希里语, 泰米尔语, 泰米尔语(拉丁化), 泰卢固语, 泰卢固语(拉丁化), 泰语, 他加禄语, 土耳其语, 维吾尔语, 乌克兰语, 乌尔都语, 乌尔都语(拉丁化), 乌兹别克语, 越南语, 意第绪语。

## 效果优化

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

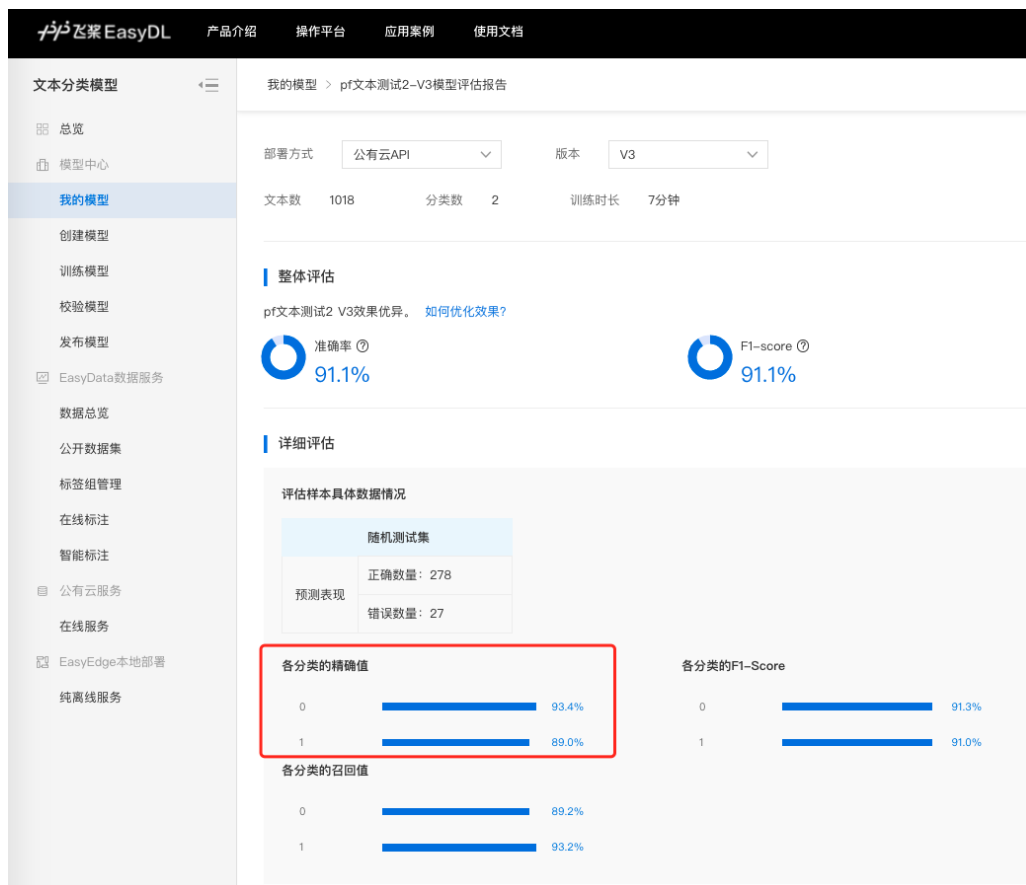
一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

### \*\*检查并优化训练数据\*\*

- 检查是否存在训练数据过少的情况，建议每个类别的文本数量不少于1000个，如果低于这个量级建议扩充
- 检查不同类别的数据量是否均衡，建议不同分类的数据量级相同，并尽量接近，如果有的类别数据量很高，有的类别数据量较低，会影响模型整体的识别效果
- 通过模型效果评估报告中的各分类的详细评估指标，有针对性地扩充训练数据，比如下图中的评估报告显示，标签为1的分类精确值数据表现较标签为0的精确值要差，即可考虑从两个方向进行针对性优化，一是增加标签为1的数据集样本数据量，二是检查现有标签为1数据集中是否存在定义模糊的情况，提升标签为1的数据集质量，优化模型效果。



- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

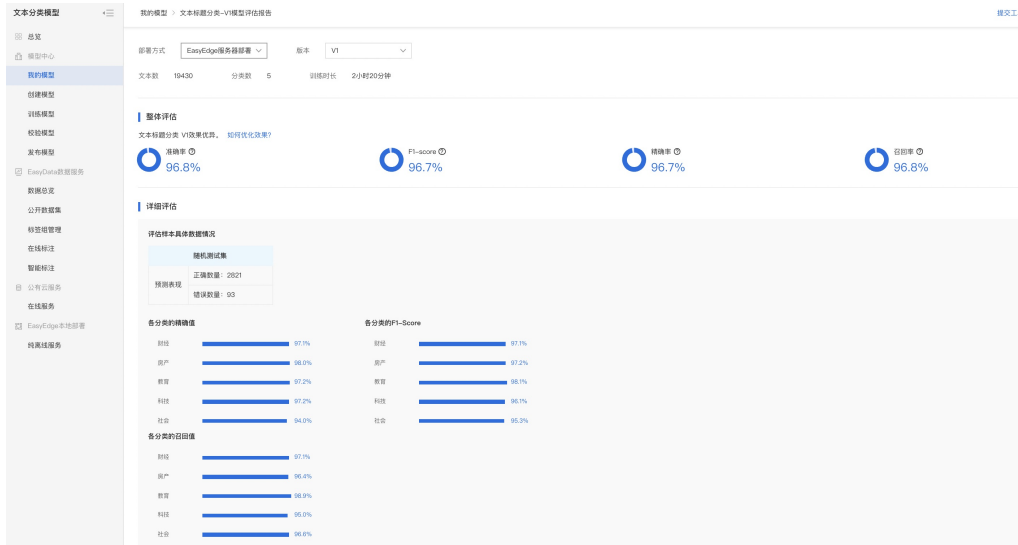
### \*\*选择高精度模型\*\*

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断优化。

## 评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果，以及完整评估结果。



「完整评估结果」页面中将记录整体评估与详细评估的报告，包括该模型整体的准确率、F1-score、精确率、召回率，以及评估样本具体数据情况，各分类的精确值、F1-Score等指标。

整体评估中，各指标的释义如下：

- 准确率：正确分类的样本数与总样本数之比
- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值

如果单个标签的文本量在100条以内，会影响评估指标的科学有效性，请确保提交的训练数据中每个标签的数据量

## \*\*模型校验\*\*

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入文本，点击「校验」后，右侧识别结果栏将输出预测结果
  - 校验数据支持两种输入方式，直接输入文本或上传txt格式文本，文本长度上限为512汉字
  - 在识别结果栏中，可进行阈值的调整。置信度在阈值以下的预测结果将不予显示。例如，阈值调整为0.85时，置信度小于85%的预测结果将不予显示。各类标签的置信度均小于85%时，识别结果栏将显示为：“没有满足条件的识别结果”
4. 若您对预测结果满意，可点击「申请上线」，进行模型的发布

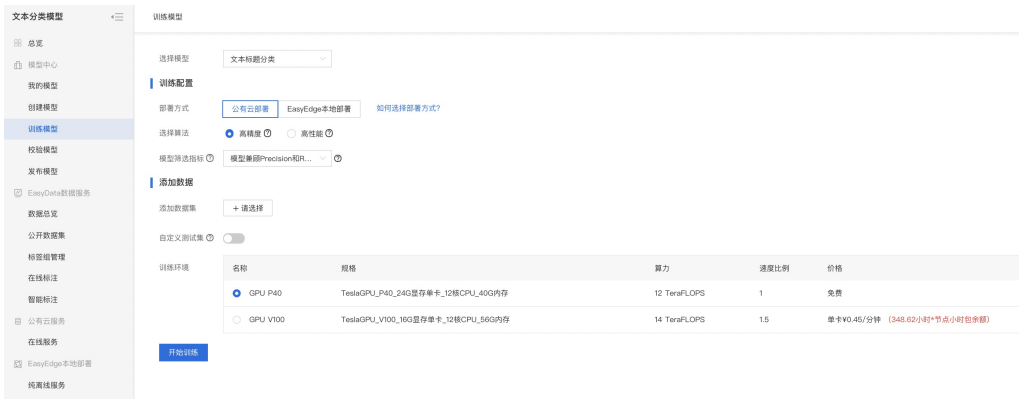
如果单个标签的文本量在100条以内，会影响评估指标的科学有效性，请确保提交的训练数据中每个标签的数据量

## 发起训练

### \*\*训练模型\*\*

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：



**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

### 部署方式

可选择「公有云部署」、「EasyEdge本地部署」。

#### 如何选择部署方式

### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备

### 选择算法

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率将更高。如果您手中的标注数据集样本较少（例如少于1000条），可选择「高精度」的算法。使用高精度的算法训练模型将会耗时更久，实验环境下1000个样本，预计在20分钟左右完成训练
- 如果您选择了高性能的模型，相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在15min内完成训练。同样的数据量情况下，效果比高精度的模型4-5%

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

### 模型筛选指标

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可（兼顾Precision精确度和Recall召回率）。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中并没有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

### Step 3 添加数据

#### 添加训练数据

- 先选择数据集，再按标签选择数据集里的文本，可从多个数据集选择文本。**注意，文本分类模型至少需要选择2个及以上分类**
- 训练时间与数据量大小、选择的算法、训练环境有关。在选择GPU P40、高性能的模型时，10000条训练样本，将在15min内完成训练

#### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

## Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

## 模型部署

### 🔗 整体介绍

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。**公有云API**

- 模型训练完毕后，为了方便企业用户一站式完成AI模型应用，文本分类模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高、在线离线均有调用需求的企业场景。**相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

### 🔗 公有云API

#### 🔗 发布API

#### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」

3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

发布模型界面示意：

**发布完成** 申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。

申请发布通过后，界面和状态示意：

【文本分类】新闻 模型ID: 19900 <span style="float: right;">全部版本 删除</span>						
应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V1	训练完成	未申请	未发布	top1准确率100.00% top5准确率100.00% <a href="#">完整评估结果</a>	<a href="#">申请发布</a> <a href="#">校验</a> <a href="#">训练</a>

## 调用API

### 接口描述

基于自定义训练出的文本分类模型，实现个性化文本识别。模型训练完毕后发布可获得定制化文本分类API 详情访问：[EasyDL 文本分类-单标签](#) 进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明

### 请求示例

HTTP 方法：`POST`

请求URL：请首先在[EasyDL 文本分类-单标签](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "top_num": 6
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
top_num	否	number	-	返回分类数量，默认为6个

请求示例代码

```
Python3

"""
EasyDL 文本分类单标签 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标文本的 本地文件路径，UTF-8编码，最大长度4096汉字**
TEXT_FILEPATH = "【您的测试文本地址，例如：./example.txt】"
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 调用API

### 接口描述

基于自定义训练出的文本分类模型，实现个性化文本识别。模型训练完毕后发布可获得定制化文本分类API 详情访问：[EasyDL 文本分类-单标签](#)进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明

### 请求示例

HTTP 方法：POST



请求URL：请首先在[EasyDL 文本分类-单标签](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "top_num": 6
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
top_num	否	number	-	返回分类数量，默认为6个

请求示例代码

Python3

```
"""
EasyDL 文本分类单标签 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标文本的 本地文件路径，UTF-8编码，最大长度4096汉字**
TEXT_FILEPATH = "【您的测试文本地址，例如：./example.txt】"
```

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 🔗 离线API

## 🔗 发布API

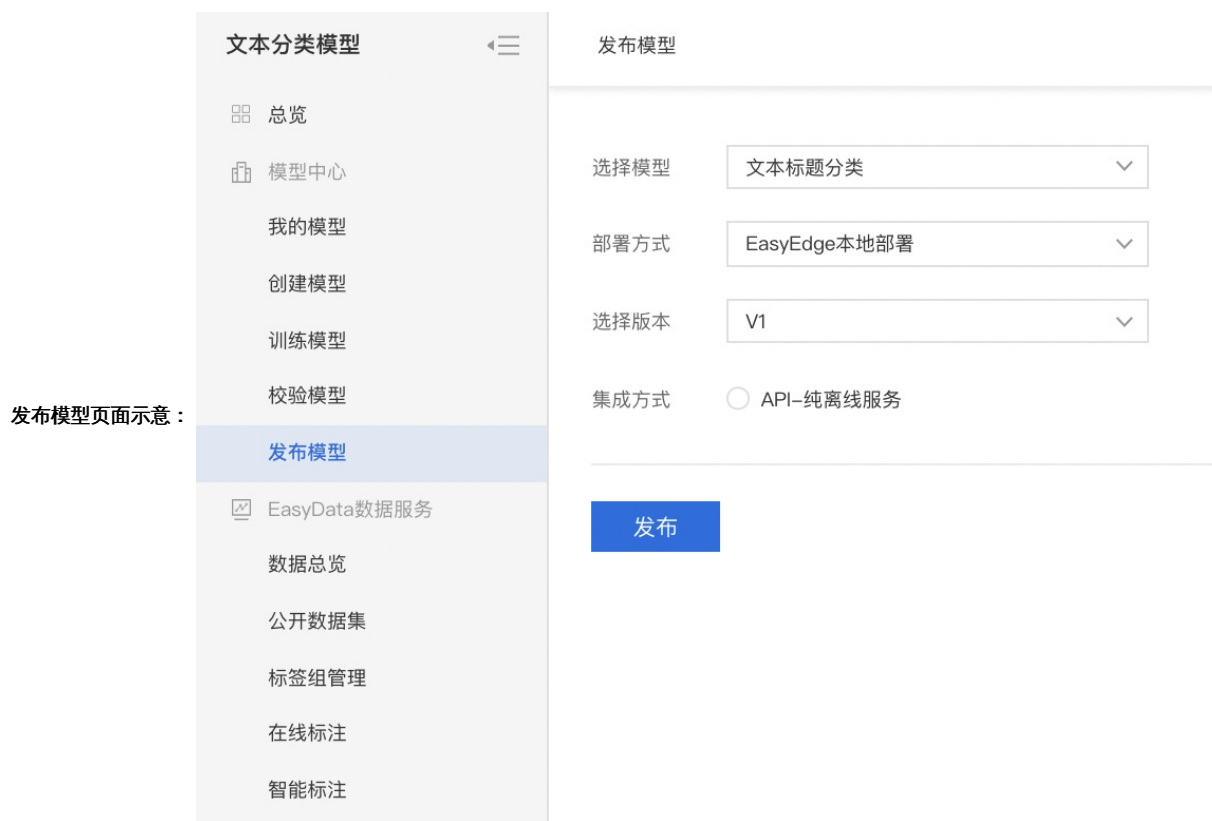
在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

### 私有API介绍

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面



2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

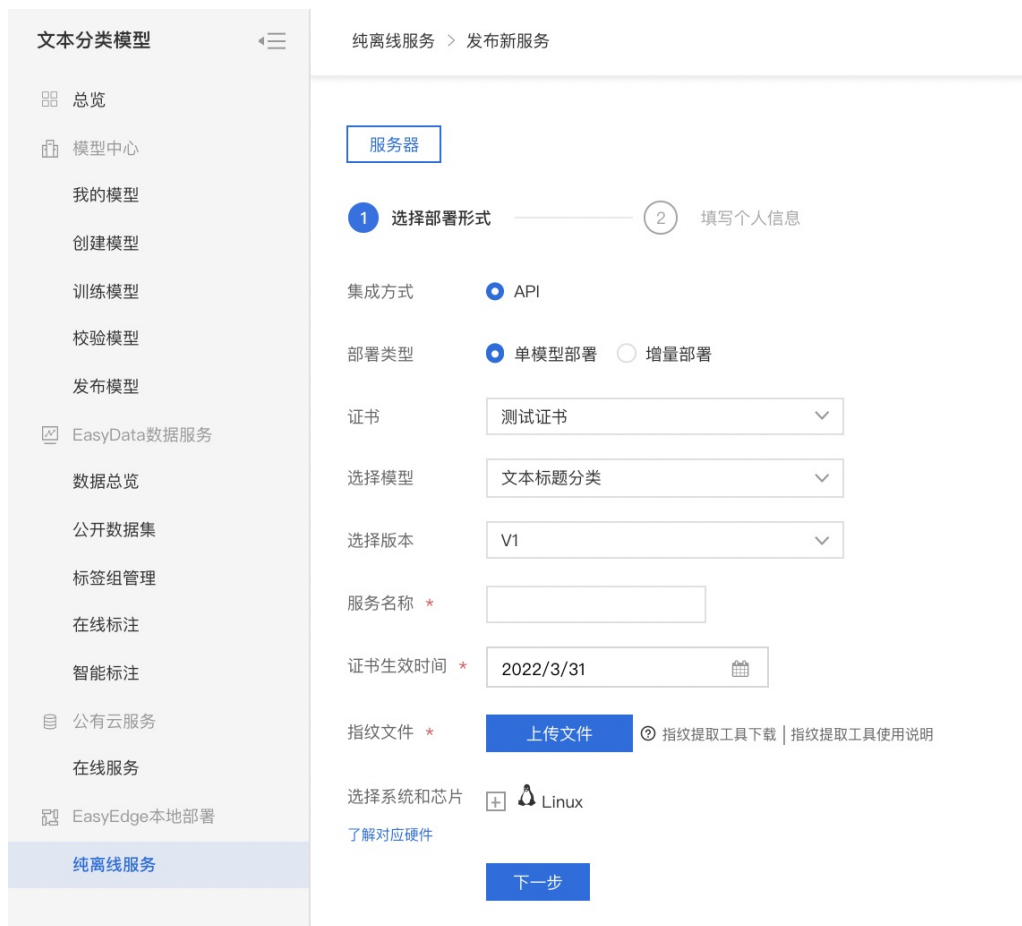
3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取

4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

发布新服务页面示意：



**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

## 调用API

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往[EasyDL 文本分类-单标签](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

## 部署包使用说明 部署方法

EasyDL定制化文本分类模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#)使用python2版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

## 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

## API参考

### 请求说明

### 请求示例

HTTP 方法：`POST`

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/TextClassification](http://{IP}:{PORT}/{DEPLOY_NAME}/TextClassification)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```
{
  "text": "<UTF-8编码数据>",
  "top_num": 5
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
top_num	否	number	-	返回分类数量，默认为6个

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如缺少必要出入参时返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大1024 UTF-8字符。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 nvidia-smi命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如easydl\_\$(DEPLOY\_NAME)\_v2

\$(DEPLOY\_NAME) :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```
**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_$(DEPLOY_NAME)_v2
cd easedl_$(DEPLOY_NAME)_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/$(DEPLOY_NAME) /home/baidu/work/$(DEPLOY_NAME)_V1
**记录当前模型的端口号**
docker ps -a |grep $(DEPLOY_NAME)
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：$(DEPLOY_NAME),前面已备份**
python2 install.py remove $(DEPLOY_NAME)
**安装当前部署包内新的EasyDL服务：$(DEPLOY_NAME)**
python2 install.py install $(DEPLOY_NAME)
**(可选操作) 更新证书**
python2 install.py lu
```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
**（可选操作）进入V1版本部署包所在目录执行license更新操作，假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 发布API

在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

### 私有API介绍

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面

发布模型页面示意：

2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取



4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

发布新服务页面示意：

**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

## 常见问题

### 🔗 训练相关问题

#### \*\*数据处理失败或者状态异常怎么办？\*\*

如是是文本分类模型上传处理失败，请先检查已上传的分类命名是否正确，是否存在特殊字符、或者增加了空格（标签仅支持中英文数字及下划线，长度不超过256字符）；然后检查上传的数据文本量是否超过上限（10万）；再检查文本是否有损坏。如果自查没有发现问题请在百度云控制台内[提交工单](#)反馈。

#### \*\*模型训练失败怎么办？\*\*

如果遇到模型训练失败的情况，请在百度云控制台内[提交工单](#)反馈。

#### \*\*已经上线的模型还可以继续优化吗？\*\*

已经上线的模型依然可以持续优化，操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练，然后在模型列表中更新线上服务，完成模型的优化

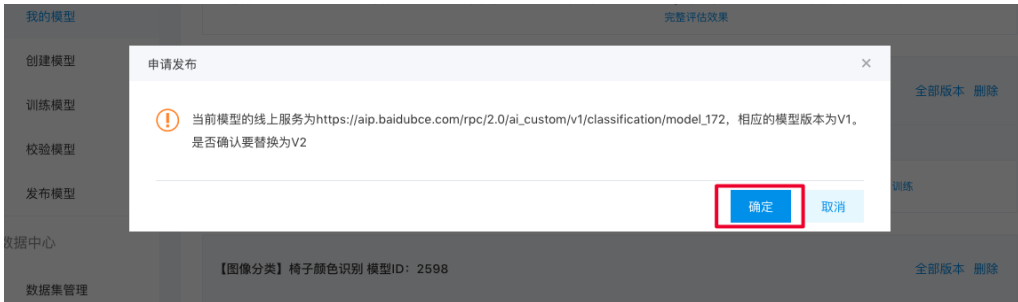
**Step 1 重新训练** 点击我的模型列表——找到需要重新训练的模型——点击训练，进行新版本模型训练

**Step 2 重新发布** 点击我的模型列表——找到新训练好的模型版本——点击申请发布

【图像分类】百美椅子训练 模型ID: 230						全部版本 删除
应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V2	训练完成	未申请	未发布	top1准确率87.61% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">校验</a> <a href="#">训练</a>
离线SDK	V1	训练完成	未申请	未发布	top1准确率85.84% top5准确率100.00% <a href="#">完整评估效果</a>	<a href="#">申请发布</a> <a href="#">训练</a>

每页显示 12 < 1 >

### Step 3 确认发布 在出现的弹窗中点击确定



### 🔗 模型效果相关问题

#### \*\*模型效果怎么调优？\*\*

如果对模型效果不满意，您可以先查看训练数据是否和实际场景要分类的文本一致，以及训练数据量是否太少。

如果训练数据量已经达到一定丰富度（如单个分类/标签的文本量超过2000条以上），效果仍然不佳，请在百度云控制台内[提交工单](#)反馈。

### 🔗 模型上线相关问题

#### \*\*希望加急上线怎么处理？\*\*

请在百度云控制台内[提交工单](#)反馈。

#### \*\*每个账号可以上线几个模型？是否可以删除已上线的模型？\*\*

不限制发布模型数量，已上线模型无法删除

### 🔗 模型部署相关问题

#### \*\*模型发布公有云部署后是否收费？调用量不够怎么办？\*\*

目前EasyDL全部文本任务均提供部分免费调用额度，并支持付费购买额外的调用额度。详情请参考：[EasyDL文本价格说明](#)

#### \*\*模型能否支持私有化部署？\*\*

目前EasyDL已支持将定制模型部署在私有服务器上，只需在发布模型时提交私有服务器部署申请，通过审核后即可获得一个月免费试用，并支持在[控制台](#)在线按设备使用年限购买授权。详情请参考：[文本私有服务器部署价格说明](#)

#### \*\*申请发布模型审核不通过都是什么原因？\*\*

可能的原因包括：

- 1、经过电话沟通当前模型存在问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝。
- 2、电话未接通且模型效果较差，会直接拒绝。

如果需要申诉，请在百度云控制台内[提交工单](#)反馈。

## 文本分类-多标签

### 整体介绍

### 🔗 简介

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

定制文本分类的模型，是基于自建分类体系的机器学习方法，可实现文本按内容类型做自动分类。平台目前提供的文本分类模型包括：文本分类（单标签）和文本分类（多标签）两种模型类型，请您根据自己的业务场景来选择合适的模型。本文介绍的是关于文本分类（多标签）的模型介绍。

文本分类（单标签）场景：如您对网络文章进行舆情分析，判断舆情是正向评价还是负向评价，即每条文本仅有一个分类标准，此问题属于单标签的文本分类场景；

文本分类（多标签）场景：如您对网络文章进行板块划分，即每条文本有两个及以上分类标准，文章可能属于娱乐、国际、生活等多个标签，则可使用多标签的文本分类模型

更多详情访问：[EasyDL自然语言处理方向](#)

## 应用场景

- 1、新闻分类：定制训练媒体文章文本的自动分类，识别文章所属的一个或多个领域标签
- 2、商品名称分类：定制训练商品名称的分类模型，识别商品所属的一个或多个品类
- 3、其他：尽情脑洞大开，训练你希望实现的文本分类多标签的模型

## 技术特色

文本分类模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

### 创建数据集并导入

#### 创建数据集

在训练模型之前，需要创建数据集。需输入数据集名称、选择相应的标注模版、选择数据去重策略，即可创建一个空数据集。

版本	数据集ID	数据量	最近导入状态	标注类型
V1	330727	0	● 导入失败	文本分类

版本	数据集ID	数据量	最近导入状态	标注类型
	tsn_文本-多标签			

**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

**导入数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。

文本分类模型

我的数据总览 > pf文本分类测试/V1/导入

**创建信息**

数据集ID	330727	版本号	V1
备注	<a href="#">🔗</a>		

**标注信息**

标注类型	文本分类	标注模板	短文本多标签
数据总量	0	已标注	0
标签个数	0	待确认	0
大小	0M		

**数据清洗**

暂未做过数据清洗任务

**导入数据**

数据标注状态  无标注信息  有标注信息

导入方式

- 请选择
- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

您可以使用4种方案上传文本分类的数据，分别为：

- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

**本地导入** 您可以通过以下三种方式进行本地数据的导入：



- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传 **以压缩包方式上传**
- 文本文件的编码方式：UTF-8，每个文本文件最长不能超过4096个汉字（字符）
- 压缩包仅支持zip格式；大小需要在5GB以内；

注意，如果您上传的数据是带有标注信息的数据，则需要在压缩包里的创建文件夹，文件夹名即是标签名，只能包含数字/字母/下划线，一个样本有多个标签，则从属于多个文件夹。例如“北京明天气温骤降，请注意保暖”的文本文件同时存在于“北京本地”和“天气”两个文件夹下。

#### 以TXT文本文件上传

- 每行样本最长不能超过4096个汉字（字符），文件编码方式：UTF-8
- txt文件内的标注数据格式要求为“文本内容\t标注标签\t...标注标签\t\n”（\t代表tab制表符，\n代表回车换行），如果是无标注信息的数据，则每行只有文本内容即可

#### 以Excel文件上传

- Excel文件内数据格式要求为：使用第一列作为待标注文本，第二列作为标注信息列（此列仅支持数字或字母），每行是一组样本，每组数据文本内容的字符数建议不超过4096，超出将被截断。
- 文件类型支持xlsx格式，单次上传限制100个文件

**BOS目录导入** 需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

**分享链接导入** 需输入链接地址。分享链接导入的要求如下：

- 仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

#### 平台已有数据集

- 导入无标注数据时，选择需要导入的数据集名称，可导入其不带标注的全部数据，或未标注的数据

- 导入已标注数据时，选择需要导入的数据集名称，可导入其某个或全部标签下的数据

**准备数据集的技巧** 文本分类任务中，可参考以下准备数据集的技巧：**设计分类**

设计整个数据集的分类体系，即抽象出文本所需识别的标签，标签也是你希望识别出的结果。例如娱乐新闻的内容类型，则可以以“男星”、“大陆”、“港台”、“童星”等分别作为标签体系。

**注意：**目前单个模型的上限为1000类，如果要超过这个量级请在百度云控制台内[提交工单反馈](#)

### 数据量

基于设计好的分类标签准备文本数据，每个标签建议至少需要准备50个以上的样本，如果想要较好的效果，建议1000-10000个文本样本，如果某些分类的文本具有相似性，需要增加更多文本。

文本的基本格式要求：目前文本文件类型支持txt，文本文件大小限制长度最大4096，格式为UTF-8字符。一个模型的文本总量限制100万个文本文件。

### 数据分布

- 训练集文本需要和实际场景要识别的文本环境一致
- 考虑实际应用场景的种种可能性，每个分类的文本需要覆盖实际场景里面存在的可能性，训练集若能覆盖的场景越多，模型的泛化能力则越强

### 可能的疑问

- 如果训练文本数据无法全部覆盖实际场景要识别的文本，怎么办？

答：训练的模型算法会有一定的泛化能力，尽可能覆盖即可。

- 多语种模型支持全球94种语言：

南非语, 阿姆哈拉语, 阿拉伯语, 阿萨姆语, 阿塞拜疆语, 白俄罗斯语, 保加利亚语, 孟加拉语, 孟加拉语(拉丁化), 布列塔尼语, 波斯尼亚语, 加泰隆语, 捷克语, 威尔士语, 丹麦语, 德语, 希腊语, 英语, 世界语, 西班牙语, 爱沙尼亚语, 巴斯克语, 波斯语, 芬兰语, 法语, 弗里斯兰语, 爱尔兰语, 苏格兰盖尔语, 加利西亚语, 古吉拉特语, 希伯来语, 印地语, 印地语(拉丁化), 克罗地亚语, 匈牙利语, 亚美尼亚语, 印尼语, 冰岛语, 意大利语, 日语, 爪哇语, 格鲁吉亚语, 哈萨克语, 高棉语, 康纳达语, 韩语, 库尔德语, 柯尔克孜语, 拉丁语, 老挝语, 立陶宛语, 拉脱维亚语, 马拉加语, 马其顿语, 马拉雅拉姆语, 蒙古语, 马拉提语, 马来语, 缅甸语, 尼泊尔语, 荷兰语, 挪威语, 奥里亚语, 旁遮普语, 巴利语, 普什图语, 葡萄牙语, 罗马尼亚语, 俄语, 梵语, 信德语, 僧伽罗语, 斯洛伐克语, 斯洛文尼亚语, 索马里语, 阿尔巴尼亚语, 塞尔维亚语, 巽他语, 瑞典语, 斯瓦希里语, 泰米尔语, 泰米尔语(拉丁化), 泰卢固语, 泰卢固语(拉丁化), 泰语, 他加禄语, 土耳其语, 维吾尔语, 乌克兰语, 乌尔都语, 乌尔都语(拉丁化), 乌兹别克斯坦语, 越南语, 意第绪语。

如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单反馈](#)

## 🔗 在线标注

### 在线标注

**\*\*Step 1 进入标注页面\*\*** 上传未标注的数据后，可以通过以下两个方式进入标注页面：

- 在「数据总览」页面，该数据集对应的操作列下，点击「标注」，即可进入标注页面
- 在「在线标注」页面，选择该数据集，即可进入标注页面

**\*\*Step 2 进行文本标注\*\*** 针对尚未进行标注的数据，通过以下方式进行标注：

- 在右边标签栏中添加标签
- 针对文本内容，选择其对应的标签（可进行多选）
- 点击下一篇，这篇文本的内容即可进行自动保存，且您将开始对下一篇文本进行标注

针对已进行标注的数据，通过以下方式进行标注修改：

- 进入需修改标签的文本的标注页面，选择右边标签栏中的标签（可进行多选）
- 点击下一篇，对此篇文本标签的修改即可进行自动保存



**\*\*Step 3 查看标注信息\*\*** 通过以下方式查看已标注的文本信息：

- 在「数据总览」页面，该数据集对应的操作列下，点击「查看」，进入查看标注页面后，点击「有标注信息」
- 通过选择左侧标签中的不同标签名称，即可查看不同标签下的文本数据



🔗 数据去重

**重复样本的定义**

一个样本包括文本内容和标签。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。例如：

文本内容	标签
未来的学和教正在改变，学生将会在家里学习，机器人将走上讲台。	education/science
未来的学和教正在改变，学生将会在家里学习，机器人将走上讲台。	education/science
未来的学和教正在改变，学生将会在家里学习，机器人将走上讲台。	AI/robot

上表三个样本均为重复样本，后两个样本虽然标签不一，但文本内容一致，也为重复样本。

Tips： “如何利用好重复样本”，如果您在模型训练过程中，需要通过增加某个类别标签的预测权重，可以通过增加此标签的重复样本来达到此目标。

**平台去重策略**

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。注意：当您确定了数据集为去重或非去重的属性后，便不可修改。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

1. 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
2. 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
3. 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

🔗 API上传

本文档主要说明当您线下已有大量的已经完成分类整理的文本数据，如何通过调用API完成文本数据的便捷上传和管理。

EasyDL数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

## 数据集创建API

### 接口描述

该接口可用于创建数据集。

### 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/create>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

## 查看数据集列表API

### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key



### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

### 查看分类（标签）列表API

### 接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

若查看声音分类的全部分类，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

添加数据API

接口描述

该接口可用于在指定数据集添加数据。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

请求说明

请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
dataset_id	是	number	数据集ID
append Label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为 IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION或 TEXT_CLASSIFICATION_MUL时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类单标签和文本分类多标签4096个汉字</b>
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称（由数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

数据集删除API

接口描述

该接口可用于删除数据集。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类(单标签)、文本分类（多标签）
dataset_id	是	number	数据集ID

若删除声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 分类（标签）删除API

### 接口描述

该接口可用于删除分类（标签）。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/label/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

若删除声音分类的子类，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 模型训练

### 效果优化

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

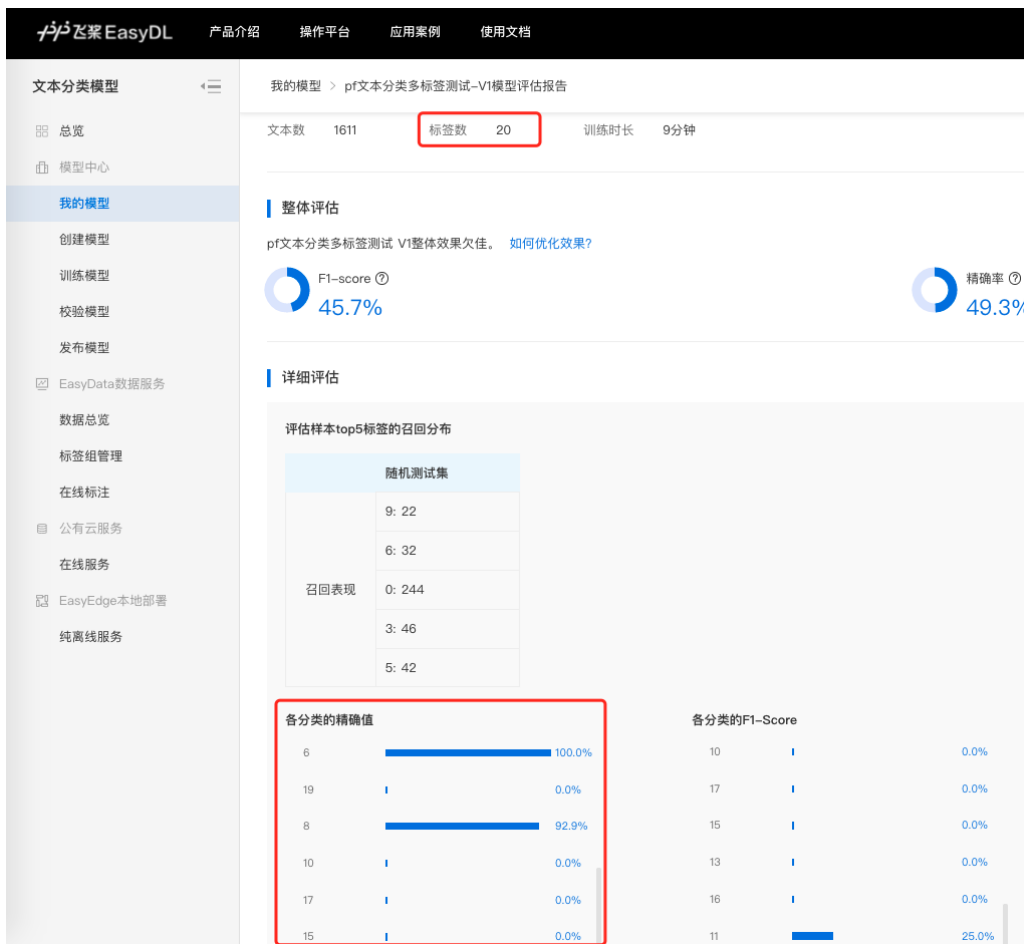
为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

#### **\*\*检查并优化训练数据\*\***

- 检查是否存在训练数据过少的情况，建议每个类别的文本数量不少于1000个，如果低于这个量级建议扩充
- 检查不同类别的数据量是否均衡，建议不同分类的数据量级相同，并尽量接近，如果有的类别数据量很高，有的类别数据量较低，会影响模型整体的识别效果
- 通过模型效果评估报告中的各分类的详细评估指标，有针对性地扩充训练数据。例如下图中评估报告显示有20个标签数，但部分标签（19、17、15等）精确值极低，即可考虑从两个方向进行针对性优化，一是增加相应标签的数据集样本数据量，二是检查现有相应标签数据集中是

否存在定义模糊的情况，提升标签数据集质量，以此达到优化模型的效果。



- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

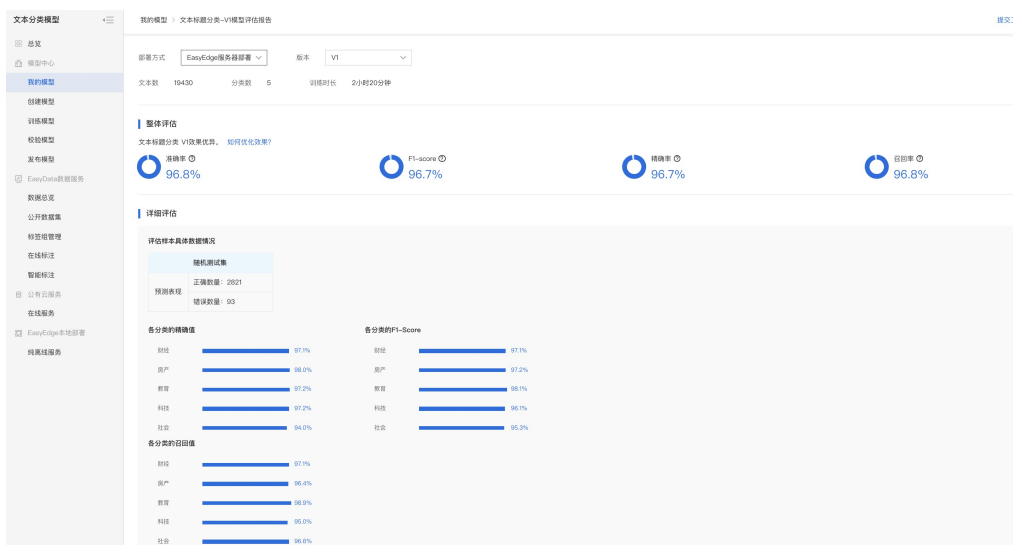
**\*\*选择高精度模型\*\***

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果，以及完整评估结果。



「完整评估结果」页面中将记录整体评估与详细评估的报告，包括该模型整体的准确率、F1-score、精确率、召回率，以及评估样本具体数据情况，各分类的精确值、F1-Score等指标。

整体评估中，各指标的释义如下：

- 准确率：正确分类的样本数与总样本数之比
- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值

如果单个标签的文本量在100条以内，会影响评估指标的科学有效性，请确保提交的训练数据中每个标签的数据量

### \*\*模型校验\*\*

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入文本，点击「校验」后，右侧识别结果栏将输出预测结果
  - 校验数据支持两种输入方式，直接输入文本或上传txt格式文本，文本长度上限为512汉字
  - 在识别结果栏中，可进行阈值的调整。置信度在阈值以下的预测结果将不予显示。例如，阈值调整为0.85时，置信度小于85%的预测结果将不予显示。各类标签的置信度均小于85%时，识别结果栏将显示为：“没有满足条件的识别结果”
4. 若您对预测结果满意，可点击「申请上线」，进行模型的发布

如果单个标签的文本量在100条以内，会影响评估指标的科学有效性，请确保提交的训练数据中每个标签的数据量

## 发起训练

### \*\*训练模型\*\*

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：

名称	规格	算力	速度比例	价格
<input checked="" type="radio"/> GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1	免费
<input type="radio"/> GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	1.5	单卡W0.45/分钟 (348.62/小时+节点小时包余额)

**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

### 部署方式

可选择「公有云部署」、「EasyEdge本地部署」。

### 如何选择部署方式

### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备



## 选择算法

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率将更高。如果您手中的标注数据集样本较少（例如少于1000条），可选择「高精度」的算法。使用高精度的算法训练模型将会耗时更久，实验环境下1000个样本，预计在20分钟左右完成训练
- 如果您选择了高性能的模型，相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在15min内完成训练。同样的数据量情况下，效果比高精度的模型4-5%

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

## 模型筛选指标

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可（兼顾Precision精确度和Recall召回率）。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中并没有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

## Step 3 添加数据

### 添加训练数据

- 先选择数据集，再按标签选择数据集里的文本，可从多个数据集选择文本。**注意，文本分类多标签模型至少需要选择2个及以上分类**
- 训练时间与数据量大小和您选择的模型类型有关，如果您选择的是高性能的模型，使用10000条训练样本将在5min内完成训练；如果您选择的是高精度的模型，使用10000条训练样本，将在60-100min完成训练

### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

## Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

#### 优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

## 模型部署

### 🔗 整体介绍

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。 **公有云API**

- 模型训练完毕后，为了方便企业用户一站式完成AI模型应用，文本分类模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高、在线离线均有调用需求的企业场景。 **相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

### 🔗 公有云API

#### 🔗 发布API

##### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

发布模型界面示意：

**文本分类模型** ☰

- ☰ 总览
- 📁 模型中心
- 我的模型
- 创建模型
- 训练模型
- 校验模型
- 发布模型
- ☑ EasyData数据服务
  - 数据总览
  - 公开数据集
  - 标签组管理
  - 在线标注
  - 智能标注
- 📁 公有云服务
  - 在线服务
- ☑ EasyEdge本地部署
  - 纯离线服务

### 发布模型

选择模型

部署方式

选择版本

服务名称 \*

接口地址 \*

其他要求

0/500

[提交申请](#)

**发布完成** 申请发布后, 通常的审核周期为T+1, 即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况, 请在百度云控制台内[提交工单](#)反馈。

申请发布通过后, 界面和状态示意:

【文本分类】新闻 模型ID: 19900 <span style="float: right;">全部版本 删除</span>						
应用类型	版本	训练状态	申请状态	服务状态	模型效果	操作
云服务	V1	训练完成	未申请	未发布	top1准确率100.00% top5准确率100.00% <a href="#">完整评估结果</a>	<a href="#">申请发布</a> <a href="#">校验</a> <a href="#">训练</a>

## 🔗 调用API

### 接口描述

基于自定义训练出的文本分类模型, 实现个性化文本识别。模型训练完毕后发布可获得定制化文本分类API 详情访问: [EasyDL 文本分类-多标签](#) 进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题, 请在百度云控制台内[提交工单](#)反馈。

### 请求说明

### 请求示例

HTTP 方法: `POST`

请求URL: 请首先在[定制化训练平台](#)进行自定义模型训练, 完成训练后可在服务列表中查看并获取url。

URL参数:

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "threshold": 0.6
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
threshold	否	number	-	对识别的文本标签进行阈值条件的筛选，默认阈值为0.5

#### 请求示例代码

Python3

```
"""
EasyDL 文本分类多标签 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标文本的 本地文件路径，UTF-8编码，最大长度4096汉字**
TEXT_FILEPATH = "【您的测试文本数据地址，例如：./example.txt】"
```

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

#### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

## 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队，在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 调用API

### 接口描述

基于自定义训练出的文本分类模型，实现个性化文本识别。模型训练完毕后发布可获得定制化文本分类API 详情访问：[EasyDL 文本分类-多标签](#)进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[定制化训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "threshold": 0.6
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
threshold	否	number	-	对识别的文本标签进行阈值条件的筛选，默认阈值为0.5

#### 请求示例代码

```
Python3
```

```

"""
EasyDL 文本分类多标签 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""

使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标文本的 本地文件路径，UTF-8编码，最大长度4096汉字**
TEXT_FILEPATH = "【您的测试文本数据地址，例如：./example.txt】"

```

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```

{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}

```

需要重新获取新的Access Token再次请求即可。

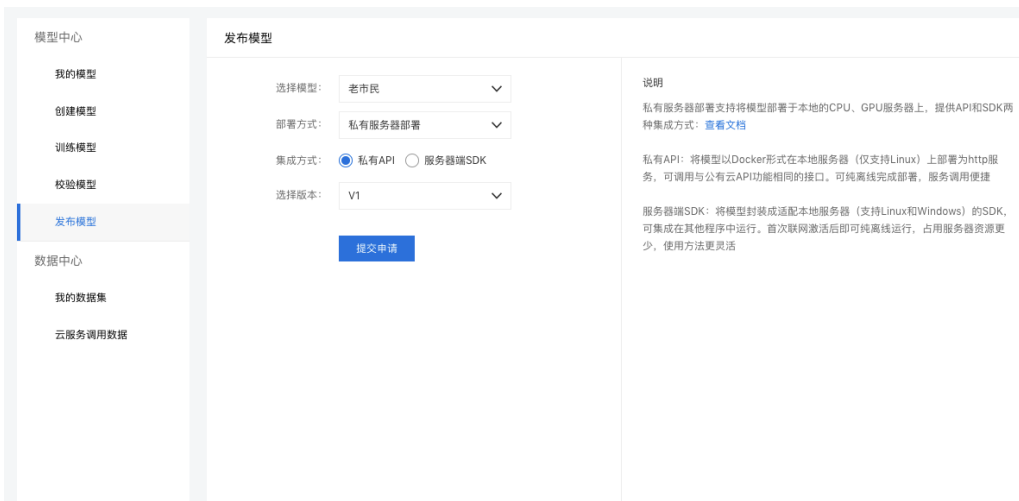
错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队，在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

#### ☞ 离线API

#### ☞ 私有部署服务-文本分类多标签

训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可将模型部署到私有服务器：

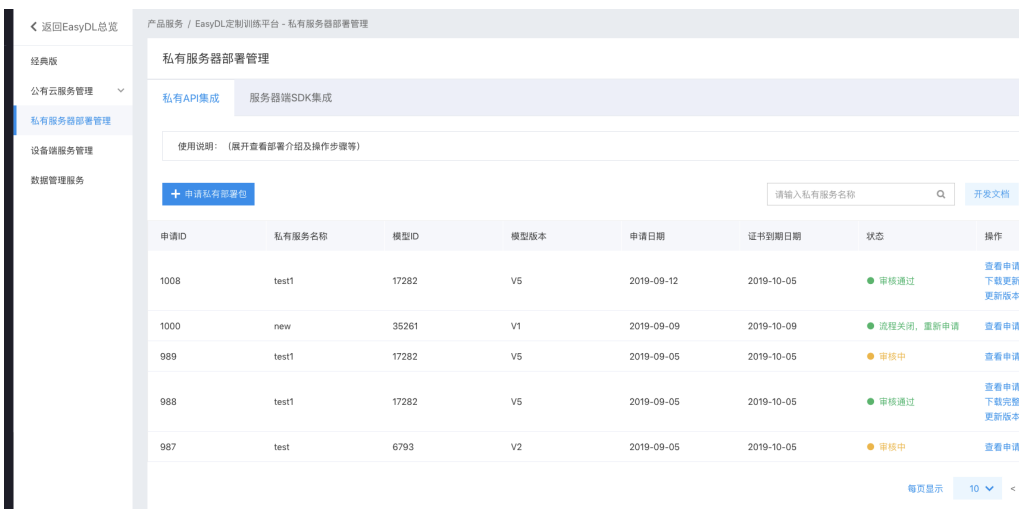




- 选择模型
- 选择部署方式「私有服务器部署」
- 选择集成方式「私有API」

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

点击「提交申请」后，前往控制台申请私有部署包。并[参考文档](#)完成集成



## 私有服务器部署价格说明

EasyDL经典版已支持将定制模型部署在私有服务器上，只需在发布模型时提交私有服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请微信搜索“BaiduEasyDL”添加小助手咨询，通过线下签订合同购买使用。

## 私有部署服务API调用说明-文本分类多标签

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往[EasyDL自然语言处理方向](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

## 部署包使用说明

### 部署方法

EasyDL定制化文本分类模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#)使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

### 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

### 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

### API参考

#### 请求说明

#### 请求示例

HTTP 方法：`POST`

请求URL：请首先在[EasyDL自然语言处理方向](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/TextClassification](http://{IP}:{PORT}/{DEPLOY_NAME}/TextClassification)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```

{
  "text": "<UTF-8编码数据>",
  "top_num": 5
}

```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字
top_num	否	number	-	返回分类数量，默认为6个

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如缺少必要出入参时返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如`easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}`：申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxvf xx.tar.gz
**解压后,进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务: ${DEPLOY_NAME},前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务: ${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}
** (可选操作) 更新证书**
python2 install.py lu

```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
** (可选操作) 进入V1版本部署包所在目录执行license更新操作,假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录,参考上述【模型更新】步骤,执行模型升级操作(即先卸载v2,后升级为v1)

## 情感倾向分析

### 整体介绍

#### 任务简介

定制情感倾向分析模型,可实现文本按情感的正向(positive)和负向(negative)做自动分类。您只需提供正向和负向标签对应的训练数据,即可训练获得情感倾向分析模型。

更多详情访问：[EasyDL自然语言处理方向](#)

#### 应用场景

1. 电商评论分类：可对商品的评论信息进行分类,将不同用户对同一商品的评论内容按情感极性予以分类展示
2. 商品舆情监控：通过对产品多维度评论观点进行倾向性分析,给用户该产品全方位的评价,方便用户进行决策
3. 舆情分类：通过对需要舆情监控的实时文字数据流进行情感倾向性分析,把握用户对热点信息的情感倾向性变化
4. 其他：尽情脑洞大开,训练你希望实现的情感倾向分析的模型

#### 技术特色

情感倾向分析模型内置**文心大模型**,将大数据预训练与多源丰富知识相结合,通过持续学习技术,不断吸收海量文本数据中词汇、结构、语义等方面的新知识,实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

### 创建数据集并导入

#### 创建数据集

在训练模型之前，需要先在数据总览【创建数据集】。只需输入数据集名称、选择数据去重策略，即可创建一个空数据集。

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态
V1	336503	0	● 已完成	情感倾向分析	0% (0/0)

**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

**导入数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。

情感倾向分析

我的数据总览 > 1111111/V1/导入

创建信息

数据集ID	346839	版本号	V1
备注	<a href="#">编辑</a>		

标注信息

标注类型	情感倾向分析	标注模板	情感倾向分析
数据总量	0	已标注	0
标签个数	2	待确认	0
大小	0M		

数据清洗

暂未做过数据清洗任务

导入数据

数据标注状态  无标注信息  有标注信息

导入方式

- 请选择
- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

您可以使用4种方案上传情感倾向分析的数据，分别为：

- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

不论您上传无标注信息的数据或有标注信息的数据，都需要以下述格式要求进行上传。区别在于上传有标注信息的数据时，需要针对正向文本内容与负向文本内容分别进行上传。每个数据集里面默认包含正向（positive）标签和负向（negative）标签。

**本地导入** 您可以通过以下三种方式进行本地数据的导入：

- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传 **以压缩包方式上传**
- 一个文本文件保存一个样本，文本文件的编码方式：UTF-8，每个文本文件最长不能超过512个汉字（字符）
- 压缩包仅支持zip格式；大小需要在5GB以内

**以TXT文本文件上传**

- 一个文本文件包含多个样本，文本文件中每行为一个样本
- 一个文本文件包含一个样本，单次上传限制100个文件，最多可上传100万个文本文件
- 每行样本最长不能超过512个汉字（字符），文件编码方式：UTF-8

**以Excel文件上传**

- Excel文件上传数据格式为每行是一个样本，每个数据文本内容的字符数建议不超过512个，超出将被截断

- 文件类型支持xlsx格式，单次上传限制100个文件
- 需确保上传的样本在sheet1中，且数据都在首列

**BOS目录导入** 需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

**分享链接导入** 需输入链接地址。分享链接导入的要求如下：

- 仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

**平台已有数据集**

- 导入无标注数据时，选择需要导入的数据集名称，可导入其不带标注的全部数据，或未标注的数据
- 导入已标注数据时，选择需要导入的数据集名称，可导入其某个或全部标签下的数据

**准备数据集的技巧** 情感倾向分析任务中，可参考以下准备数据集的技巧：**设计分类**

情感倾向分析的数据集，默认只使用正向和负向两种标签的数据，所以数据集中无需创建标签，您只需准备对应情感倾向的标签数据即可。

**数据量**

每个标签建议至少需要准备50个以上的样本，如果想要较好的效果，建议准备1000-10000个文本样本，如果某些分类的文本具有相似性，需要增加更多文本。**数据分布**

- 训练集文本需要和实际场景要识别的文本内容的业务范围一致，且标签对应文本的数量分布一致。如训练集的业务范围是图书商品的情感倾向分析，而预计线上对应的场景或业务是电子产品的情感倾向分析，此时两者不一致，将会导致模型实际应用效果不佳
- 考虑实际应用场景有多种可能性，每个场景都需要准备相对应的训练数据，训练集若能覆盖的场景越多，模型的泛化能力则越强
- 建议对高频的业务场景尽量做到覆盖，并通过线上bad case来进行训练数据的优化

如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单](#)反馈

🔗 **在线标注**

**在线标注**

**\*\*Step 1 进入标注页面\*\*** 上传未标注的数据后，可以通过以下两个方式进入标注页面：

- 在「数据总览」页面，该数据集对应的操作列下，点击「标注」，即可进入标注页面

The screenshot shows the '我的数据总览' (My Data Overview) page in the EasyDL console. The page has a navigation menu on the left with '数据总览' (Data Overview) selected. The main content area displays a table of datasets. The table has columns: 版本 (Version), 数据集ID (Dataset ID), 数据量 (Data Volume), 最近导入状态 (Latest Import Status), 标注类型 (Label Type), 标注状态 (Label Status), 清洗状态 (Cleaning Status), and 操作 (Action). Two datasets are listed: V2 (ID: 296177, Volume: 100, Status: 已完成) and V1 (ID: 296016, Volume: 112, Status: 已完成). Both have '情感倾向分析' as the label type and '100%' as the label status. The '操作' column for V2 contains buttons for '查看', '多人标注', '导入', and '标注'. The '标注' button is highlighted with a red box. Below the table, there is a section for 'binbin\_test\_1127' (ID: 253950) with buttons for '新增版本', '全部版本', and '删除'.

- 在「在线标注」页面，选择该数据集，即可进入标注页面



针对尚未进行标注的数据，通过以下方式进行标注：

- 选择右边标签栏中的标签。若该文本内容为积极情感，则选择positive；若该文本内容为消极情感，则选择negative。系统默认选择上方的标签
- 点击下一篇，此篇文本的内容即可进行自动保存，且您将开始对下一篇文本进行标注

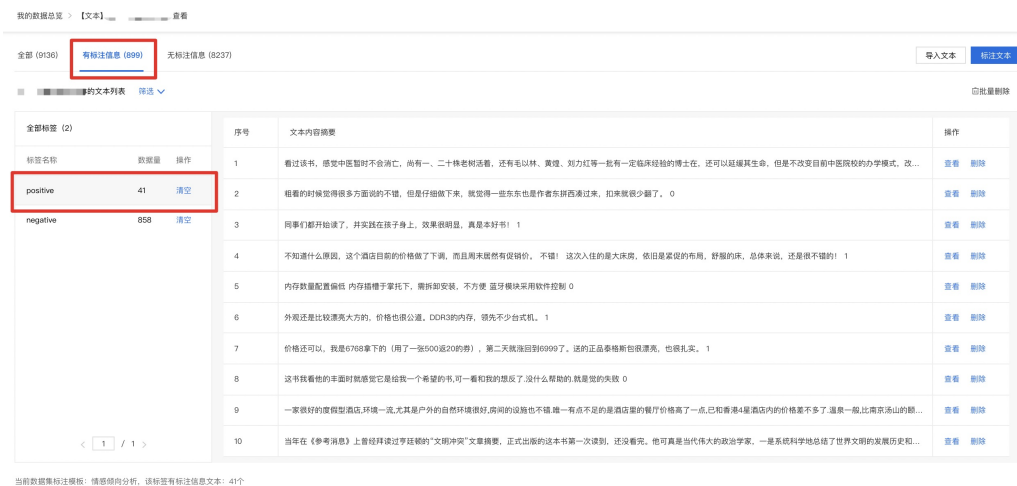
针对已进行标注的数据，通过以下方式进行标注修改：

- 进入需修改标签的文本的标注页面，选择右边标签栏中的标签
- 点击下一篇，对此篇文本标签的修改即可进行自动保存



标注技巧: 对于批量文本需要标注为同一个标签的情况，可以在右侧标签区域将标签置顶，进而提高标注效率 \*\*Step 3 查看标注信息\*\* 通过以下方式查看已标注的文本信息：

- 在「数据总览」页面，该数据集对应的操作列下，点击「查看」，进入查看标注页面后，点击「有标注信息」
- 通过选择左侧标签中的不同标签名称，即可查看不同标签下的文本数据



### 🔗 数据去重

#### 重复样本的定义

一个样本包括文本内容和标签。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。

例如：

文本内容	标签
这个酒店的地段不错，交通方便	1
这个酒店的地段不错，交通方便	1
这个酒店的地段不错，交通方便	0



上表三个样本均为重复样本，后两个样本虽然标签不一，但文本内容一致，也为重复样本。

Tips：如果您在模型训练过程中，需要通过增加某个类别标签的预测权重，可以通过增加此标签的重复样本来达到此目标

### 平台去重策略

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。

注意：当您确定了数据集为去重或非去重的属性后，便不可修改

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

- 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
- 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
- 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

### API上传

本文档主要说明当您线下已有大量的已经完成分类整理的文本数据，如何通过调用API完成文本数据的便捷上传和管理。

EasyDL经典版数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

### 数据集创建API

#### 接口描述

该接口可用于创建数据集。

#### 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/create

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

查看数据集列表API

接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

请求说明

请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/list

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL, SENTI_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）、情感倾向分析
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态, 包括shared、smart和空值, 分别表示共享中、智能标注中、非特殊状态

## 查看分类 (标签) 列表API

## 接口描述

该接口可用于查看分类 (标签)。返回分类 (标签) 的名称、包含数据量等信息。

## 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

## 请求说明

## 请求示例

HTTP 方法 : POST

请求URL : <https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下 :

参数	值
Content-Type	application/json

Body中放置请求参数, 参数详情如下 :

## 请求参数

字段	必选	类型	说明
type	是	string	数据集类型, 可包括: IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL, SENTI_CLASSIFICATION 分别对应: 图像分类、物体检测、图像分割、声音分类、文本分类 (单标签)、文本分类 (多标签)、情感倾向分析
dataset_id	是	number	数据集ID
start	否	number	起始序号, 默认0
num	否	number	数量, 默认20, 最多100

若查看声音分类的全部分类, 在type参数应传「SOUND\_CLASSIFICATION」

## 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法 : `POST`

请求URL : `https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity`

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下 :

参数	值
Content-Type	application/json

Body中放置请求参数, 参数详情如下 :

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL, SENTI_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类（单标签）、文本分类（多标签）、情感倾向分析
dataset_id	是	number	数据集ID
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION或TEXT_CLASSIFICATION_MUL时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类单标签和多标签为4096个汉字；情感倾向分析为512个汉字</b>
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称（由数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

##### 接口描述

该接口可用于删除数据集。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION, TEXT_CLASSIFICATION_MUL 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类(单标签)、文本分类（多标签）
dataset_id	是	number	数据集ID

若删除声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 分类（标签）删除API

情感倾向分析，不可删除分类标签。

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 模型训练

### 🔗 创建模型

#### \*\*创建模型\*\*

在模型中心目录中选择「创建模型」，填写模型名称、模型归属、所属行业、应用场景、邮箱地址、联系方式、功能描述等信息，即可创建模型。

目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。

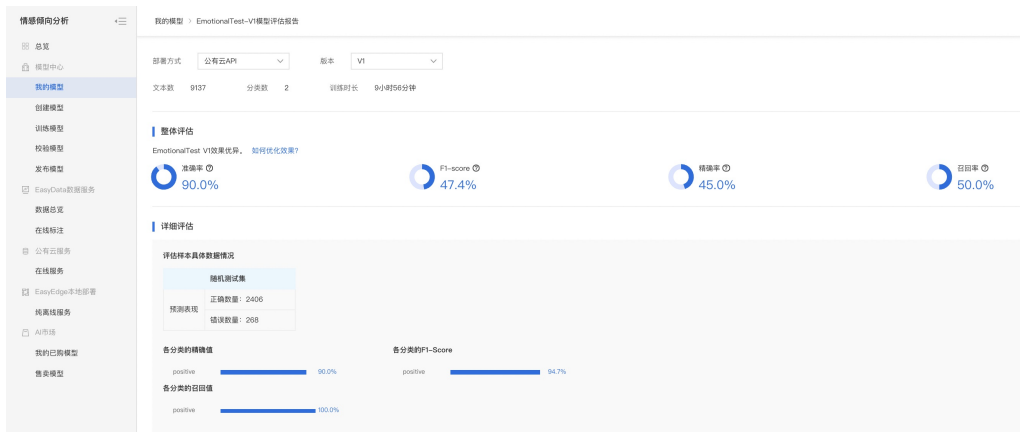
模型创建成功后，即可在「我的模型」中看到刚刚创建的模型。

注：

1. 创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型。
2. 目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练。
3. 如果您是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务。

## 评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果，以及完整评估结果。



「完整评估结果」页面中将记录整体评估与详细评估的报告，包括该模型整体的准确率、F1-score、精确率、召回率，以及评估样本具体数据情况，各分类的精确值、F1-Score、召回值等指标。

整体评估中，各指标的释义如下：



- 准确率：正确分类的样本数与总样本数之比
- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值

如果单个标签的文本量在100条以内，会影响评估指标的科学有效性，请确保提交的训练数据中每个标签的数据量

### **\*\*模型校验\*\***

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入文本，点击「校验」后，右侧识别结果栏将输出预测结果
  - 校验数据支持两种输入方式，直接输入文本或上传txt格式文本，文本长度上限为512汉字
  - 在识别结果栏中，可进行阈值的调整。置信度在阈值以下的预测结果将不予显示。例如，阈值调整为0.85时，置信度小于85%的预测结果将不予显示。当positive与negative两类标签的置信度均小于85%时，识别结果栏将显示为：“没有满足条件的识别结果”
4. 若您对预测结果满意，可点击「申请上线」，进行模型的发布

### 🔗 效果优化

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

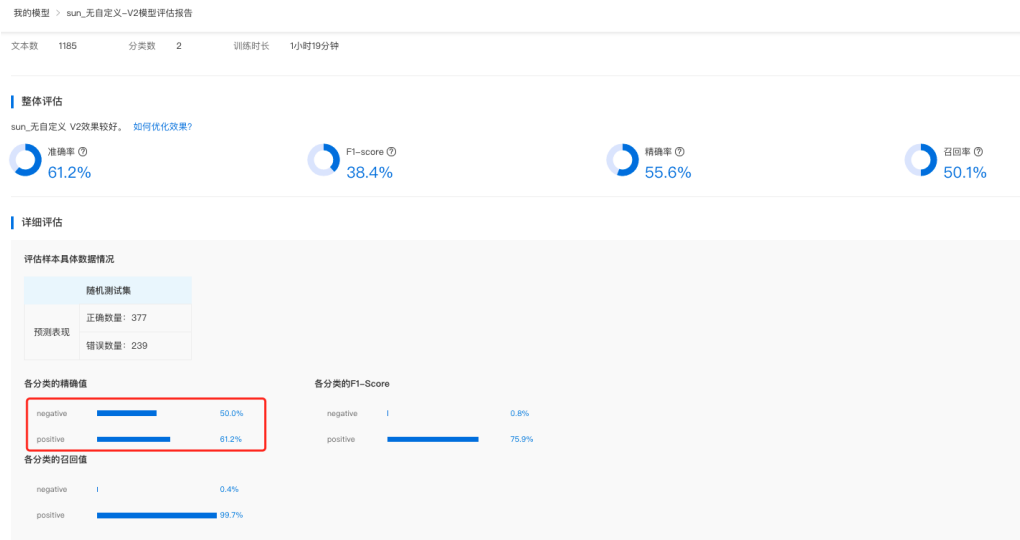
一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

### **\*\*检查并优化训练数据\*\***

- 检查是否存在训练数据过少的情况，建议每个类别的文本数量不少于1000个，如果低于这个量级建议扩充
- 检查不同类别的数据量是否均衡，建议不同分类的数据量级相同，并尽量接近，如果有的类别数据量很高，有的类别数据量较低，会影响模型整体的识别效果
- 通过模型效果评估报告中的各分类的详细评估指标，有针对性地扩充训练数据，比如下图中的评估报告显示，负向分类精确值数据表现较差，即可考虑从两个方向进行针对性优化，一是增加负向数据集样本数据量，二是检查现有负向数据集中是否存在定义模糊的情况，提升负向数据集质量，优化模型效果。



- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

**\*\*选择高精度模型\*\***

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

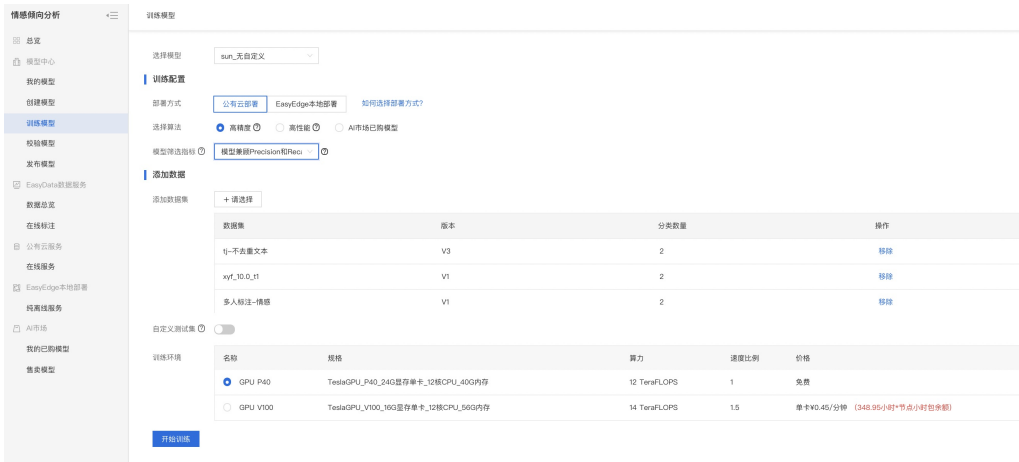
「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断优化

发起训练

**\*\*训练模型\*\***

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：



**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

**部署方式**

可选择「公有云部署」、「EasyEdge本地部署」。

[如何选择部署方式](#)

**选择设备**

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备

**选择算法**

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率将更高。使用10000条训练样本，将在60-100min完成训练
- 如果您选择了高性能的模型，相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在5min内完成训练

如果您已从AI市场购买了模型算法，也可以基于已购模型的算法训练：[前往AI市场购买](#)

「高精度」算法内置[文心大模型](#)，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

### 模型筛选指标

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中并没有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

### Step 3 添加数据

#### 添加训练数据

- 先选择数据集，再按标签（positive、negative）选择数据集里的文本，可从多个数据集选择文本
- 训练时间与数据量大小、选择的算法、训练环境有关。在选择GPU P40、高性能的模型时，10000条训练样本，将在5min内完成训练

#### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

### Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

## 模型部署

### 🔗 整体介绍

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。**公有云API**

模型训练完毕后，为了方便企业用户一站式完成AI模型应用，情感倾向分析模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。适用于绝大多数通用情感倾向分析场景，如电商评论分析、商品舆情监控等。

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高的企业场景，如医美平台用户评论分析。**相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

### 🔗 公有云API

### 🔗 发布API

#### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

发布模型界面示意：

**情感倾向分析** ☰

☰ 总览

📁 模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

📁 EasyData数据服务

数据总览

在线标注

📁 公有云服务

在线服务

📁 EasyEdge本地部署

纯离线服务

📁 AI市场

我的已购模型

售卖模型

发布模型

---

选择模型

部署方式

选择版本

服务名称 \*

接口地址 \*

其他要求 

若接口无法满足您的需求，请描述希望解决的问题，500汉字以内

0/500

提交申请

**发布完成** 申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。

申请发布通过后，界面和状态示意：

【情感倾向分析】 gq_sen_调参后高精度 模型ID: 10125						☰ 训练	🕒 历史版本	🗑️ 删除
部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作		
公有云API	V1	训练完成	审核成功	已发布	准确率: 92.85% F1-score : 0.93 <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">服务详情</a> <a href="#">校验</a>		

## 🔗 调用API

### 接口描述

基于自定义训练出的情感倾向分析模型，实现个性化情感倾向分析。模型训练完毕后发布可获得定制化情感倾向分析API。详情访问：[EasyDL 情感倾向分析](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管理员

### API参考 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL 情感倾向分析](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

**在线调试** EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

**错误码**

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队，在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 调用API

### 接口描述

基于自定义训练出的情感倾向分析模型，实现个性化情感倾向分析。模型训练完毕后发布可获得定制化情感倾向分析API。详情访问：[EasyDL 情感倾向分析](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管理员

### API参考 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL 情感倾向分析](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>"
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度4096汉字

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

**在线调试** EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。



错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队，在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 🔗 离线API

## 🔗 发布API

在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

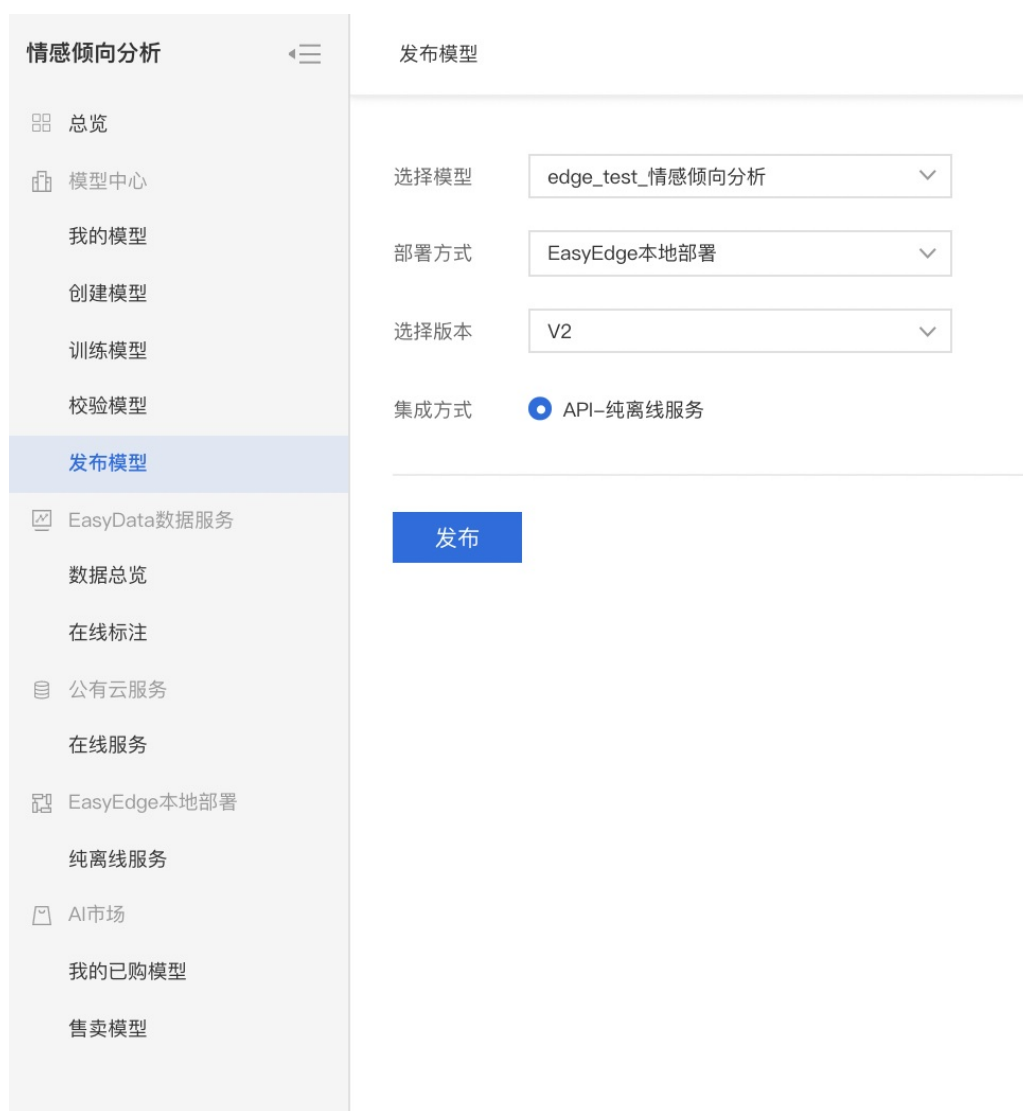
### 私有API介绍

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面

**发布模型页面示意：**



2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取

4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

发布新服务页面示意：

**情感倾向分析** <三

- 总览
- 模型中心
  - 我的模型
  - 创建模型
  - 训练模型
  - 校验模型
  - 发布模型
- EasyData数据服务
  - 数据总览
  - 在线标注
- 公有云服务
  - 在线服务
- EasyEdge本地部署
- 纯离线服务
- AI市场
  - 我的已购模型
  - 售卖模型

纯离线服务 > 发布新服务

服务器

1 选择部署形式
 2 填写个人信息

集成方式  API

部署类型  单模型部署  增量部署

证书

选择模型

选择版本

服务名称 \*

服务名称不能为空

证书生效时间 \*

指纹文件 \*  [① 指纹提取工具下载](#) | [指纹提取工具使用说明](#)

指纹文件不能为空

选择系统和芯片  Linux

了解对应硬件 选择系统和芯片不能为空

**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

## 调用API

**接口描述** 本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请前往[EasyDL 情感倾向分析](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#) ,与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管理员

## 部署包使用说明 部署方法

EasyDL情感倾向分析模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#) 使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

## 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询。

## API参考

### 请求说明

### 请求示例

HTTP 方法：[POST](#)

请求URL：请首先在[EasyDL自然语言处理方向](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/SentimentClassification](http://{IP}:{PORT}/{DEPLOY_NAME}/SentimentClassification)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```
{
  "text": "<UTF-8编码数据>"
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个汉字，超出将被截断

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如缺少必要出入参时返回：

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大4096 UTF-8字符。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 nvidia-smi命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如easydl\_\$(DEPLOY\_NAME)\_v2

\$(DEPLOY\_NAME) :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```
**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_$(DEPLOY_NAME)_v2
cd easedl_$(DEPLOY_NAME)_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/$(DEPLOY_NAME) /home/baidu/work/$(DEPLOY_NAME)_V1
**记录当前模型的端口号**
docker ps -a |grep $(DEPLOY_NAME)
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：$(DEPLOY_NAME)，前面已备份**
python2 install.py remove $(DEPLOY_NAME)
**安装当前部署包内新的EasyDL服务：$(DEPLOY_NAME)**
python2 install.py install $(DEPLOY_NAME)
**（可选操作）更新证书**
python2 install.py lu
```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
** (可选操作) 进入V1版本部署包所在目录执行license更新操作,假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 文本实体关系抽取

### 整体介绍

#### 简介

定制实体关系抽取模型，是指从文本中抽取预定义的实体类型及实体间的关系类型，得到包含语义信息的实体关系三元组，每个实体关系三元组由两个实体及其关系构成。

更多详情访问：[EasyDL自然语言处理方向](#)

#### 应用场景

对内容中的关键实体及实体间的关系类型进行识别和抽取，如：

- 金融研报信息识别
- 法律案件抽取
- 其他：尽情脑洞大开，训练你希望实现的文本实体关系抽取的模型

#### 技术特色

文本实体关系抽取模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化，当前模型主要应用于中文环境下的语义分析。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

#### 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



### 数据准备

#### 创建数据集并导入

##### 创建数据集

在训练模型之前，需要在【数据总览】里面“创建数据集”。需输入数据集名称、选择相应的标注模版、选择数据去重策略，即可创建一个空数据集。



**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

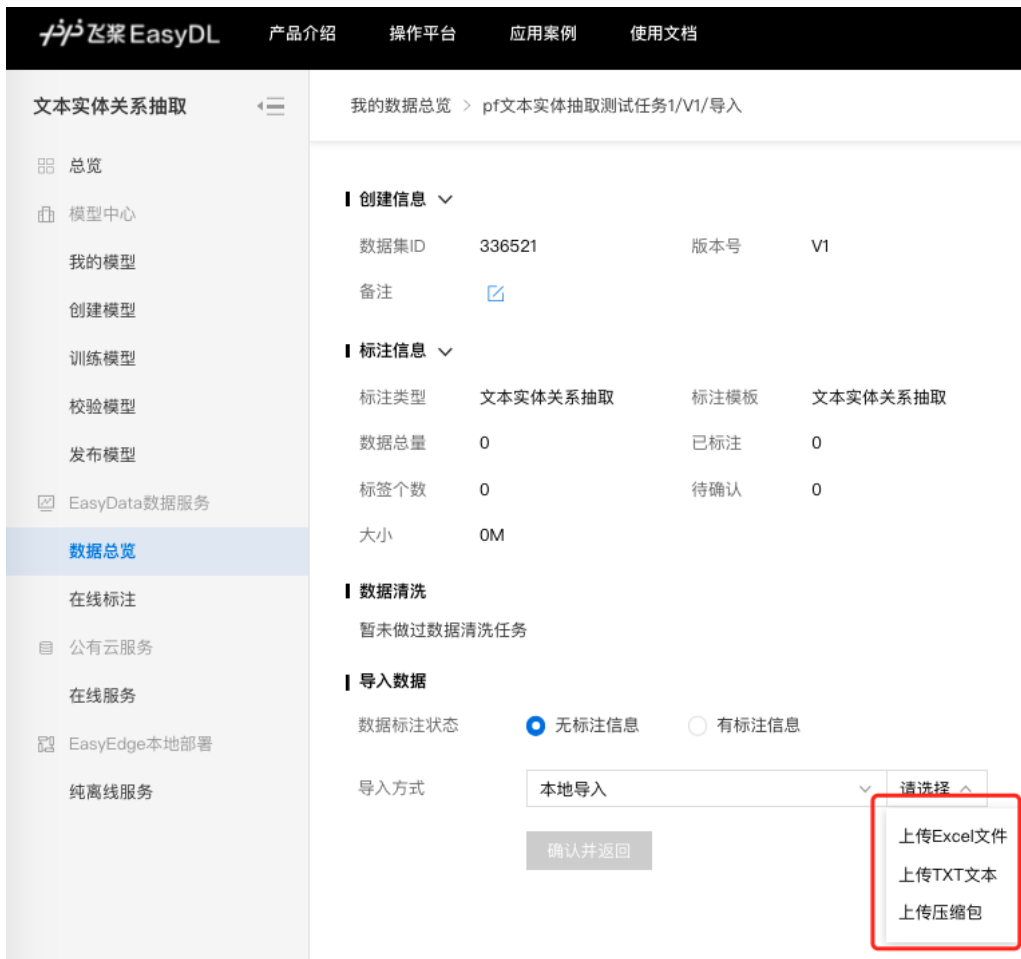
**\*\*导入无标注数据\*\*** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。



您可以使用3种方案上传数据，分别为：

- 本地导入
- BOS目录导入
- 分享链接导入

**本地导入** 您可以通过以下三种方式进行本地数据的导入：



- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传

通过压缩包上传时，需注意：

- 压缩包内每一个txt文件为一个样本，文本文件编码须为UTF-8，每个样本字符数不得超过512个字符（包括汉字、数字、符号等），超出将被截断
- 压缩包的格式为zip；压缩包最大不超过5G；[详见数据样例](#)

通过TXT文本上传时，需注意：

- 文本实体关系抽取数据txt文件中，每一行为一个样本，文本文件的编码格式须为UTF-8，每个样本字符数不得超过512个字符（包括汉字、数字、符号等），超出将被截断，[详见数据样例](#)。
- 文件格式支持txt格式，单次可上传100个文件，最多可上传100万个文件。

通过Excel文件上传时，需注意：

- 如果您上传的文本实体关系抽取数据未Excel文件，那么要求您的Excel文件每行为一个样本，每个样本字符数不得超过512个字符（包括汉字、数字、符号等），超出将被截断。注意，表头作为首行将被系统忽略。
- 文件格式支持xlsx格式，单次可上传100个文件，[详见数据样例](#)。

**\*\*BOS目录导入\*\***

需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

分享链接导入 需输入链接地址。分享链接导入的要求如下：

仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接 导入有标注数据 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。



**已标注数据上传方式：** 针对已标注的文本实体关系抽取数据集上传目前本平台仅支持Excel文件格式上传。

以Excel文件导入

- 要求上传的Excel文件，首行为表头，表头表示每一列代表的数据类型，依次为“文本内容、实体关系1、实体关系2、..”。其中实体关系内格式为：{实体1位置,实体1类别},{实体2位置,实体2类别},实体关系。每个标注内均以英文逗号间隔，且内容顺序不可变。[详见数据样例](#)。
- 第二行起每行为一个样本，每个样本文本内容字符数不得超过512个字符（包括汉字、符号、数字等），超出将报错；
- 目前Excel文件格式支持xlsx格式，单次可上传100个文件；文本样例如下。

文本内容	实体关系1
今年年初，党中央、国务院根据国内外经济形势的变化，及时作出扩大内需、保持经济持续快速增长的重大决策。	{{5,7},ORG}, {{9,11},ORG},lead

**准备数据集的技巧** 在每个数据集项目中可以包含多个实体及其关系的文本数据，每个文本数据的实体数量以及关系数据可以不同。以下是文本实体关系抽取任务的小tips，请您查收：

- 思考实体类型：根据您所需要的具体场景，来考虑您的文本数据中包含的实体类型数量
- 思考实体关系类型：根据您已有的实体，考虑各实体之间的关系

**可能的疑问**

- 什么是实体关系抽取？

答：实体关系抽取是指从文本中抽取预定义的实体类型及实体间的关系类型，得到包含语义信息的实体关系三元组，每个实体关系三元组由两个实体及其关系构成，如<实体1，实体2，实体关系>

- 如果训练文本数据无法全部覆盖实际场景要识别的文本，怎么办？

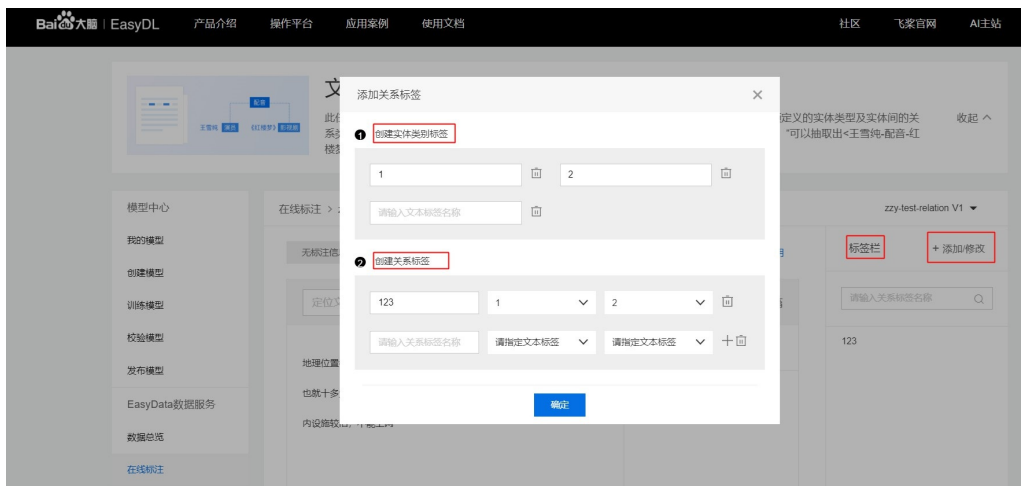
答：训练的模型算法会有一定的泛化能力，尽可能覆盖即可

**在线标注**

**\*\*文本实体关系抽取数据标注\*\***

如果您上传的是文本实体关系抽取数据集为无标注数据集，为了整个模型的正确运行，您可以点击【去标注】根据以下两步操作完成实体关系抽取的数据标注工作。首先您需要创建实体类别，并设置实体关系，第二步您需要根据您上传的文本实体关系抽取数据，对每个样本选择实体，标注实体类别和关系。下面将详细介绍以上两个步骤。**创建实体类别，设置实体关系** 首先，我们需要对上传的无标注数据进行实体类型的设置与实体关系类型的设置。如下图所示，首先点击右上角的【添加/修改】功能。

在弹出来的弹框中，创建实体类别标签，实体类别选择时相互独立。在创建关系标签时，关系标签目录下，第一个空格为关系类型，后两个空格为实体类别，点击空格后在下拉框中选择相应的实体类型，即第二个空格为实体1，第三个空格为实体2。需要注意的是，在关系类型的创建中，与实体类型相组合可以明确实体关系的方向，如【因果关系】中，如果实体1为原因，实体2为结果，则关系方向为【因果关系，实体1，实体2】，反之则为【因果关系，实体2，实体1】。



在创建实体类别与关系类别的时候，需注意以下几点：

- 创建的实体类别标签不能相同，同理创建的关系标签也不能相同。

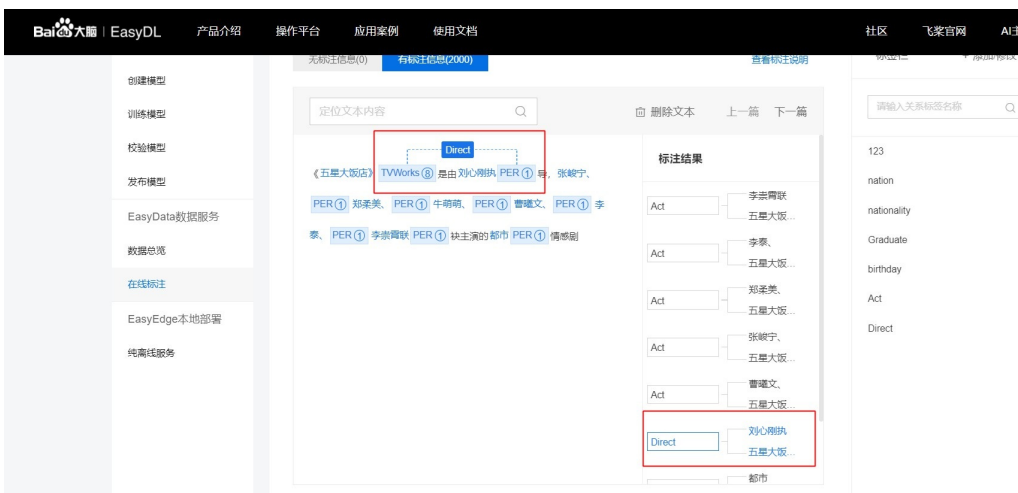
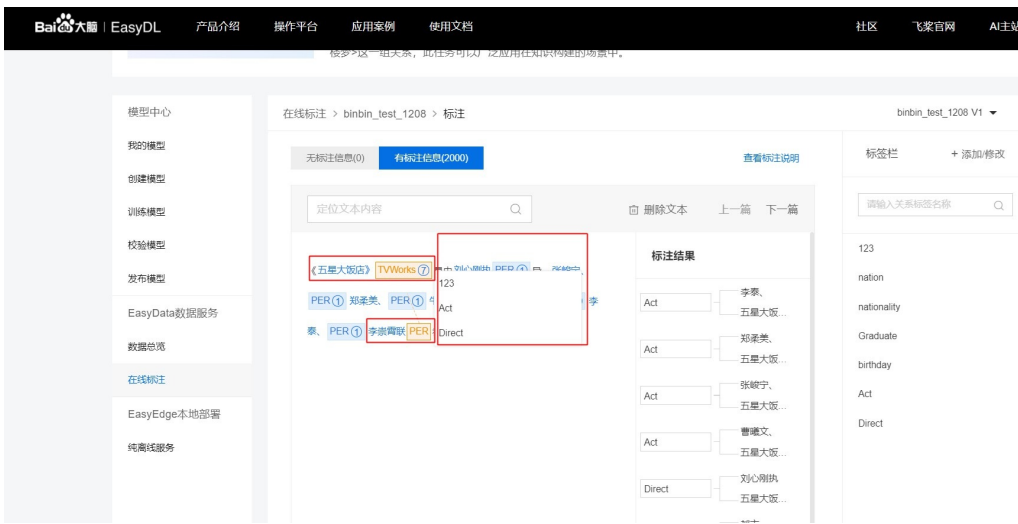
- 每个实体类别名称不超过10个字，每个实体关系名称也得在10个字以内。

**选择实体，标注实体类别和关系** 在上述创建完实体类别，设置实体关系后，针对上传的未标注数据，对数据中每个样本进行实体标注以及关系标注。**实体标注** 对每个样本实体标注时，点击鼠标左键，覆盖您想要标注的实体字符，在弹出的选项框中选中您所要的实体关系类别，在一个文本中，可以存在多个实体。采样以上方法，根据您的具体样本，可进行详细标注。



### 关系标注

当您完成一个文本的实体类别标注后，您可以对实体间的关系类型进行标注。选中实体1，点击鼠标左键，移动鼠标到实体2，鼠标点击左键则两个实体之间建立联系，此时在弹出的实体关系标签中选中所需标签完成关系标注。点击右键此时产生一条曲线，连接实体1和实体2。当标注完成后，可以看到标注结果栏产生刚刚标注完成的实体关系。任意点击标注栏的实体关系，可在文本内容上看到对应的关系。如您想删除已标注的关系，可在标注结果部分选中关系点击删除即可。



根据您的上传的数据样本，每个样本中的实体数与关系数可以不同。

### 数据去重

**重复样本的定义** 一个样本包括文本内容和实体关系类型。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定

为两个样本是重复样本。例如：

文本内容	实体关系1
今年年初，党中央、国务院根据国内外经济形势的变化，及时作出扩大内需、保持经济持续快速增长的重大决策。	{{[5,7],ORG},{[9,11],ORG},lead
今年年初，党中央、国务院根据国内外经济形势的变化，及时作出扩大内需、保持经济持续快速增长的重大决策。	{{[5,7],ORG}, {[9,11],ORG},friends
今年年初，党中央、国务院根据国内外经济形势的变化，及时作出扩大内需、保持经济持续快速增长的重大决策。	{{[5,7],ORG}, {[9,11],ORG},friends

上表三个样本均为重复样本，前两个样本虽然实体关系不同，但文本内容一致，为重复样本，后两个样本的文本内容与实体关系都一致，则也为重复样本。根据文本出现的顺序，最后一次重复样本将代替之前的重复样本。

小Tips：“如何利用好重复样本”如果您的数据存在样本种类不均衡的现象，您可以通过将重复样本数量小的那一类，使其样本数量增加到与数据量大的一类样本数量相近，以提高模型训练的效果，这种方法也称为“上采样”。

**平台去重策略** 平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。注意：当您确定了数据集为去重或非去重的属性后，便不可修改。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

- 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
- 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
- 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

## 模型训练

### 创建模型

#### 步骤 Step 1 创建模型

在【模型中心】或者【模型中心-我的模型】点击创建模型。

**Step 2 填写基本信息** 选择模型类型、提交模型名称、模型描述、联系方式即可创建模型。

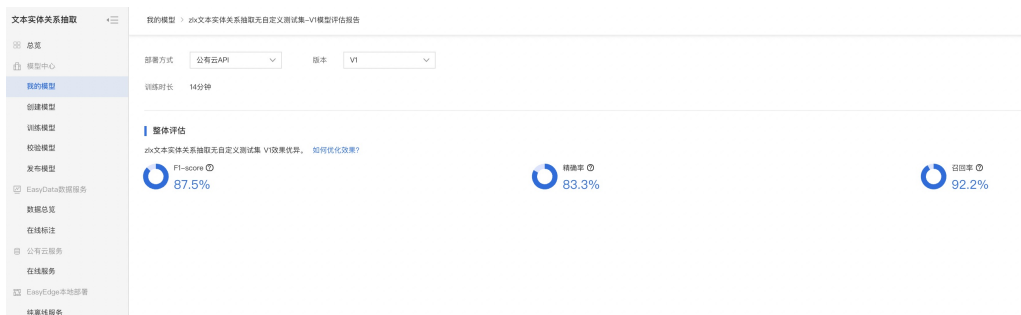
**Step 3 查看已创建的模型** 模型创建成功后，可以在【我的模型】中看到刚刚创建的模型。

- 1.创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
- 2.目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。
- 3.如果您是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

### 评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果。

「完整评估结果」页面中将记录整体评估的报告，包括该模型整体的F1-score、精确率、召回率。可以切换查看训练集与自定义测试集的效果评估报告。



整体评估中，各指标的释义如下：

- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值

### \*\*模型校验\*\*

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入文本，点击「校验」后，右侧识别结果栏将输出预测结果，您就可以看见识别出的实体关系以及实体关系数
  - 校验数据支持两种输入方式，直接输入文本或上传txt格式文本，文本长度上限为512汉字
4. 若您对预测结果满意，可点击「申请上线」，进行模型的发布

## 🔗 训练模型

### \*\*训练模型\*\*

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：



**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

### 部署方式

可选择「公有云部署」、「EasyEdge本地部署」。

#### 如何选择部署方式

### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备

### 选择算法

提供「高精度」算法。

- 1000条标注数据，在P40机器上预计在10分钟左右完成训练

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

### 模型筛选指标

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可（兼顾Precision精确度和Recall召回率）。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中沒有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

### Step 3 添加数据

#### 添加训练数据

- 需选择1个数据集。建议您使用的每个实体关系的样本数应达到1000个以上，再启动训练，如果您提交的训练数据中，每个实体关系的样本数不足1000，可能会影响模型的训练效果
- 训练时间与数据量大小和您选择的模型类型有关

#### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

##### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

### Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关 (<https://ai.baidu.com/ai-doc/EASYDL/pki8mrx4m>)
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

## 模型部署

### 🔗 整体介绍

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。公有云API

- 模型训练完毕后，为了方便企业用户一站式完成AI模型应用，文本实体关系抽取模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。

- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高、在线离线均有调用需求的企业场景。**相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

## 公有云API

### 发布API

#### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

**发布完成** 申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。

## 调用API

### 接口描述

基于自定义训练出的文本实体关系抽取模型，实现个性化实体关系识别。模型训练完毕后发布可获得定制化实体关系抽取API。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明 请求示例

HTTP 方法：POST

请求URL：请首先在定制化训练平台进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>"
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个字符最大长度512个字符

注意：通过API接口预测时，模型仅接收512个字符（包括汉字和标点符号）内的文本，超出将被截断。

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
content	是	string	预测的文本内容（最多仅支持512个字符，包括汉字、字母、符号等）
entities	是	array	此为实体数组，可包含多个实体
+alias	是	string	每个实体的识别信息，建议按顺序1、2、3、...依次梳理标注的实体值，其中1代表第一个实体，2代表第二个实体，依次类推
+offset	是	array	实体所在的位置
+span	是	string	实体内容
+tag	是	string	实体的类别
relations	是	array	实体关系的数据，可以包括多个实体关系
+alias	是	string	每个实体关系识别信息，建议按顺序1、2、3、...依次梳理标注的实体关系值，其中1代表第一个实体关系，2代表第一个实体关系，依次类推
+predicate	是	string	实体关系类型
+from	是	string	实体关系的主体实体
+to	是	string	实体关系的目标实体

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。



错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 调用API

### 接口描述

基于自定义训练出的文本实体关系抽取模型，实现个性化实体关系识别。模型训练完毕后发布可获得定制化实体关系抽取API。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明 请求示例

HTTP 方法：POST

请求URL：请首先在定制化训练平台进行自定义模型训练，完成训练后可在服务列表中查看并获取url。



参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>"
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个字符

注意：通过API接口预测时，模型仅接收512个字符（包括汉字和标点符号）内的文本，超出将被截断。

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_message	否	string	错误描述信息，当请求错误时返回
content	是	string	预测的文本内容（最多仅支持512个字符，包括汉字、字母、符号等）
entities	是	array	此为实体数组，可包含多个实体
+alias	是	string	每个实体的识别信息，建议按顺序1、2、3、...依次梳理标注的实体值，其中1代表第一个实体，2代表第二个实体，依次类推
+offset	是	array	实体所在的位置
+span	是	string	实体内容
+tag	是	string	实体的类别
relations	是	array	实体关系的数据，可以包括多个实体关系
+alias	是	string	每个实体关系识别信息，建议按顺序1、2、3、...依次梳理标注的实体关系值，其中1代表第一个实体关系，2代表第一个实体关系，依次类推
+predicate	是	string	实体关系类型
+from	是	string	实体关系的主体实体
+to	是	string	实体关系的目标实体

#### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 发布API

在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

## 私有API介绍

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面

发布模型页面示意：

2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取
4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

## 调用API

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)

- 进入[EasyDL社区交流](#) ,与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

## 部署包使用说明 部署方法

EasyDL定制化文本实体关系抽取模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#) 使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

使用方法: 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

## 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

## API参考

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/EntityRelationExtraction](http://{IP}:{PORT}/{DEPLOY_NAME}/EntityRelationExtraction)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```

{
  "text": "<UTF-8编码数据>"
}

```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）

### 返回说明

## 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回
error_msg	否	string	错误描述信息, 当请求错误时返回
content	是	string	预测的文本内容 (最多仅支持512个字符, 包括汉字、字母、符号等)
entities	是	array	此为实体数组, 可包含多个实体
+alias	是	string	每个实体的识别信息, 建议按顺序1、2、3、...依次梳理标注的实体值, 其中1代表第一个实体, 2代表第二个实体, 依次类推
+offset	是	array	实体所在的位置
+span	是	string	实体内容
+tag	是	string	实体的类
relations	是	array	实体关系的数据, 可以包括多个实体关系
+alias	是	string	每个实体关系识别信息, 建议按顺序1、2、3、...依次梳理标注的实体关系值, 其中1代表第一个实体关系, 2代表第一个实体关系, 依次类推
+predicate	是	string	实体关系类型
+from	是	string	实体关系的主体实体
+to	是	string	实体关系的目标实体

## 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如缺少必要出入参时返回:

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面，找到您的服务器API发布记录，点击【更新版本】，选择「更新包」或「完整包」来发布。两者区别：

包类型	描述
更新包	仅包含最新的模型应用，需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务，需执行download.sh脚本下载所需完整依赖文件

2、（CPU模型可忽略）如果您训练的模型为GPU版本，系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令，根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录（建议标记对应模型的版本号，便于区分不同模型版本），如`easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}` :申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：${DEPLOY_NAME}，前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务：${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}
**(可选操作) 更新证书**
python2 install.py lu

```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
** (可选操作) 进入V1版本部署包所在目录执行license更新操作,假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 文本实体抽取

### 整体介绍

#### 简介

定制文本实体抽取模型，实现对文本进行内容抽取，并识别为自定义的实体类别。平台提供文本实体标注的工具，您可在平台上传文档，完成标注后可直接进行模型训练。

更多详情访问：[EasyDL自然语言处理方向](#)

#### 应用场景

对内容中的关键实体进行识别和抽取，如：

- 金融研报信息识别
- 法律案件文书实体抽取
- 医疗病例实体抽取
- 其他：尽情脑洞大开，训练你希望实现的文本实体抽取的模型

#### 技术特色

文本实体抽取模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

#### 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

#### API上传

本文档主要说明当您线下已有大量的已经完成整理的文本数据，如何通过调用API完成文本数据的便捷上传和管理。

EasyDL数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一

致。

## 数据集创建API

### 接口描述

该接口可用于创建数据集。

### 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：`https://aip.baidubce.com/rpc/2.0/easydl/dataset/create`

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

## 查看数据集列表API

### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明



### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

### 查看实体类别列表API

### 接口描述

该接口可用于查看数据集的实体类别。返回实体类别的名称、包含数据量等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	实体类别总数
results	否	array(object)	实体类别列表
+label_id	否	string	实体类别ID
+label_name	否	string	实体类别名称
+entity_count	否	number	样本数量

添加数据API

接口描述

该接口可用于在指定数据集添加数据。

接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

请求说明

请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
dataset_id	是	number	数据集ID
appendLabel	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为TEXT_ENTITY_EXTRACTION时，填入utf-8编码的文本。文本实体抽取限制512个字符（包括汉字、数字、字母）
entity_name	是	string	文件名
labels	是	array(object)	实体类别数据
+label_name	是	string	实体类别名称（由数字、字母、中划线、下划线组成），长度限制20B
+offset	是	array	文本实体抽取任务需要给出，是抽取的具体实体内容的位置，从entity_content中，第一个字符记为0起算，以数组"[n,m]"的形式填入

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

#### 接口描述

该接口可用于删除数据集。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
dataset_id	是	number	数据集ID

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 实体类别删除API

#### 接口描述

该接口可用于删除实体类别。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_ENTITY_EXTRACTION
dataset_id	是	number	数据集ID
label_name	是	string	实体类别名称

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。

- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

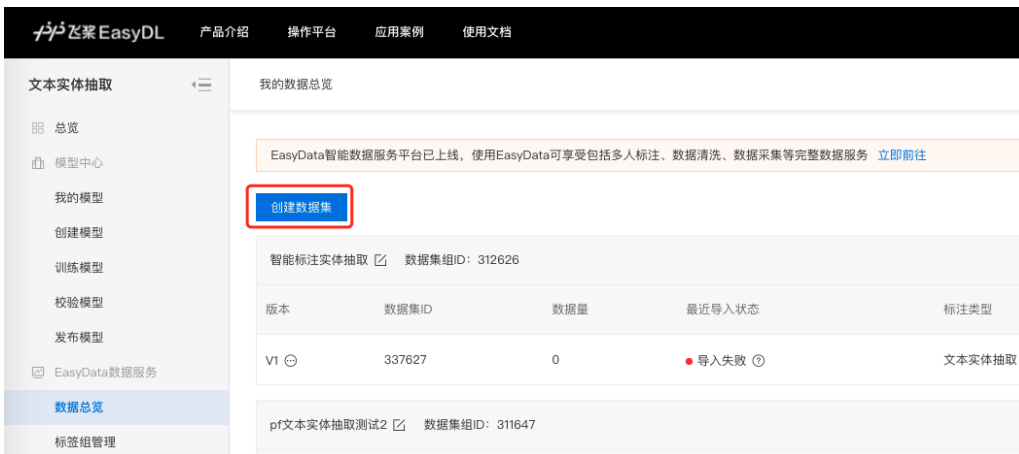
需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类/实体类别不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 🔗 创建数据集并导入

### 创建数据集

在训练模型之前，需要在【数据总览】里面“创建数据集”。需输入数据集名称、选择相应的标注模版、选择数据去重策略，即可创建一个空数据集。



**数据自动去重**即平台对您上传的数据进行重复样本的去重。建议创建数据集时选择「数据自动去重」

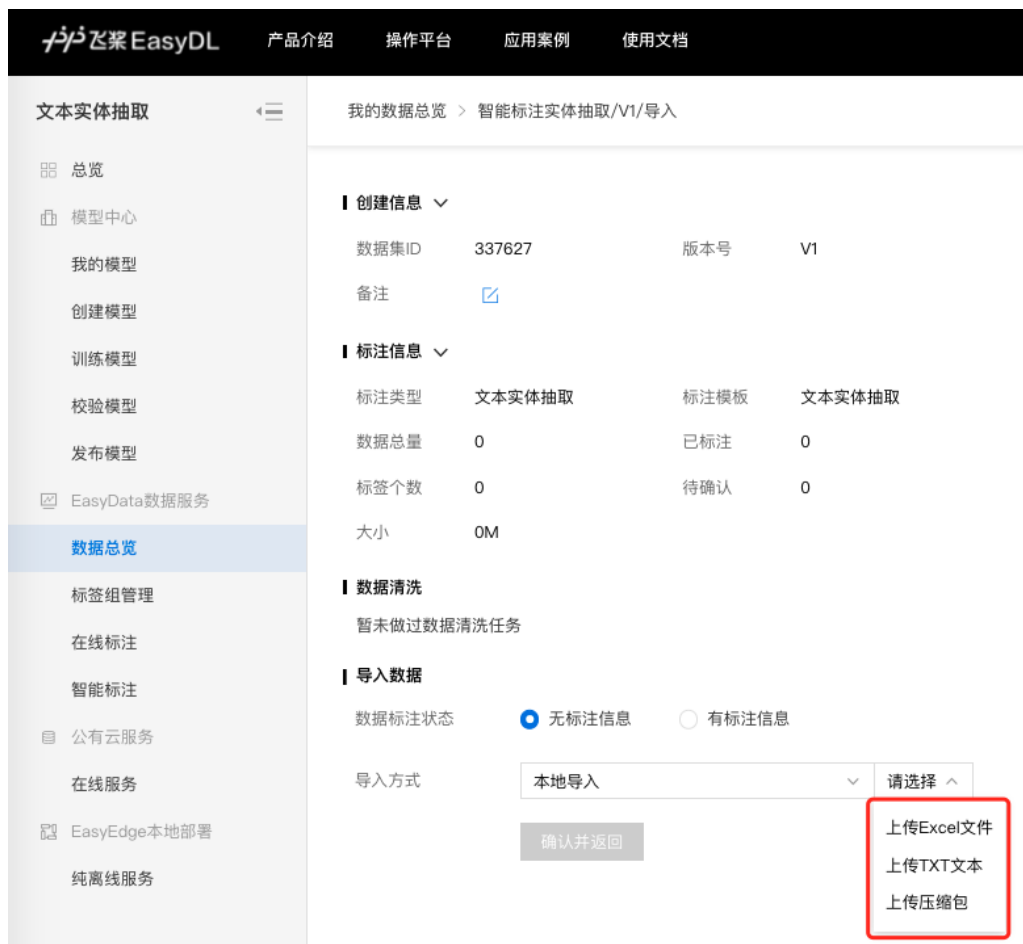
**导入无标注数据** 创建数据集后, 在「数据总览」页面中, 找到该数据集, 点击右侧操作列下的「导入」, 即可进入导入数据页面。



您可以使用3种方案上传数据, 分别为：

- 本地导入
- BOS目录导入
- 分享链接导入

**本地导入** 您可以通过以下三种方式进行本地数据的导入：



- 以压缩包的方式上传
- 以TXT文本文件方式上传
- 以Excel文件的方式上传

#### 通过压缩包上传时，需注意：

- 压缩包内的一个文本文件将作为一个样本上传。压缩包格式为.zip格式，压缩包内文件类型支持txt，编码仅支持UTF-8
- 每组数据的数建议不超过512个字符，超出将被截断

#### 通过TXT文本上传时，需注意：

- 文本文件内数据格式要求为"文本内容\n"（即每行一个样本，使用回车换行），每一行表示一组数据，每组数据的数建议不超过512个字符，超出将被截断
- 文本文件类型支持txt，编码仅支持UTF-8，单次上传限制100个文本文件，最多可上传100万个文件

#### 通过Excel文件上传时，需注意：

- Excel文件内首行为表头，每行为一个样本，每个样本字符数不得超过512个字符，超出将被截断
- 文件格式支持xlsx格式，单次可上传100个文件

#### BOS目录导入 需选择Bucket地址与对应的文件夹地址。

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入。

#### 分享链接导入 需输入链接地址。分享链接导入的要求如下：

- 仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

**导入有标注数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面。

您可以使用本地上传的方案上传数据。您可以通过以下三种方式进行本地数据的导入：

- 以Excel文件的方式上传

- 以API的形式导入

通过Excel文件上传时，需注意：

- Excel文件内首行为表头，每行为一个样本，每个样本字符数不得超过512个字符，超出将被截断
- 文件格式支持xlsx格式，单次可上传100个文件

通过API上传时，需注意：

- 可参考以下文档：[实体抽取API数据管理](#)

### 什么是实体类别？

实体类别 (Entity Type) 是指某类事物的集合，每一类数据对象的个体称为实体，如人/角色 (例如学生)，对象 (例如发票)，概念 (例如简介) 或事件 (例如交易)。实体类别名称由数字、中英文、中/下划线组成，长度上限256字符

## 在线标注

通过平台导入「无标注信息」的数据集后，可对无标注数据进行标注操作。

### 创建标签

进入到待标注的数据集，您需要在右侧的标签栏中创建标签，点击「添加/搜索标签」后，即可输入标签名称，注意平台仅支持数字和字母的标签名

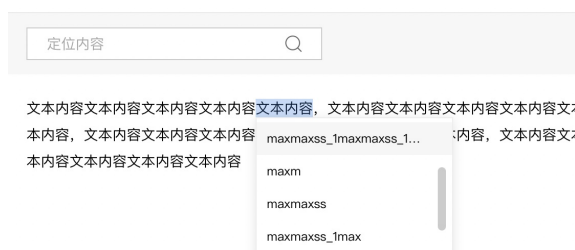


添加完标签后，可以添加标签的备注信息，如下图：



### 标注实体

您可以在文中划选需要标注的文本，然后在弹出的下落标签中选择需要标注的标签，如下图。也可以在划选文本后，在右侧的标签区域点击选择标签

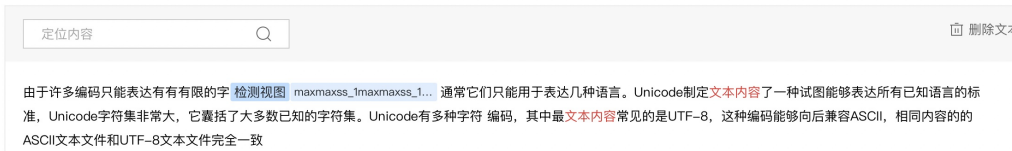


您也可以通过点击某个标签的「连续标注」功能 (如图中红框标记)，在文中通过连续划选文本内容来进行连续标注。如下图：





在您需要进行在文中进行关键词检索时，可以在文本上方的「定位内容」区域，输入文本内容，便可在文本区域高亮出检索的关键词，以方便您定位内容和标注。



**\*\*查看标注信息\*\***

您可以在标签中，点击「高亮」图标按钮，来达到让文本区域高亮显示标注信息，以方便您的查看标注情况。图标按钮如下图所示：



🔗 数据去重

**重复样本的定义**

一个文本实体抽取的样本包括文本内容和实体类别。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。例如：

以下3条都是文本实体抽取任务的重复样本，样本示例如下：

文本内容	实体类别
今天北京的空气不错	北京：loc
今天北京的空气不错	今天：date
今天北京的空气不错	北京：local

上述两个表中，都代表三个样本均为重复样本，后两个样本虽然标签不一，但文本内容一致，也为重复样本。

Tips：“如何利用好重复样本”，如果您在模型训练过程中，需要通过增加某个类别标签的预测权重，可以通过增加此标签的重复样本来达到此目标。

**平台去重策略**

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。注意：当您确定了数据集为去重或非去重的属性后，便不可修改。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

1. 数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖
2. 数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖
3. 数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

## 模型训练

### 创建模型

#### 步骤 Step 1 创建模型

在【模型中心】或者【模型中心-我的模型】点击创建模型。

Step 2 填写基本信息 选择模型类型、提交模型名称、模型描述、联系方式即可创建模型。

Step 3 查看已创建的模型 模型创建成功后，可以在【我的模型】中看到刚刚创建的模型。

- 1.创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
- 2.目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。
- 3.如果您是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

### 评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果。

「完整评估结果」页面中将记录整体评估的报告，包括该模型整体的F1-score、精确率、召回率。可以切换查看训练集与自定义测试集的效果评估报告。



整体评估中，各指标的释义如下：

- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值

如果单个文本实体类别的文本量在100条以内，评估指标的计算将出现误差，建议每个文本实体类别提供超过1000个实体样本

#### **\*\*模型校验\*\***

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入文本，点击「校验」后，右侧识别结果栏将输出预测结果
  - 校验数据支持两种输入方式，直接输入文本或上传txt格式文本，文本长度上限为512汉字
4. 若您对预测结果满意，可点击「申请上线」，进行模型的发布

如果单个文本实体类别的文本量在100条以内，评估指标的计算将出现误差，建议每个文本实体类别提供超过1000个实体样本

### 效果优化

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

### \*\*检查并优化训练数据\*\*

- 检查是否存在训练数据过少的情况，建议每个文本实体类别提供超过1000个实体，如果低于这个量级建议扩充
- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

### \*\*选择高精度模型\*\*

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

## 🔗 训练模型

### \*\*训练模型\*\*

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：

**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

### 部署方式

可选择「公有云部署」、「EasyEdge本地部署」。

### 如何选择部署方式

### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备

### 选择算法

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率将更高。如果您手中的标注数据集样本较少（例如少于1000条），可选择「高精度」的算法。使用高精度的算法训练模型将会耗时更久，实验环境下1000个样本，预计在20分钟左右完成训练
- 如果您选择了高性能的模型，相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在15min内完成训练。同样的数据量情况下，效果比高精度的模型4-5%

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

### 模型筛选指标

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可（兼顾Precision精确度和Recall召回率）。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中沒有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

### Step 3 添加数据

#### 添加训练数据

- 可选择多个数据集。注意，文本实体抽取模型至少需要有1个及以上实体类别，样本数建议超过1000
- 训练时间与数据量大小和您选择的模型类型有关

#### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

### Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

## 模型部署

### 🔗 整体介绍

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。 **公有云API**

- 模型训练完毕后，为了方便企业用户一站式完成AI模型应用，文本实体抽取模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高、在线离线均有调用需求的企业场景。 **相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

### 🔗 公有云API

#### 🔗 发布API

##### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

**发布完成** 申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。

### 🔗 调用API

#### 接口描述

基于自定义训练出的文本实体抽取模型，实现个性化文本实体识别。模型训练完毕后发布可获得定制化实体抽取API 详情访问：[定制化训练和服务平台](#)进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

#### 请求说明

#### 请求示例

HTTP 方法：`POST`

请求URL：请首先在[定制化训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>"
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个字符

注意：通过API接口预测时，模型仅接收512个字符（包括汉字和标点符号）内的文本，超出将被截断。

#### 请求示例代码

Python3

```
"""
EasyDL 文本实体抽取 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标文本的 本地文件路径，UTF-8编码，最大长度512个汉字，超出将被截断**
TEXT_FILEPATH = "【您的测试文本数据地址，例如：./example.txt】"
```

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
content	是	string	预测的文本内容（最多仅支持512个字）
results	否	array(object)	分类结果数组
+span	否	string	抽取的具体实体内容
+offset	否	array(number)	数组由两个元素组成，分别是实体的起始位置和终止位置，从entity_content中，第一个字符记为0起算
+tag	否	string	识别对应span的实体类别

#### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code** : 错误码。
- **error\_msg** : 错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

☁ 离线API

☁ 发布API



在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

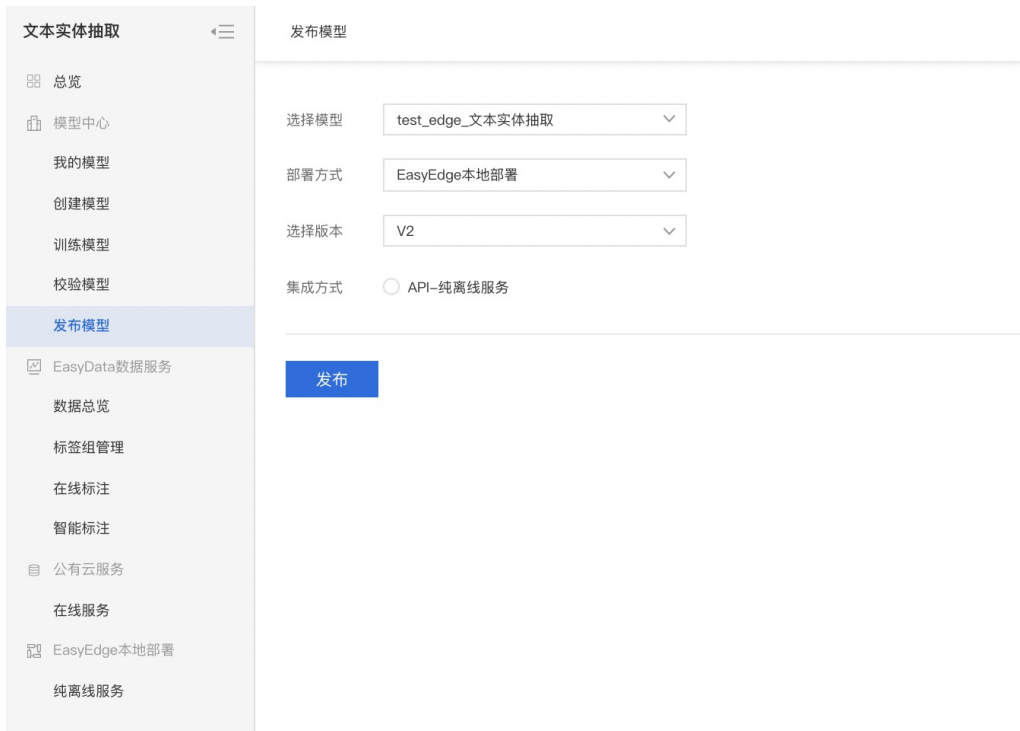
### 私有API介绍

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面

**发布模型页面示意：**



2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取
4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

### 调用API

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管



## 部署包使用说明

### 部署方法

EasyDL定制化文本分类模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#) 使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

### 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络联通性测试、容器关键报错日志输出等

使用方法: 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

### 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

### API参考

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL自然语言处理方向](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/EntityExtraction](http://{IP}:{PORT}/{DEPLOY_NAME}/EntityExtraction)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```

{
  "text": "<UTF-8编码数据>"
}

```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回
error_msg	否	string	错误描述信息, 当请求错误时返回
content	是	string	预测的文本内容 (最多仅支持512个字)
results	否	array(object)	分类结果数组
+span	否	string	抽取的具体实体内容
+offset	否	array(number)	数组由两个元素组成, 分别是实体的起始位置和终止位置, 从entity_content中, 第一个字符记为0起算
+tag	否	string	识别对应span的实体类别

### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如缺少必要出入参时返回:

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (868826008) 或工单联系技术支持团队
336001	Invalid Argument	入参格式有误, 比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误, 比如缺少必要参数代码格式是否有误。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336003	Base64解码失败	文本格式有误或base64编码有误, 请根据接口文档检查格式, base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法, 目前支持文本文件类型为支持txt, 文本文件大小限制长度最大512 UTF-8字符。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336005	解码失败	文本编码错误 (不是utf-8), 目前支持文本文件类型为支持txt。如果遇到请重试, 如反复失败, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求, 可恢复正常, 若反复重试依然报错或有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面, 找到您的服务器API发布记录, 点击【更新版本】, 选择「更新包」或「完整包」来发布。两者区别:

包类型	描述
更新包	仅包含最新的模型应用, 需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务, 需执行download.sh脚本下载所需完整依赖文件

2、(CPU模型可忽略) 如果您训练的模型为GPU版本, 系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令, 根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录 (建议标记对应模型的版本号, 便于区分不同模型版本), 如 `easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}`: 申请时填写的服务名称

以如下场景举例说明：模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后，进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务：${DEPLOY_NAME}，前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务：${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}
**（可选操作）更新证书**
python2 install.py lu

```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
**（可选操作）进入V1版本部署包所在目录执行license更新操作，假如部署包在/opt目录下，以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 短文本相似度

### 整体介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

定制短文本相似度模型，是基于深度学习技术，可实现对两个文本进行相似度的比较计算。

更多详情访问：[EasyDL自然语言处理方向](#)

#### 应用场景

1. 搜索场景下的搜索信息匹配
2. 新闻媒体场景下的新闻推荐
3. 新闻媒体场景下的标题去重
4. 客户场景下的问题匹配

#### 技术特色

短文本相似度模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等

方面的新知识，实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选底座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

### 上传数据集

您可以上传带有标注信息的数据，和无标注信息的数据。您可以根据自己的情况，选择上传方式，目前平台提供上传方式有：

- 上传Excel文件
- 上传TXT文本
- 上传压缩包
- 通过API导入

下面分别为您介绍几种上传方式

#### 以Excel文件上传

- Excel文件内数据格式要求为：每行是一个样本，使用第一列和第二列分别作为需要计算相似度的两个文本，第三列为相似度标签（如果导入无标注数据，此列无数据）。第一列和第二列的文本内容的字符数建议不超过512个，超出将被截断。
- 文件类型支持xlsx格式，单次上传限制100个文件
- 请确保您上传的样本在sheet1中，且数据都在首列。注意，首行作为表头将被系统忽略

#### 以压缩包方式上传

- 压缩包格式为.zip格式，单个压缩包限制5G以内
- 压缩包内文本文件类型为txt，每个txt每行数据格式要求为“文本内容1\t文本内容2\t标注结果\n”，标注结果仅用1/0表示，1代表相似，0代表不相似。一行表示一组数据，每个文本可以有多个短文本组数据，每组数据字符数建议不超过1024个字符（约512个汉字）

#### 以TXT文本文件上传

- 支持文本文件类型为txt，编码仅支持UTF-8，单次上传限制100个文本文件。
- 短文本相似度的数据格式要求为“文本内容1\t文本内容2\t标注结果\n”，一行表示一组数据，每组数据字符数建议不超过1024个字符（约512个汉字），可上传多个文本文件

#### 通过API方式导入

您可以通过API导入文档，查看上传数据的方式

### API上传

本文档主要说明当您线下已有大量的已经完成分类整理的文本数据，如何通过调用API完成文本数据的便捷上传和管理。

EasyDL文本数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

#### 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 数据集创建API

#### 接口描述

该接口可用于创建数据集。

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/create>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，TEXT_MATCHING 短文本相似度
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

### 查看数据集列表API

#### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，短文本相似度为TEXT_MATCHING
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

查看分类（标签）列表API

接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。短文本相似度标签仅存在0和1两个标签，其中0代表不相似，1代表相似。

请求说明

请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/label/list

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

## 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，短文本相似度为TEXT_MATCHING
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认2，最多2

## 返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

## 添加数据API

## 接口描述

该接口可用于在指定数据集添加数据。

## 请求说明

## 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

## 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，短文本相似度为TEXT_MATCHING
dataset_id	是	number	数据集ID
appendLabel	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为TEXT_MATCHING时，填入utf-8编码的文本。每行文本内容的格式为“文本内容1\t文本内容2”。 内容限制为：文本相似度中每个文本不得超过512个字（包括标点）
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称：仅可为0或1，其中0代表不相似，1代表相似

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 数据集删除API

#### 接口描述

该接口可用于删除数据集。

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，短文本相似度为TEXT_MATCHING
dataset_id	是	number	数据集ID

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 分类（标签）删除API

#### 接口描述

该接口可用于删除分类（标签），短文本相似度不支持删除标签。

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```



需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用, 请再次请求, 如果持续出现此类错误, 请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在, 请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数, 请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法, 请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 模型训练

### 🔗 创建模型

#### 步骤 Step 1 创建模型

在【模型中心】或者【模型中心-我的模型】点击创建模型。

**Step 2 填写基本信息** 选择模型类型、提交模型名称、模型描述、联系方式即可创建模型。

**Step 3 查看已创建的模型** 模型创建成功后, 可以在【我的模型】中看到刚刚创建的模型。

短文本相似度模型

模型列表 > 创建模型

模型类别 短文本相似度

模型名称 \*

模型归属  公司  个人

请输入公司名称

所属行业 \* 请选择行业

应用场景 \* 请选择应用场景

邮箱地址 \* z\*\*\*\*\*@baidu.com

联系方式 \* 135\*\*\*\*\*919

功能描述 \*

0/500

完成

- 1.创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
- 2.目前单个用户在每种类型的模型下最多可创建**10**个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。
- 3.如果您是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

## 效果优化

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

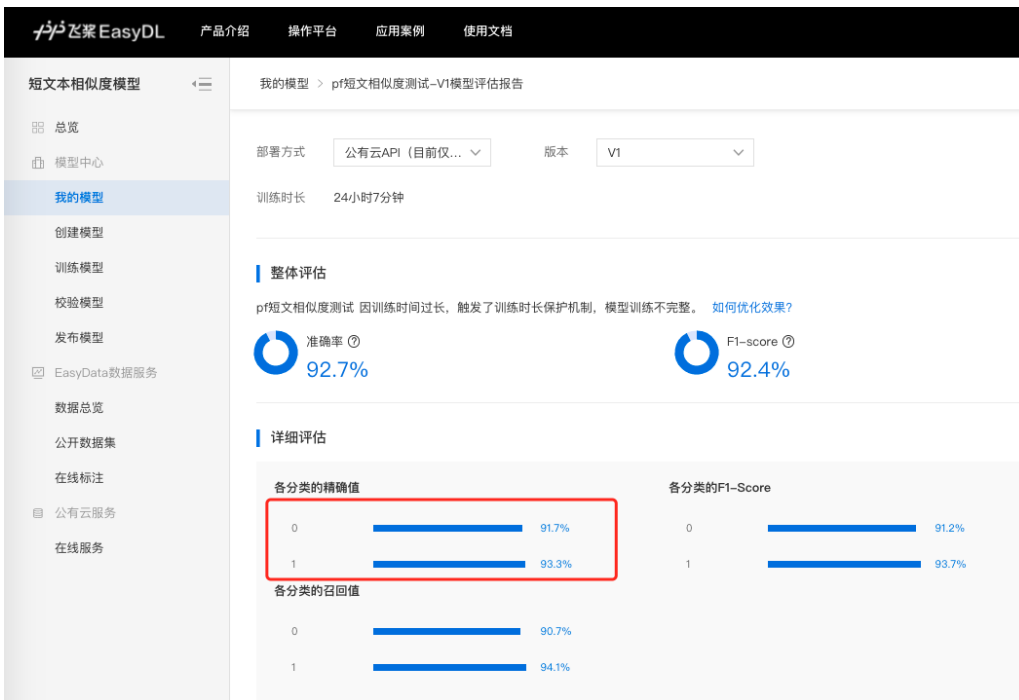
一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

### **\*\*检查并优化训练数据\*\***

- 检查是否存在训练数据过少的情况，建议文本数量不少于1000个，如果低于这个量级建议扩充
- 通过模型效果评估报告中的详细评估指标，有针对性地扩充训练数据。例如下图评估报告中显示“0”（不相似）的精确值较低，可考虑从两方面进行优化，一是适当增加“0”不相似数据集样本量，而是检查当前数据集中“0”不相似数据集质量，是否有定义模糊的情况等，以此达到模型优化的效果。



- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

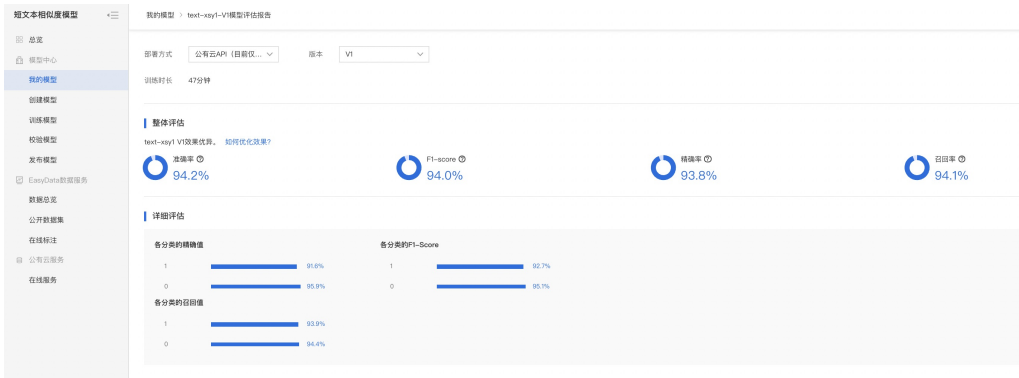
### \*\*选择高精度模型\*\*

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

## 评估效果

**\*\*模型评估\*\*** 模型训练完成后，可以在「我的模型」列表中查看该模型的效果，以及完整评估结果。



「完整评估结果」页面中将记录整体评估与详细评估的报告，包括该模型整体的准确率、F1-score、精确率、召回率，以及评估样本具体数据情况，各分类的精确值、F1-Score等指标。

整体评估中，各指标的释义如下：

- 准确率：正确分类的样本数与总样本数之比
- F1-score：给每个类别相同的权重，计算每个类别的F1-score，然后求平均值
- 精确率：给每个类别相同的权重，计算每个类别的精确率，然后求平均值
- 召回率：给每个类别相同的权重，计算每个类别的召回率，然后求平均值


如果在训练阶段，使用的数据集中，相似或不相似的文本量在100条以内，训练出来的模型的效果评估报告的参考价值较小，建议您训练时数据量准备充足

**\*\*模型校验\*\***

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧输入两对文本，点击「校验」后，右侧识别结果栏将输出预测结果

如果在训练阶段，使用的数据集中，相似或不相似的文本量在100条以内，训练出来的模型的效果评估报告的参考价值较小，建议您训练时数据量准备充足

 发起训练**\*\*训练模型\*\***

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：

**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置****部署方式**

可选择「公有云部署」。

**选择算法**

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率将更高。如果您手中的标注数据集样本较少（例如少于1000条），可选择「高精度」的算法。使用高精度的算法训练模型将会耗时更久，实验环境下1000个样本，预计在20-60分钟左右完成训练
- 如果您选择了高性能的模型，相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在10min内完成训练。同样的数据量情况下，效果比高精度的模型4-5%

「高精度」算法内置[文心大模型](#)，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

**模型筛选指标**

选择不同的模型选择方式，对应的模型各项效果指标将有所不同。如果没有特殊场景的要求，使用默认即可（兼顾Precision精确度和Recall召回率）。有以下指标可供选择：

- 模型兼顾Precision和Recall：挑选模型时，兼顾Precision精确度和Recall召回率，如场景中并没有对精度或召回的特别要求，建议您使用此默认指标
- Precision最高的模型：挑选模型时，优先挑选Precision精度最高的模型作为部署模型
- Recall最多的模型：挑选模型时，优先选择召回率最高的模型作为部署模型
- ACC最大的模型：挑选模型时，优先挑选预测样本数最多的模型作为部署模型
- Loss最小的模型：挑选模型时，优先挑选预测偏差最小的模型

**Step 3 添加数据****添加训练数据**

- 可选择多个数据集
- 训练时间与数据量大小和您选择的模型类型有关，如果您选择的是高性能的模型，使用10000条训练样本将在10min内完成训练；如果您选择的是高精度的模型，使用10000条训练样本，将在20-60min完成训练

**添加自定义测试集**

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

### Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

### 模型部署

☞ 公有云API

☞ 发布API

#### 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

发布模型界面示意：

**短文本相似度模型** ◀

- 总览
- 模型中心
- 我的模型
- 创建模型
- 训练模型
- 校验模型
- 发布模型
- EasyData数据服务
- 数据总览
- 公开数据集
- 在线标注
- 公有云服务
- 在线服务

**发布模型**

选择模型:

部署方式:

选择版本:

服务名称 \*

接口地址 \*

其他要求:

**发布完成** 申请发布后, 通常的审核周期为T+1, 即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况, 请在百度云控制台内[提交工单](#)反馈。

## 调用API

本文档主要说明定制化模型发布后获得的API如何使用, 如有疑问可以通过以下方式联系我们:

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#), 与其他开发者进行互动
- 加入EasyDL官方QQ群 (群号:868826008) 联系群管

## 接口描述

基于自定义训练出的短文本匹配模型, 实现个性化短文本相似度计算。模型训练完毕后发布可获得定制API

## 接口鉴权

1、在[EasyDL——控制台](#)创建应用

2、应用列表页获取AK SK

应用名称	AppID	API Key	Secret Key	创建时间	操作
1	1482660602	AzS2nqX2DXGv0pxX1qZZE5Xm	***** 显示	2019-10-21 16:20:03	报表 管理 删除
2	1482657697	sif9YPOK6U4f4zIHddxwGxp	***** 显示	2019-08-23 15:17:15	报表 管理 删除
3	1482657696	WiyprCqjKHmKYW4uWXqfa20	***** 显示	2019-08-23 15:16:01	报表 管理 删除
4	1482657673	NdvOkPPI949FoiBrW6eOoDL1	***** 显示	2019-08-22 18:18:47	报表 管理 删除
5	1482657672	GkMah53SGTAUJLS2Tb7bRa0A	***** 显示	2019-08-22 18:18:32	报表 管理 删除
6	1482657671	scKo2V4AH1Iya5WjS4sD5tXS	***** 显示	2019-08-22 18:16:52	报表 管理 删除
7	1482657670	Gz3LQLGKAvl3DSTRxexMY1UL	***** 显示	2019-08-22 18:16:37	报表 管理 删除
8	1482657669	JNriZSjDsa5y20A12MslYX0B9	***** 显示	2019-08-22 18:16:06	报表 管理 删除
9	1482657668	KHbAdxabR9zr0GItwKK6IFLK	***** 显示	2019-08-22 18:15:51	报表 管理 删除

## 请求说明

## 请求示例

HTTP 方法: POST

请求URL：请首先进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，需以json方式请求。

Body请求示例：

```
{
  "text_a": "<UTF-8编码数据>",
  "text_b": "<UTF-8编码数据>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text_a	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）
text_b	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
score	是	number	文本相似度，值的范围[0,1]，相似度递增

在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本编码错误等等，可检查下文本编码、代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	文本格式有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336004	输入文本大小不合法	文本超出大小限制，每个文本限制512个字符（包括汉字、字符、数字或字母），有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	文本解码失败	文本编码错误，请检查并修改文本格式
336006	缺失必要参数	text字段缺失（未上传文本）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队

## 🔗 调用API

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

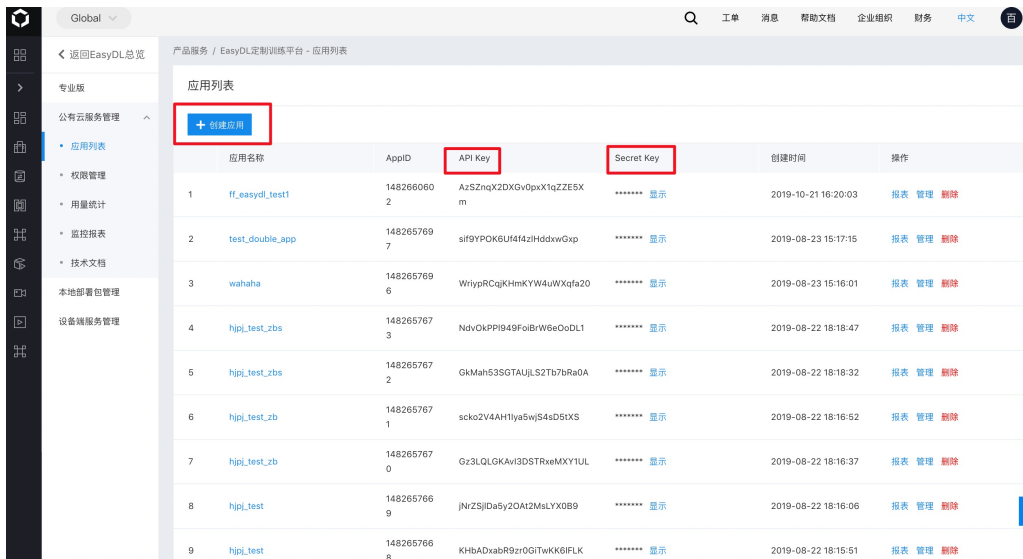
## 接口描述

基于自定义训练出的短文本匹配模型，实现个性化短文本相似度计算。模型训练完毕后发布可获得定制API



## 接口鉴权

- 1、在EasyDL——控制台创建应用
- 2、应用列表页获取AK SK



## 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，需以json方式请求。

Body请求示例：

```
{
  "text_a": "<UTF-8编码数据>",
  "text_b": "<UTF-8编码数据>"
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必填	类型	可选值范围	说明
text_a	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）
text_b	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）

## 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
score	是	number	文本相似度，值的范围[0,1]，相似度递增

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

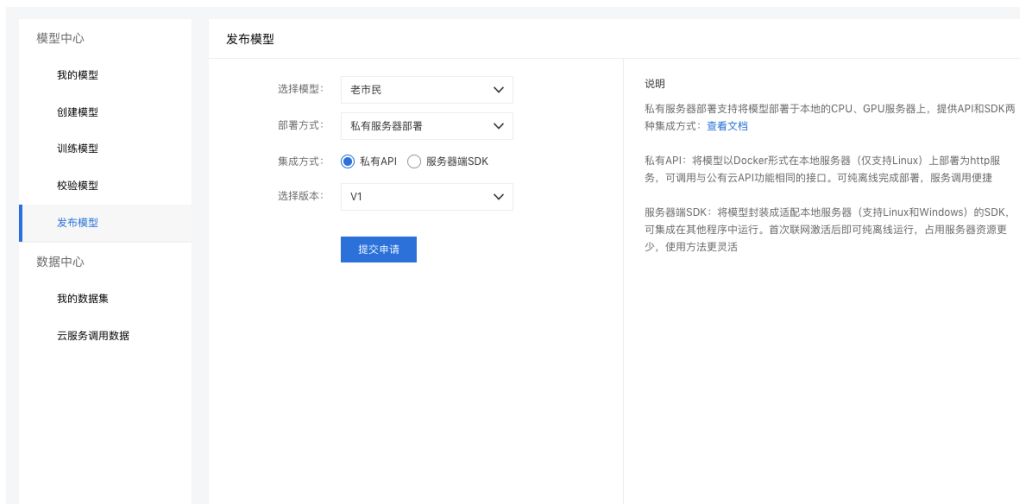
需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（868826008）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本编码错误等等，可检查下文本编码、代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	文本格式有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336004	输入本文大小不合法	文本超出大小限制，每个文本限制512个字符（包括汉字、字符、数字或字母），有疑问请通过QQ群（868826008）或工单联系技术支持团队
336005	文本解码失败	文本编码错误，请检查并修改文本格式
336006	缺失必要参数	text字段缺失（未上传文本）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队

🔒 发布私有部署服务器

🔒 发布私有部署服务器

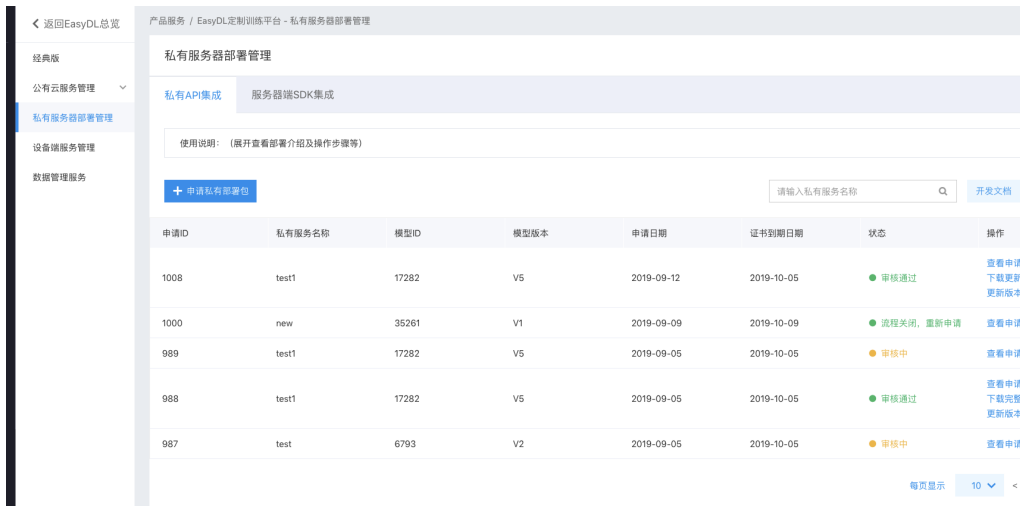
训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可将模型部署到私有服务器：



- 选择模型
- 选择部署方式「私有服务器部署」
- 选择集成方式「私有API」（当前仅支持此方式）

将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

点击「提交申请」后，前往控制台申请私有部署包。并[参考文档](#)完成集成



## 私有服务器部署价格说明

EasyDL经典版已支持将定制模型部署在私有服务器上，只需在发布模型时提交私有服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请微信搜索“BaiduEasyDL”添加小助手咨询，通过线下签订合同购买使用。

## 私有部署服务API调用说明文档

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往EasyDL进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

## 部署包使用说明

### 部署方法

EasyDL定制化短文本相似度模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#) 使用python2 版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

## 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

## API参考

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/TextMatching](http://{IP}:{PORT}/{DEPLOY_NAME}/TextMatching)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```

{
  "text_a": "<UTF-8编码数据>",
  "text_b": "<UTF-8编码数据>"
}

```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
text_a	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）
text_b	是	string	-	文本数据，UTF-8编码。最大长度512个字符（包括汉字、字母、数字和符号）

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
score	是	number	文本相似度, 从0-1, 相似度递增

### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如缺少必要出入参时返回:

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (868826008) 或工单联系技术支持团队
336001	Invalid Argument	入参格式有误, 比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误, 比如缺少必要参数代码格式是否有误。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误, 请根据接口文档检查格式, base64编码请求时注意要去掉头部。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336004	输入文件大小不合法	文本超出大小限制, 文本限4M以内, 请根据接口文档检查入参格式, 有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336005	文本解码失败	文本编码错误, 请检查并修改文本格式
336006	缺失必要参数	文本字段内容缺失
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求, 可恢复正常, 若反复重试依然报错或有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面, 找到您的服务器API发布记录, 点击【更新版本】, 选择「更新包」或「完整包」来发布。两者区别:

包类型	描述
更新包	仅包含最新的模型应用, 需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务, 需执行download.sh脚本下载所需完整依赖文件

2、(CPU模型可忽略) 如果您训练的模型为GPU版本, 系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令, 根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录 (建议标记对应模型的版本号, 便于区分不同模型版本), 如 `easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}`: 申请时填写的服务名称

以如下场景举例说明: 模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后,进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务: ${DEPLOY_NAME},前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务: ${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}
** (可选操作) 更新证书**
python2 install.py lu

```

**模型回滚** 以如下场景举例说明：模型版本从V2回滚至V1

方法一：

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
** (可选操作) 进入V1版本部署包所在目录执行license更新操作,假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录,参考上述【模型更新】步骤,执行模型升级操作(即先卸载v2,后升级为v1)

## 评论观点抽取

### 整体介绍

#### 简介

Hi,您好,欢迎使用百度EasyDL定制化训练和服务平台。

定制评论观点抽取的模型,是基于自建抽取体系的机器学习方法,可实现从评论文本中抽取评价片段、评价维度、评价观点、并判断评论的情感倾向。

更多详情访问：[EasyDL自然语言处理方向](#)

#### 应用场景

评论数据中隐藏着大量的商业价值：

- 新品用户分析：商家很难单纯从新产品消费额、出货量等结构化数据中找到是否满足消费者需求的答案,可通过新产品的用户评论,完成用户分析。
- 挖掘产品问题：挖掘出有价值的信息,比如产品的问题、用户的偏好、竞品的差异性,甚至洞察出新的商业机会。
- 辅助消费决策：通过对比同一类型产品不同商品或商家的评论观点信息,可以辅助用户进行消费决策。
- 互联网舆情分析：商家可通过对评论及其情感倾向的分析,监控品牌和商品的舆情信息变化。

## 🔗 技术特色

评论观点抽取模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

**文心大模型**是百度发布的产业级知识增强大模型，是千行百业AI开发的首选底座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 🔗 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

### 🔗 创建数据集并导入

#### 1. 创建数据集

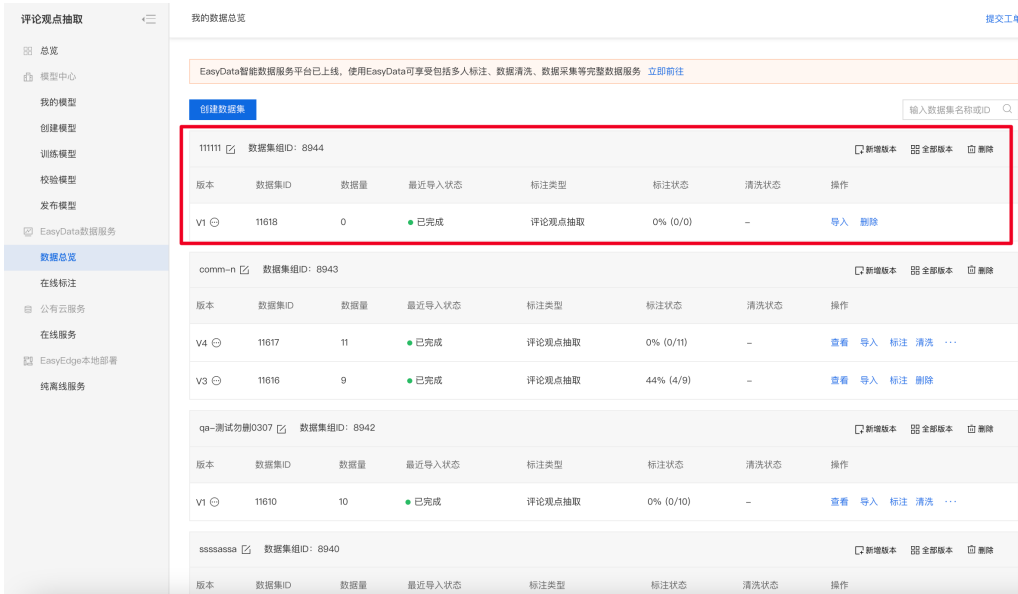
1. 选择【EasyDate数据服务】目录下数据总览，点击“创建数据集”。



3. 输入数据集名称，选择数据集属性：是否对数据进行去重操作，详细方法见数据去重策略。

4. 点击完成，在数据总览目录下可以看到生成一个空数据集项目。





## 2. 导入未标注文本数据

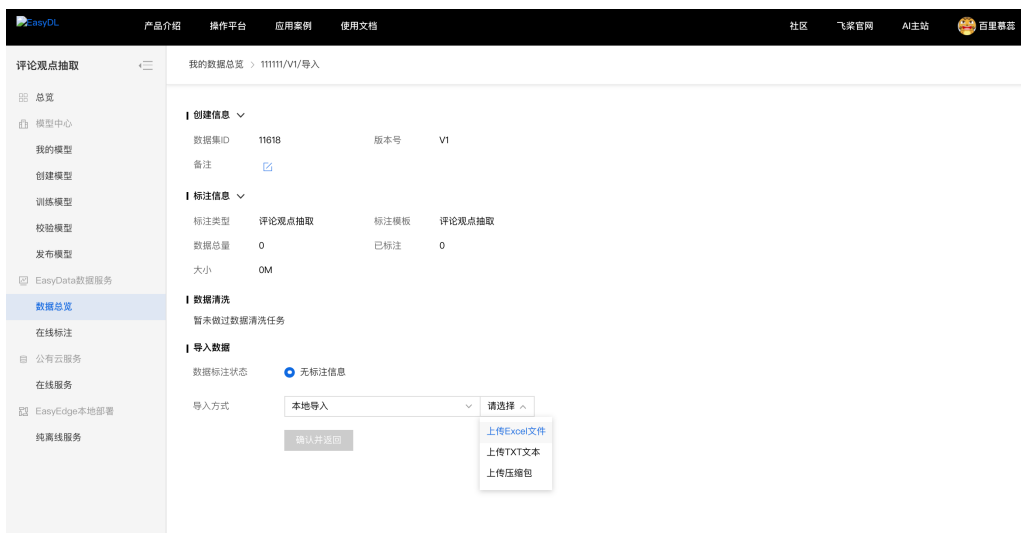
点击【导入】进入到新创建的评论观点抽取数据集中，平台暂只支持上传无标注信息的数据。



导入方式，分别为「本地导入」、「BOS目录导入」、「分享链接导入」、「平台已有数据集」；

通过本地导入时，可通过excel文件、TXT文件、压缩包形式上传

- 通过文本上传时，需注意：



文本文件内数据格式要求为"文本内容\n"（即每行一个样本，使用回车换行），每一行表示一组数据，每组数据的数建议不超过512个字符，超出将被截断

文本文件类型支持txt，编码仅支持UTF-8，单次上传限制100个文本文件，最多可上传100万个文本文件。

- 通过压缩包上传时，需注意：

压缩包内的一个文本文件将作为一个样本上传。压缩包格式为.zip格式，压缩包内文件类型支持txt，编码仅支持UTF-8。

每组数据的字数建议不超过512个字符，超出将被截断。

- 通过Excel文件上传时，需注意：

Excel文件内首行为表头，每行为一个样本，每个样本字符数不得超过512个字符，超出将被截断

文件格式支持xlsx格式，单次可上传100个文件。

#### 通过BOS目录导入格式要求

请确保将全部文本已通过txt文件保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入

#### 分享链接导入格式要求

请确保将全部文本文件保存至同一压缩包，压缩包仅支持zip格式，压缩前源文件大小限制5G以内；仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接

#### 通过平台已有数据集导入

直接点选您需要的数据集即可导入。

#### 其他：暂不支持API接口上传服务

### 🔗 数据去重策略

#### 重复样本的定义

重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。例如：

文本内容
理发师的手艺真不错
理发师的手艺真不错

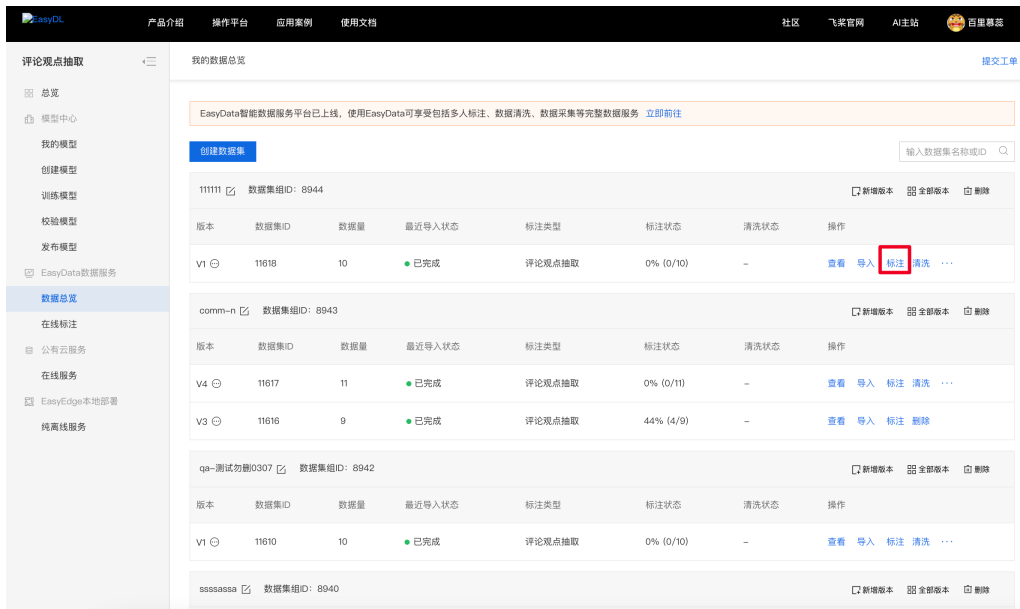
#### 平台去重策略

平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。注意：当您确定了数据集为去重或非去重的属性后，便不可修改。

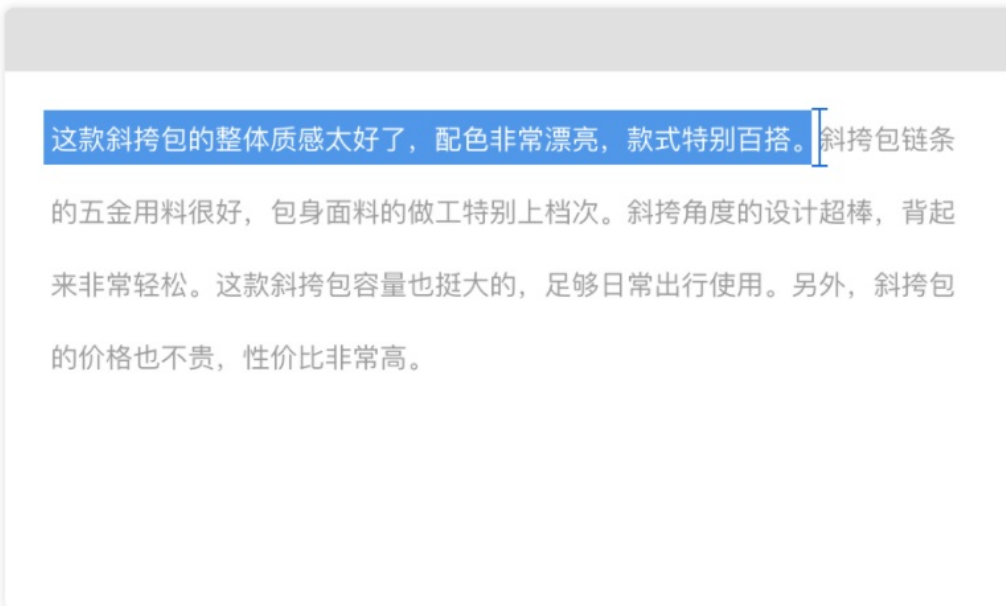
当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。

### 🔗 数据标注

通过平台导入「无标注信息」的数据集后，可对无标注数据进行标注操作。



1. 选取评价片段



2. 选取

评价对象 (必填)

画选评价片段后，会出现弹窗如下，请激活图中按钮划选评价对象或手动输入评价对象。



3. 自动预标注

- 自动预标注可在您输入评价对象之后，自动为您标注该评价片段中针对该评价对象的评价维度、评价观点，以及标注情感倾向。

在线标注 > 111111 > 标注

全部(10) 无标注信息(0) 有标注信息(10) 简约模式  关 标注说明

定位文本内容   标为负例  删除文本 | 上一篇 下一篇

这款斜挎包的整体质感太好 **斜挎包-质感-好-正向** 了，配色非常漂亮 **斜挎包-配色-漂亮-正向**，款式特别百搭 **斜挎包-款式-特别百搭-正向**。斜挎包链条的五金**用料很好**，包身面料的做工特别上档次。斜挎角度的设计超棒，背起来非常轻松。

\* 评价对象

**自动预标**  自动标出评价维度、评价观点、情感倾向

以下三个字段，如无可标注信息，可直接保存

评价维度

起始位置   结尾位置

评价观点

起始位置   结尾位置

情感倾向

- 预标注完成后您也可以激活划选按钮对预标注的评论维度进行增加、删除、修改等操作，评价观点暂不支持多个。
- 如不使用自动预标功能，您也可以激活划选按钮自行选择文字内容。
- 对于评价维度、评价观点，您还可以点击加减号左右移动选择文本范围。

在线标注 > 111111 > 标注

全部(10) 无标注信息(0) 有标注信息(10) 简约模式  关 标注说明

定位文本内容   标为负例  删除文本 | 上一篇 下一篇

这款斜挎包的整体质感太好 **斜挎包-质感-好-正向** 了，配色非常漂亮 **斜挎包-配色-漂亮-正向**，款式特别百搭 **斜挎包-款式-特别百搭-正向**。斜挎包链条的五金**用料很好**，包身面料的做工特别上档次。斜挎角度的设计超棒，背起来非常轻松。

\* 评价对象

自动预标  自动标出评价维度、评价观点、情感倾向

以下三个字段，如无可标注信息，可直接保存

评价维度

起始位置   结尾位置

评价观点

起始位置   结尾位置

情感倾向

4.完成标注 完成标注后效果如图：



确认标注无误, 可点击下一篇 (即为保存)。

**5. 标为负例** 若您认为此篇文本无任何评论观点, 则可支持选择【标为负例】, 此类样本将不会参与训练, 后续模型遇到此类样本也无需预测。

**6. 简约模式** 开启简约模式后, 会隐藏掉详细的标注结果 (评价对象-评价维度-评价观点-情感倾向), 方便您阅读文本内容, 如图为简约模式与普通模式对比:



## 模型训练

### 创建模型

#### 步骤

1. 在【模型中心-我的模型】点击创建模型或直接进入到【模型中心-创建模型】;
2. 选择模型类型、提交模型名称、模型描述、联系方式即可创建模型。
3. 模型创建成功后, 可以在【我的模型】中看到刚刚创建的模型。



1. 创建模型后可持续新增模型版本, 因此不必每次训练模型都创建模型
2. 目前单个用户在每种类型的模型下最多可创建10个模型, 每个模型均支持多次训练, 若需要创建超过10个以上的模型, 请在百度云控制台内[提交工单](#)反馈。
3. 如果您是是企业用户, 建议您按照真实企业信息进行填写, 便于EasyDL团队后续更好的为您服务

效果评估

模型评估报告

训练完成后, 可以在【我的模型】列表中看到模型效果, 以及详细的模型评估报告。

「完整评估结果」页面中将记录整体评估与详细评估的报告, 包括该模型整体的准确率、F1-score、精确率、召回率, 以及评估样本具体数据情况, 各分类的精确值、F1-Score、召回值等指标。

整体评估中, 各指标的释义如下:

- F1-score : 给每个类别相同的权重, 计算每个类别的F1-score, 然后求平均值
- 精确率 : 给每个类别相同的权重, 计算每个类别的精确率, 然后求平均值
- 召回率 : 给每个类别相同的权重, 计算每个类别的召回率, 然后求平均值

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V2	训练完成	已发布	召回率: 52.94% 精确度: 71.08% F1-score: 0.599 <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">服务详情</a> <a href="#">校验</a>



实际效果可以在左侧目录中找到【模型校验】功能进行校验, 或者发布为接口后测试。

模型校验操作步骤:

上传文本/自行输入文本--输入评价对象和勾选抽取内容--校验按钮--右侧返回json结果。

1. 在上传文本时, 只支持上传格式为txt, 文本文件内数据格式要求为"文本内容\n" (即每行一个样本, 使用回车换行), 每一行表示一组数

据，每组数据的数建议不超过512个字符，超出将被截断。

2. 您可以手动输入您想要抽取的评价对象，也支持不输入评价对象进行抽取。
3. 选择您想要抽取的评价字段（希望抽取评价片段、评价维度、评价观点，还是判断评论情感倾向，此处只支持单选）
4. 校验过程需要等待数秒，返回结果会在右侧以json形式显示出来，支持下载查看json文档

#### 模型校验示意图：



#### 优化效果

通过模型迭代、检查并优化训练数据、选择高精度模型等方法，能够提升模型效果。 **\*\*模型迭代\*\***

一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

#### **\*\*检查并优化训练数据\*\***

- 新增训练数据
- 通过模型效果评估报告中的各分类的详细评估指标，有针对性地扩充训练数据
- 检查测试模型的数据与训练数据的文本类型与风格是否一致，如果不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

#### **\*\*选择高精度模型\*\***

在训练模型时，选择高精度的模型，将提升模型的预测准确率。

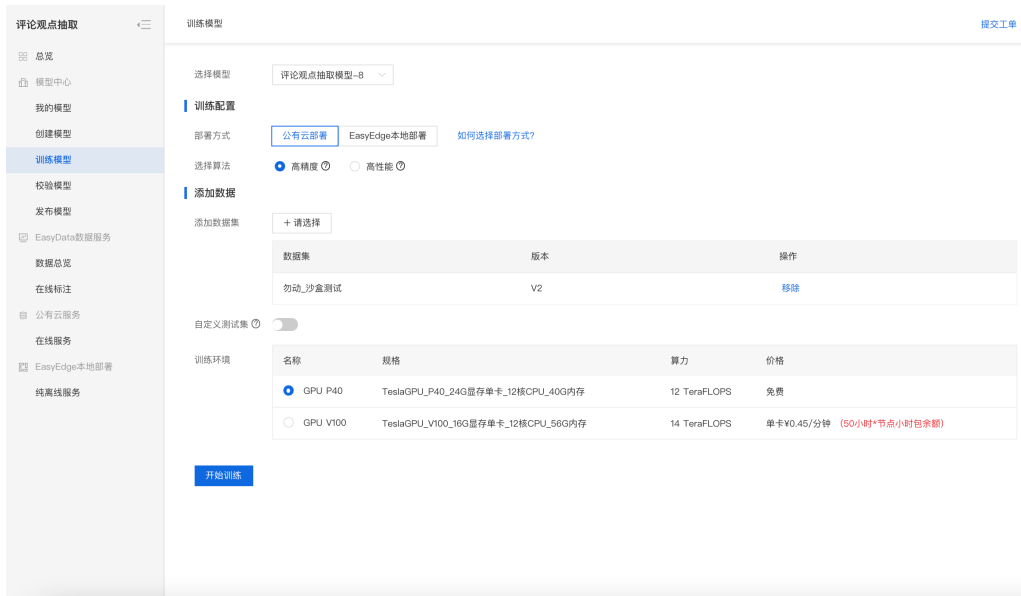
「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

#### 发起训练

#### **\*\*训练模型\*\***

完成数据的标注，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：



### Step 1 选择模型 选择此次训练的模型 Step 2 训练配置

#### 部署方式

可选择「公有云部署」、「EasyEdge本地部署」。

#### 如何选择部署方式

#### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择
- 如果您选择了「公有云部署」，无需选择设备

#### 选择算法

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。如果您手中的标注数据集样本较少（例如少于1000条），可选择「高精度」的算法；如果您手中有充足的数据集，您可选择「高性能」的算法。

- 高精度：预测准确率效果更高，训练时长与训练文本的长度和数量成正比，1000个样本预计在20分钟完成训练
- 高性能：在相同续联数据量的情况下，有着更快的预测速度，但准确度效果平均损失1~4%

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

### Step 3 添加数据

#### 添加训练数据

- 先选择数据集，再按标签（评价观点词、评价维度、情感倾向、评价片段）选择数据集里的文本，可从多个数据集选择文本
- 训练时间与数据量大小、选择的算法、训练环境有关

#### 添加自定义测试集

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

#### 添加自定义测试集的目的：

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可



## Step 4 训练模型

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面
- 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。可参考[价格说明](#)

## 模型发布

### 🔗 整体说明

训练完成后，可将模型部署在公有云服务器、私有化服务器上，通过API进行调用。**公有云API**

- 模型训练完毕后，为了方便企业用户一站式完成AI模型应用，评论观点抽取模型支持将模型发布成为在线的restful API接口，可以参考[示例文档](#)通过HTTP请求的方式进行调用，快速集成在业务中进行使用。
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

**相关费用** 将模型发布为公有云API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。详见[EasyDL价格文档](#)。

### 私有服务器部署

支持将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷。适用于对数据敏感度、隐私性要求较高、在线离线均有调用需求的企业场景。**相关费用** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月免费试用。如需购买永久使用授权，请[提交工单](#)咨询。

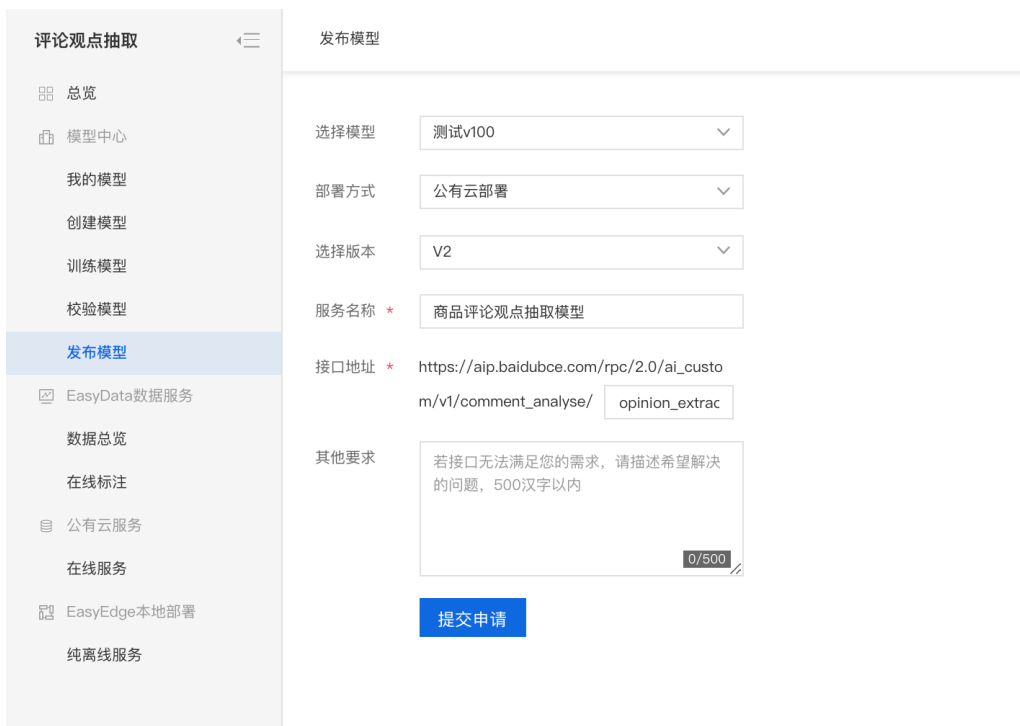
### 🔗 公有云API

### 🔗 发布API

#### 发布模型生成在线API

训练完毕后就可以在左侧目录栏中找到【发布模型】，发布模型表单页面需要自定义接口地址后缀、服务名称，即可申请发布。

发布模型界面示意：



或者，在我的模型列表——找到新训练好的模型版本——点击申请发布

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V2	● 训练完成	未发布	召回率: 52.94% 精确度: 71.08% F1-score: 0.599 完整评估结果	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>

申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成，如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈

申请发布通过后，界面和状态示意：

服务名称	模型名称	模型版本	服务状态	调用单价	更新时间	操作
评论观点抽取模型	测试v100	V2	● 发布中	9点/次	2022-03-10 16:31	-
1111111111	test-xsy	V7	● 已发布	9点/次	2022-03-09 11:30	<a href="#">服务详情</a> <a href="#">更新版本</a>
沙盒模型测试	沙盒测试-勿动	V3	● 已发布	4点/次	2022-03-07 16:07	<a href="#">服务详情</a> <a href="#">更新版本</a>

## 调用API

### 接口描述

基于自定义训练出的评论观点抽取模型，实现对评论文本中评价维度、评价观点的抽取以及对评价观点的情感倾向的判断。模型训练完毕后发布可获得定制化评论观点抽取API 详情访问：[定制化训练和服务平台](#)进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[定制化训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "电影故事结构很松散，但是小丽的演技一如既往的不错，配合她的男演员挺好的，比如说小杰，他在对手戏中有很强的互动性",
  "analyse_object": "小丽",
  "analyse_type": 1
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	超过512个token将被截断
analyse_object	否	string	-	评论实体对象
analyse_type	是	int	1/2/3/4	只能选择枚举值的一个

请求示例代码

Python3

```
**coding=utf-8**

import sys
import json

**保证兼容python2以及python3**
IS_PY3 = sys.version_info.major == 3
if IS_PY3:
    from urllib.request import urlopen
    from urllib.request import Request
    from urllib.error import URLError
    from urllib.parse import urlencode
    from urllib.parse import quote_plus
else:
    import urllib2
    from urllib import quote_plus
    from urllib2 import urlopen
    from urllib2 import Request
    from urllib2 import URLError
```

返回说明

字段	是否必选	类型	说明
analyse_type	是	int	1代表评价片段，2代表评价维度，3代表评价观点词，4代表评价情感倾向；示例：analyse_type:"1"或analyse_type:"4"，每次请求仅为单值；
log_id	是	number	唯一的log id，用于问题定位
result	是	array(object)	需要计算的评价对象数组，元素为字典
+start_offset	是	int	开始位置
+prob	是	float	置信度
+end_offset	是	int	结束位置
+text	是	string	抽取文本
text	是	string	输入文本，超过512个token将被截断

### 返回示例

```
{
  "analyse_type":1,
  "log_id":7000918336750814129,
  "result":[
    {
      "start_offset": 10,
      "prob": 0.4721265733242,
      "end_offset": 24,
      "text": "但是小丽的演技一如既往的不错"
    }
  ],
  "text": "电影故事结构很松散，但是小丽的演技一如既往的不错，配合她的男演员挺好的，比如说小杰，他在对手戏中有很强的互动性"
}
```

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 调用API

### 接口描述

基于自定义训练出的评论观点抽取模型，实现对评论文本中评价维度、评价观点的抽取以及对评价观点的情感倾向的判断。模型训练完毕后发布可获得定制化评论观点抽取API 详情访问：[定制化训练和服务平台](#)进行训练。

更多训练模型过程中的常见问题请查看 [常见问题文档](#)。

如有其它问题，请在百度云控制台内[提交工单](#)反馈。

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[定制化训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "电影故事结构很松散，但是小丽的演技一如既往的不错，配合她的男演员挺好的，比如说小杰，他在对手戏中有很强的互动性",
  "analyse_object": "小丽",
  "analyse_type": 1
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	超过512个token将被截断
analyse_object	否	string	-	评论实体对象
analyse_type	是	int	1/2/3/4	只能选择枚举值的一个

请求示例代码

Python3

```
**coding=utf-8**

import sys
import json

**保证兼容python2以及python3**
IS_PY3 = sys.version_info.major == 3
if IS_PY3:
    from urllib.request import urlopen
    from urllib.request import Request
    from urllib.error import URLError
    from urllib.parse import urlencode
    from urllib.parse import quote_plus
else:
    import urllib2
    from urllib import quote_plus
    from urllib2 import urlopen
    from urllib2 import Request
    from urllib2 import URLError
```

返回说明

字段	是否必选	类型	说明
analyse_type	是	int	1代表评价片段，2代表评价维度，3代表评价观点词，4代表评价情感倾向；示例：analyse_type:"1"或analyse_type:"4"，每次请求仅为单值；
log_id	是	number	唯一的log id，用于问题定位
result	是	array(object)	需要计算的评价对象数组，元素为字典
+start_offset	是	int	开始位置
+prob	是	float	置信度
+end_offset	是	int	结束位置
+text	是	string	抽取文本
text	是	string	输入文本，超过512个token将被截断

### 返回示例

```
{
  "analyse_type":1,
  "log_id":7000918336750814129,
  "result":[
    {
      "start_offset": 10,
      "prob": 0.4721265733242,
      "end_offset": 24,
      "text": "但是小丽的演技一如既往的不错"
    }
  ],
  "text": "电影故事结构很松散，但是小丽的演技一如既往的不错，配合她的男演员挺好的，比如说小杰，他在对手戏中有很强的互动性"
}
```

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群（868826008）或在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## ☞ 纯离线服务

## ☞ 发布API

在训练模型时，您需要选择「EasyEdge本地部署」的训练方式，才能发布本地部署的私有API。

### 私有API介绍

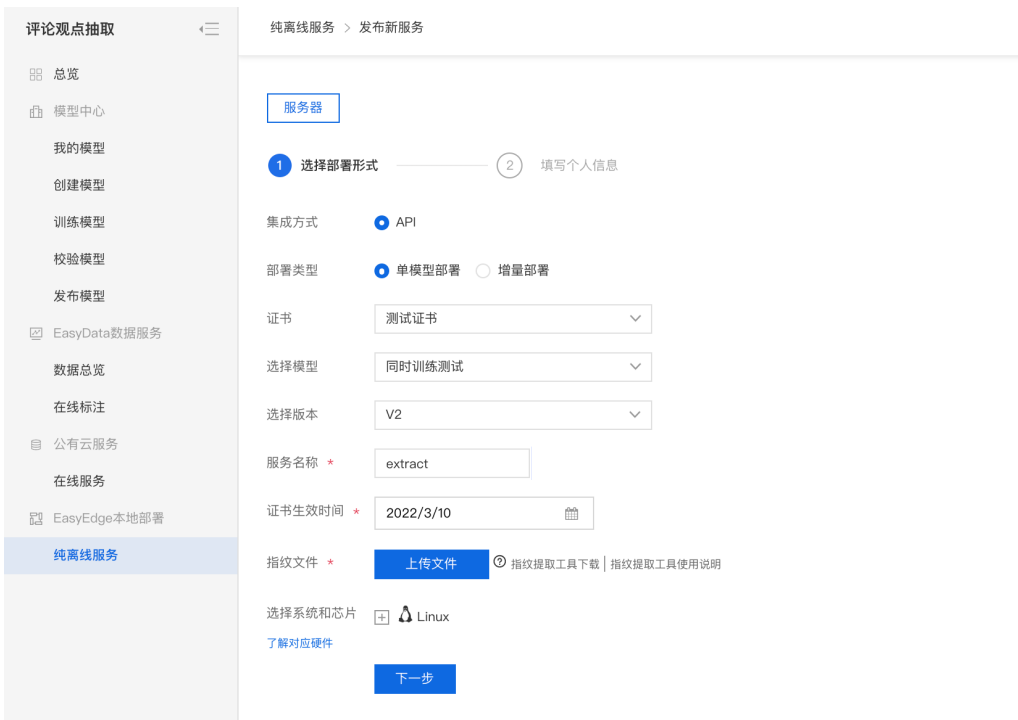
将模型以Docker形式在本地服务器（仅支持Linux）上部署为http服务，可调用与公有云API功能相同的接口。可纯离线完成部署，服务调用便捷

**发布私有API的流程** 训练完毕后，您可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可将模型部署到私有服务器：

1. 在「发布模型」页面中，选择模型及模型的版本，选择部署方式为「EasyEdge本地部署」、集成方式为「API-纯离线服务」。点击「发布」，即可跳转至「发布新服务」页面

**发布模型页面示意：**





2. 在「发布新服务」页面，选择部署类型，填写服务名称、证书生效时间等信息，选择对应的系统和芯片。

- 部署类型可支持单模型部署和增量部署
- 增量部署申请，指需要在一台服务器上部署多个模型部署包时使用。进行增量部署时，需在「已部署服务」选择同台服务器历史中最近部署的部署包，此步骤用来关联不同部署包中的license文件

3. 上传指纹文件。详细操作见[指纹提取工具说明](#)，可通过[指纹工具](#)进行指纹的提取

4. 点击下一步，填写个人详细信息后即可发布。发布完成后，即可在服务器目录下看到发布处于审核中的状态

个人信息的填写仅供EasyDL团队了解您，便于后续合作接洽，不会作为其他用途使用

5. 等待审核通过，前往「纯离线服务」等待部署包制作完成后，下载部署包，并[参考文档](#)完成集成

**价格说明** EasyDL已支持将定制模型部署在本地服务器上，只需在发布模型时提交本地服务器部署申请，通过审核后即可获得一个月**免费试用**。

如需购买永久使用授权，请[提交工单](#)咨询。

## 🔗 调用API

本文档主要说明定制化模型本地部署后，如何使用本地API。如还未训练模型，请先前往[EasyDL](#)进行训练。

如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:868826008）联系群管

## 部署包使用说明 部署方法

EasyDL定制化文本分类模型的本地部署通过EasyPack实现，目前提供单机一键部署的方式。

在EasyDL申请、下载部署包后，在本地服务器新建目录（建议目录命名规则：easyDL\_服务名称\_模型版本号），将软件包上传至该目录。请参考[EasyPack-单机一键部署](#)使用python2版本来部署，部署成功后，启动服务，即可调用与在线API功能类似的接口。

## 运维检查

EasyDL服务器API部署应用健康检查（或故障排查）脚本：[trouble\\_shooting.tar](#)

脚本能力：鉴权服务健康检测、容器状态检查、端口探活、网络连通性测试、容器关键报错日志输出等

**使用方法:** 将脚本上传至服务器任意目录（或在服务器直接下载），并解压后运行。

```

**解压**
tar vxf trouble_shooting.tar
**执行**
bash trouble_shooting.sh

```

### 授权说明

本地部署包根据服务器硬件（CPU单机或GPU单卡）进行授权，只能在申请时提交的硬件指纹所属的硬件上使用。

部署包测试期为1个月，如需购买永久授权，可[提交工单](#)咨询

### API参考

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL](#)进行自定义模型训练，完成训练后申请本地部署，本地部署成功后拼接url。

请求URL：[http://{IP}:{PORT}/{DEPLOY\\_NAME}/TextClassification](http://{IP}:{PORT}/{DEPLOY_NAME}/TextClassification)

- IP：服务本地部署所在机器的ip地址
- PORT：服务部署后获取的端口
- DEPLOY\_NAME：申请时填写的本地服务名称

Header如下：

参数	值
Content-Type	application/json

Body请求示例：

```

{
  "text": "电影故事结构很松散，但是小丽的演技一如既往的不错，配合她的男演员挺好的，比如说小杰，他在对手戏中有很强的互动性",
  "analyse_object": "小丽 ",
  "analyse_type ":1
}

```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
text	是	string	-	超过512个token将被截断
analyse_object	否	string	-	评论实体对象
analyse_type	是	int	1/2/3/4	只能选择枚举值的一个

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
analyse_type	是	int	1代表评价片段, 2代表评价维度, 3代表评价观点词, 4代表评价情感倾向; 示例: analyse_type:"1"或analyse_type:"4", 每次请求仅为单值;
log_id	是	number	唯一的log id, 用于问题定位
result	是	array(object)	需要计算的评价对象数组, 元素为字典
+start_offset	是	int	开始位置
+prob	是	float	置信度
+end_offset	是	int	结束位置
+text	是	string	抽取文本
text	是	string	输入文本, 超过512个token将被截断

### 返回示例

```
{
  "analyse_type": 1,
  "log_id": 7000918336750814000,
  "result": [
    {
      "start_offset": 10,
      "prob": 0.4721265733242,
      "end_offset": 24,
      "text": "但是小丽的演技一如既往的不错"
    }
  ],
  "text": "电影故事结构很松散, 但是小丽的演技一如既往的不错, 配合她的男演员挺好的, 比如说小杰, 他在对手戏中有很强的互动性"
}
```

### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如缺少必要出入参时返回:

```
{
  "error_code": 336001,
  "error_msg": "Invalid Argument"
}
```

错误码	错误信息	描述
336000	Internal error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (868826008) 或工单联系技术支持团队
336001	Invalid Argument	入参格式有误, 比如缺少必要参数、文本的编码UTF-8等问题。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误, 比如缺少必要参数代码格式是否有误。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误, 请根据接口文档检查格式, base64编码请求时注意要去掉头部。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336004	输入文件大小不合法	文本大小不合法, 目前支持文本文件类型为支持txt, 文本文件大小限制长度最大1024 UTF-8字符。有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
336005	解码失败	文本编码错误 (不是utf-8), 目前支持文本文件类型为支持txt。如果遇到请重试, 如反复失败, 请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求, 可恢复正常, 若反复重试依然报错或有疑问请通过QQ群 (868826008) 或工单联系技术支持团队
337000	Auth check failed	离线鉴权调用失败

**模型更新/回滚操作说明** **模型更新** 1、在EasyDL-纯离线服务发布页面, 找到您的服务器API发布记录, 点击【更新版本】, 选择「更新包」或「完整包」来发布。两者区别:

包类型	描述
更新包	仅包含最新的模型应用, 需执行download.sh脚本下载所需镜像等依赖文件
完整包	包含模型应用和其他鉴权服务, 需执行download.sh脚本下载所需完整依赖文件

2、(CPU模型可忽略) 如果您训练的模型为GPU版本, 系统会生成多份下载链接。请在GPU服务器执行 `nvidia-smi` 命令, 根据返回的Cuda Version来选择对应的部署包链接下载。

3、在服务器新建目录 (建议标记对应模型的版本号, 便于区分不同模型版本), 如 `easydl_${DEPLOY_NAME}_v2`

`${DEPLOY_NAME}`: 申请时填写的服务名称

以如下场景举例说明: 模型版本从V1升级至V2

```

**1.新模型准备**
**创建指定版本的目录**
mkdir easedl_${DEPLOY_NAME}_v2
cd easedl_${DEPLOY_NAME}_v2
**将部署包上传至服务器该目录并解压**
tar zxf xx.tar.gz
**解压后, 进入指定目录执行download脚本下载模型所依赖文件**
cd original && bash download.sh
**2.旧模型备份**
**历史模型备份**
cp -r /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V1
**记录当前模型的端口号**
docker ps -a |grep ${DEPLOY_NAME}
**3.模型升级**
cd package/Install
**卸载当前已安装的旧的easyDL服务: ${DEPLOY_NAME}, 前面已备份**
python2 install.py remove ${DEPLOY_NAME}
**安装当前部署包内新的EasyDL服务: ${DEPLOY_NAME}**
python2 install.py install ${DEPLOY_NAME}
**(可选操作) 更新证书**
python2 install.py lu

```

**模型回滚** 以如下场景举例说明: 模型版本从V2回滚至V1

方法一:

```

**重命名当前v2模型目录名称**
mv /home/baidu/work/${DEPLOY_NAME} /home/baidu/work/${DEPLOY_NAME}_V2
**使用V1版本**
cp -r /home/baidu/work/${DEPLOY_NAME}_V1 /home/baidu/work/${DEPLOY_NAME}
**停止当前模型容器**
docker ps -a |grep ${DEPLOY_NAME}
docker rm -f ${容器名}
**创建新的容器**
cd /home/baidu/work/${DEPLOY_NAME} && bash start/start-1.sh
**（可选操作）进入V1版本部署包所在目录执行license更新操作，假如部署包在/opt目录下,以您实际目录为准**
cd /opt/easydl_${DEPLOY_NAME}_V1/original/package/Install && python2 install.py lu

```

方法二：

进入模型V1所在目录，参考上述【模型更新】步骤，执行模型升级操作（即先卸载v2，后升级为v1）

## 文本创作（已下线）

### 文本创作介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

定制文本创作模型，基于ERNIE 3.0大模型实现对输入文本内容进行创作和续写。平台提供标注工具，您可在平台上传文档，完成标注后可直接进行模型训练。

更多详情访问：[EasyDL自然语言处理方向](#)

**应用场景** 1、新闻资讯创作：训练新闻资讯文本的自动续写和创作模型，节省大量编辑人力

2、文章摘要：训练对网络媒体等文章的自动摘要，进而实现各类文章的自动摘要

3、诗歌对联创作、专业文本续写：定制训练文学创作的模型，对诗词、对联和专业文本进行创作和续写

4、营销广告文案创作：定制训练广告文案创作的模型，给广告营销带来新的创作灵感

5、其他：尽情脑洞大开，训练你希望实现的创作模型 **技术特色** 文本创作模型内置文心大模型，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

文心大模型是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

**使用流程** 训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。

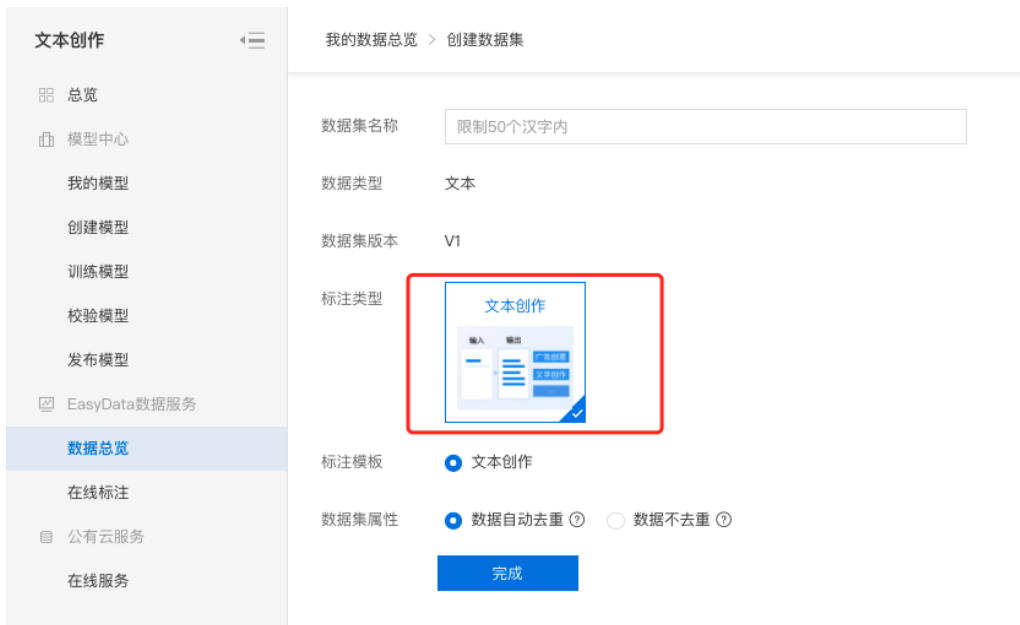


### 数据准备

#### 创建数据集并导入

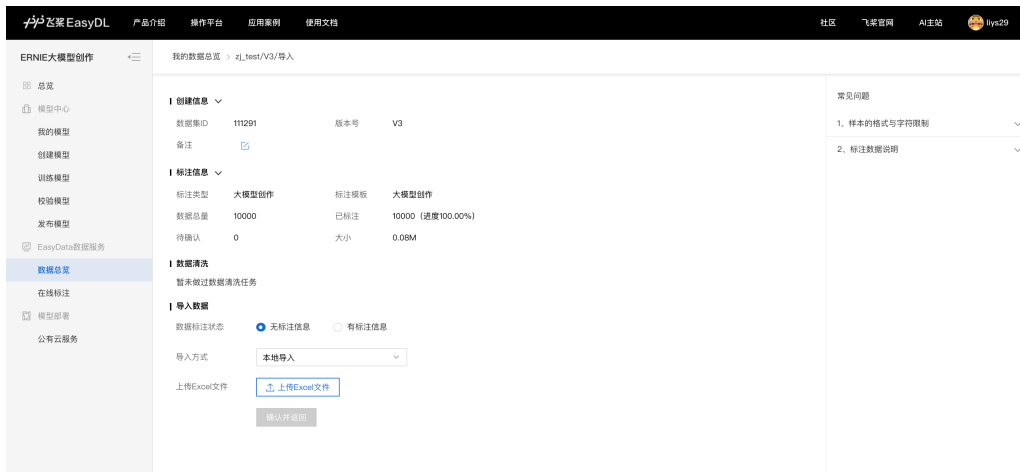
##### 1. 创建数据集

您可以在左侧导航栏中，选择“数据总览”并点击主内容区域的按钮「创建数据集」，默认数据类型为“文本”，标注类型为“文本创作”。



## 2. 导入文本数据

进入到新创建的文本创作数据集中。您可以在文本创作任务的数据集中，上传带有标注信息的数据，和无标注信息的数据。

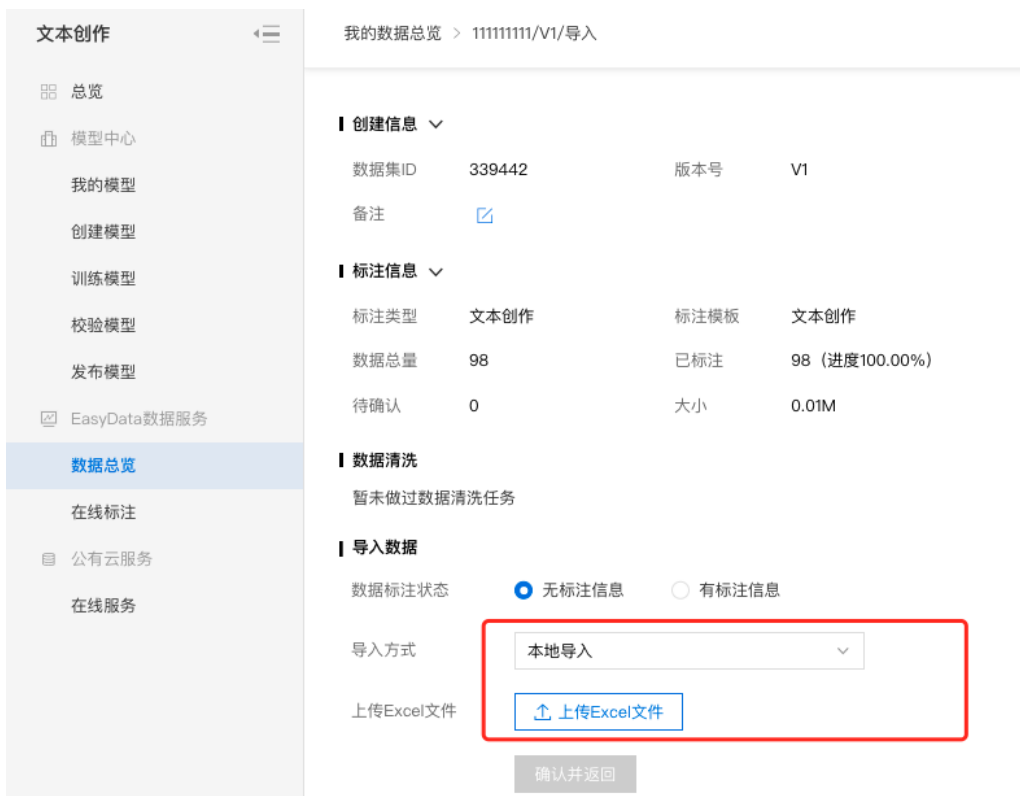


在数据导入方式选择本地数据集，根据您已有的数据存储格式，选择上传格式。目前对未标注数据和已标注数据都仅提供了Excel文件上传方式。

### 2.1 未标注数据上传方式：

#### 以Excel文件上传

1. 第一列作为原始文本，每行是一组样本，首行为表头默认将被忽略，每组数据文本内容的字符数不超过512个字符（包括中英文、数字、符号等），超出的字符可正常保存，但可能无法参与训练。详见平台导入数据处的数据样例。
2. 文件类型支持xlsx格式，单次上传限制100个文件；文件格式示意图如下：



示例：例如在歌词创作场景中，希望用户输入歌名，由模型创作歌词，则上传文本为：“歌名：晴天”。

请注意，“歌名：”作为样本的前缀，需要固定在每一个样本中添加，不固定的前缀，将影响模型效果；

示例样本请详见平台导入数据处的数据样例。

## 2.2 已标注数据上传方式：

### 以Excel文件导入

- Excel文件内数据格式要求为：首行为表头，将不录入数据集中，第一列和第二列分别作为模型输入文本和模型输出文本
- 每行是一组样本，输入文本不超过512个字符，输出文本不超过128个字符，超出的字符可正常保存，但可能无法参与训练。（字符包括中英文、数字、符号等）
- 文件类型支持xlsx格式，单次上传限制100个文本文件；文件格式示意图如下：

输入文本内容	输出文本内容
文本内容a1	文本内容a2
文本内容a1	文本内容b2
文本内容a3	文本内容c1

示例：例如在歌词创作场景中，希望用户输入歌名，由模型创作歌词，此场景的标注数据形式可有多种：

数据格式一：

- 输入文本内容：“歌名：晴天；歌词：”
- 输出文本内容：“故事的小黄花 从出生那年就飘着 童年的荡秋千”

在数据格式一中，输入到模型的文本是“歌名：晴天；歌词：”，“歌名：”作为模型输入的前缀，需要固定在每一个样本中添加，不固定的前缀，将影响模型效果；“歌词：”作为模型输出样本的后缀，需要固定在每一个样本中添加，不固定的后缀，将影响模型效果；

您需要在模型预测阶段，确保回传模型输入包含了用户输入的内容，并且拼接了前缀“歌名：”和后缀“歌词：”，在模型服务返回内容时，则直接输出歌词。

数据格式二：

- 输入文本内容：“歌名：晴天；”
- 输出文本内容：“歌词：故事的小黄花 从出生那年就飘着 童年的荡秋千”

在数据格式二中，前缀“歌名：”和后缀“歌词：”分别在模型的输入和输出中。“歌名：”作为输入文本的前缀，需要固定在每一个样本中添加，不固定的前缀，将影响模型效果；“歌词：”作为输出样本的前缀，需要固定在每一个样本中添加，不固定的后缀，将影响模型效果；

您需要确保预测推理阶段，模型输入包含了用户输入的内容，并且拼接了前缀“歌名：”

数据格式三：

- 输入文本内容：“晴天”
- 输出文本内容：“故事的小黄花 从出生那年就飘着 童年的荡秋千”

在数据格式三中，前缀“歌名：”和后缀“歌词：”都不存在模型的训练数据，则所有样本都不要添加前缀和后缀。则您需要确保在预测推理阶段，模型输入仅有歌名内容。

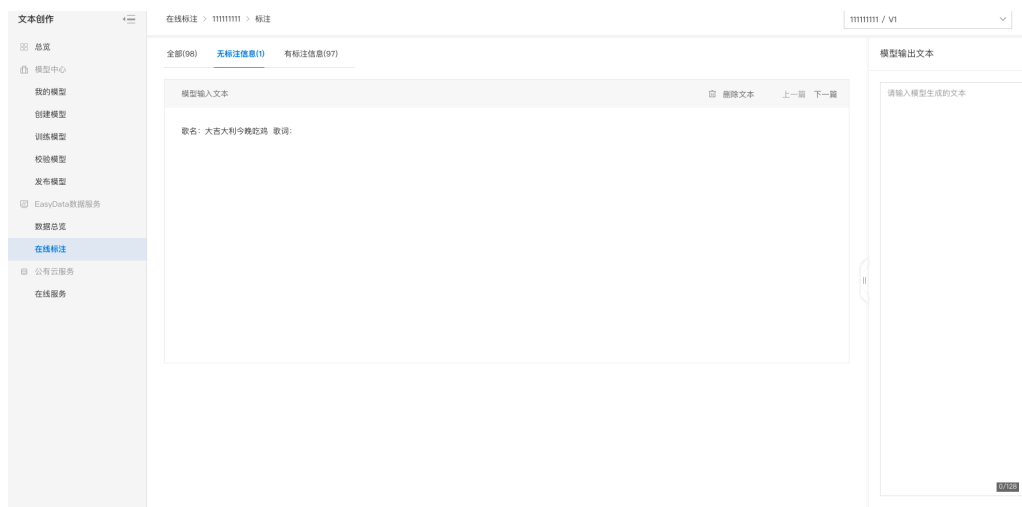
详见平台导入数据处的数据样例。

4. 上传时，单个数据集总量仅支持上传10000条样本（包括标注数据和未标注数据）。

## 🔗 文本创作数据标注

### 文本创作数据标注

1. 图中所示，模型输出文本框中，您可直接编辑模型输出文本，如图所示：



2. 编辑完成后，点击“下一篇”按钮自动保存并进入下一篇数据

## 🔗 文本创作数据集去重策略说明

**重复样本的定义** 一个样本包括文本内容和标签。重复样本的定义，是指您上传的数据中，存在两个样本的文本内容完全一致。则被判定为两个样本是重复样本。例如：

模型输入内容	模型输出内容
歌名：晴天；歌词：	故事的小黄花 从出生那年就飘着 童年的荡秋千
歌名：晴天；歌词：	故事的小黄花
歌名：晴；歌词：	故事的小黄花 从出生那年就飘着 童年的荡秋千

上表三个样本均为重复样本，前两个样本虽然标签不一，但文本内容一致，也为重复样本。

Tips：“如何利用好重复样本”，如果您在模型训练过程中，需要通过增加某个类别标签的预测权重，可以通过增加此标签的重复样本来达到此目标。



**平台去重策略** 平台提供了可去重的数据集，即对您上传的数据进行重复样本的去重。注意：当您确定了数据集为去重或非去重的属性后，便不可修改。

当您创建了一个去重的数据集时，在后续上传数据的过程中，平台可通过检验您当前上传的样本与已上传到此数据集下的样本是否相同，如果相同，则会使用新的样本替代旧的样本。此时分为几种情况，如下：

数据集中有未标注样本，上传重复的已标注样本，此时未标注样本将被覆盖

数据集中有已标注样本，上传重复的未标注样本，此时已标注样本将被覆盖

数据集中有已标注样本，上传不同标注的已标注样本，此时已有的标注样本将被覆盖

## 模型训练

### 模型创建

**创建模型** 在左侧目录【模型中心】下点击创建模型，按照指示，填入相应的模型名称，选择模型归属、所属行业、应用场景，以及您的下关系信息。填写完成后点击下一步，即可我完成模型的创建。

模型列表 > 创建模型

模型类别 文本创作

模型名称 \*

您的身份 企业管理者 企业员工 学生 教师

公司名称 \* 百度  
企业认证流程较快，认证过程中您可继续创建模型，完成后系统会自动同步状态。

所属行业 \* 请选择行业

应用场景 \* 请选择应用场景

邮箱地址 \* r\*\*\*\*\*@163.com

联系方式 \* 186\*\*\*\*356

功能描述 \* 0/500

完成

模型创建完成后，在左侧目录【模型中心】下点击我的模型，可以看到您已经创建成功的模型。

- 1.创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型
- 2.目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。
- 3.如果您是是企业用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务

### 模型训练

#### 创建任务

当您的模型以及数据集创建完成后，您可以点击左边目录导航栏中的【训练模型】，创建模型或选择您已经创建的模型，再添加您要使用的训练数据集，建议您使用的每个文本创作数据的样本数应达到1000个以上，再启动训练。

**训练环境** 平台为您提供了GPU算力机器，TeslaGPU V100\_32G显存单卡 80核CPU\_640G内存，训练设备数默认为8（暂不支持增删机器）。

添加数据集以及完成配置后，点击开始训练即可启动训练。

注意：文本创作任务，每次最高支持1万条样本的训练，训练时间最长约1个小时。在您提交任务后，需要与平台其他用户任务排队等待算力机器，此时间由排队任务数决定。

## 模型效果评估

### 模型评估报告

1. 校验指标：仅提供 BLEU-4指标：Bilingual evaluation understudy，BLEU 的分数取值范围是 0~100%，分数越接近100%，说明生成的句子质量越高。
2. 训练完成后，可以在【我的模型】列表中看到模型效果，以及详细的模型评估报告。

### 模型校验

实际效果可以在左侧目录中找到【模型校验】功能进行校验，或者发布为接口后测试。

校验过程中，您只需要输入或者上传校验文本（文本上限长度为512字符），即可进行校验，输出仅支持52token。

当前模型精确率 81.97% [评估报告](#)
识别结果 [如何优化效果?](#)

请输入校验的文本，或 [点击上传文本](#)    支持文本格式：txt，文本长度上限为512汉字（字符）

请输入文本

0/512

输出文本    仅支持出书52个token

没有满足条件的识别结果

[校验](#)
[再次上传文本](#)
[申请上线](#)

## 模型发布

### 模型发布

#### 发布模型生成在线API

模型训练完毕后，为了方便企业用户一站式完成AI模型应用，支持将模型发布成为在线的restful API接口，可以参考示例文档通过HTTP请求的方式进行调用，快速集成在业务中进行使用。具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求。

左侧目录栏中找到【模型部署-公有云服务-发布新服务】，填写发布模型表单页面，自定义接口地址后缀、服务名称，即可申请发布，平台上发布模型流程完全一致。

发布模型界面示意如下：

文本创作
公有云部署
[提交工单](#)

**公有云服务说明** [点击收起](#)

1. 发布公有云服务，将训练完成的模型部署在百度云服务器，通过API接口调用模型。如何将模型发布成为公有云服务以及调用代码示例参考：[参考文档](#)
2. 每个公有云服务根据选择算法配置不同，单次调用所消耗的点数也不同。具体的单价可以在下方服务列表“调用单价”中查看。换算规则：1G=0.001元。
3. 每个公有云服务发布成功后享有10000点免费额度，仅作为该服务调用消耗，如需稳定使用，建议在[控制台](#)开通付费。

[发布新服务](#)
[控制台](#)
输入模型名称

服务名称	模型名称	模型版本	服务状态	调用单价	更新时间	操作
emie3v1	binbin_test_1109	V1	● 已发布	-	2021-11-09 21:52	<a href="#">服务详情</a> <a href="#">更新版本</a>

或者，在模型中心——我的模型——选择模型发布。

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V4	训练完成	未发布	BLEU-4: 5.90% <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>
公有云API	V1	训练终止	未发布	-	<a href="#">查看版本配置</a>

发布后：

No.	数据集名称	操作
1	binbin_test_1109V1	<a href="#">查看详情</a>

申请发布后，通常的审核周期为T+1天，即当天申请第二天可以审核完成，如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈

## 🔗 文本创作-调用API文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

在百度云控制台内[提交工单](#)

进入[EasyDL社区交流](#)，与其他开发者进行互动

加入EasyDL官方QQ群（群号:868826008）联系群管

**接口描述** 基于自定义训练出的创作模型，实现基于输入文本内容的个性化创作。模型训练完毕后发布可获得定制API。请求说明 HTTP 方法：POST

请求URL：请首先在[定制化训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，定制化文本分类服务以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "max_gen_len": "64"
}
```

Body中放置请求参数，参数详情请看模型请求参数。

模型请求参数：

参数	是否必选	类型	可选值范围	说明
text	是	string	512token以内	输入文本，超过512个token将被截断
max_gen_len	否	int	0-128token	生成时返回字符数，可选0-128，可按需设置，通常生成字符数越少，用户等待时间越少。默认取值为：64

请求示例：

```
##### coding=utf-8

import sys
import json

##### 保证兼容python2以及python3
IS_PY3 = sys.version_info.major == 3
if IS_PY3:
    from urllib.request import urlopen
    from urllib.request import Request
    from urllib.error import URLError
    from urllib.parse import urlencode
    from urllib.parse import quote_plus
else:
    import urllib2
    from urllib import quote_plus
    from urllib2 import urlopen
    from urllib2 import Request
    from urllib2 import URLError
    from urllib import urlencode

reload(sys)
sys.setdefaultencoding('utf8')

##### 防止https证书校验不正确
import ssl

ssl._create_default_https_context = ssl._create_unverified_context

##### 百度云控制台获取到ak, sk以及
##### EasyDL官网获取到URL

##### ak
```

```
API_KEY = 'kQWXQ8oe5G5T7ATzXXXXXXXX'

##### sk
SECRET_KEY = 'Y30GtHsKzyH6fUUsQI32GvoBXXXXXXXX'

##### url
EASYDL_TEXT_CLASSIFY_URL = "https://aip.baidubce.com/rpc/2.0/ai_custom/v1/text_gen/lirics_gen"

""" TOKEN start """
TOKEN_URL = 'https://aip.baidubce.com/oauth/2.0/token'
"""

    获取token
"""

def fetch_token():
    params = {'grant_type': 'client_credentials',
              'client_id': API_KEY,
              'client_secret': SECRET_KEY}
    post_data = urlencode(params)
    if (IS_PY3):
        post_data = post_data.encode('utf-8')
    req = Request(TOKEN_URL, post_data)
    try:
        f = urlopen(req, timeout=5)
        result_str = f.read()
        print('success')
    except URLError as err:
        print(err)
    if (IS_PY3):
        result_str = result_str.decode()

    result = json.loads(result_str)

    if ('access_token' in result.keys() and 'scope' in result.keys()):
        if not 'brain_all_scope' in result['scope'].split(' '):
            print("please ensure has check the ability")
            exit()
        return result['access_token']
    else:
        print('please overwrite the correct API_KEY and SECRET_KEY')
        exit()

"""

    调用远程服务
"""

def request(url, data):
    if IS_PY3:
        req = Request(url, json.dumps(data).encode('utf-8'))
    else:
        req = Request(url, json.dumps(data))

    has_error = False
    try:
        f = urlopen(req)
        result_str = f.read()
        if (IS_PY3):
            result_str = result_str.decode()
        return result_str
    except URLError as err:
        print(err)

if __name__ == '__main__':

    # 获取access token
    token = fetch_token()
```

```

# 拼接url
url = EASYDL_TEXT_CLASSIFY_URL + "?access_token=" + token

text = "歌名：晴天；歌词："

# 请求接口
# 测试
response = request(url,
    {
        'text': text,
        'max_gen_len': 128
    })

result_json = json.loads(response)['result']['content']

print('u{0}'.format(result_json))

```

**模型返回参数：**

参数	是否必选	类型	可选值范围	说明
log_id	是	number	-	唯一的log id，用于问题定位
+content	否	string	-	返回的生成结果
+is_truncate	否	boolean	0或1	返回的生成结果是否被截断，1为被截断，0为没被截断，与设置的max_gen_len的token数有关

**示例样本1：**

以歌词创作场景为例，模型输入（入参）为歌词名称，模型输出（出参）为歌词内容，假设训练数据中，输入文本的数据模板的为：“歌名：xxx；歌词：”，输出文本的数据模板为：“xxxx”（xxxx代表生成歌词内容）。

当用户输入：“夏日的海边”

- 入参text字段为：“歌名：夏日的海边；歌词：”
- 出参content为：“故事的小黄花 从出生那年就飘着 童年的荡秋千”

其中，“歌名：”作为样本的前缀，需要固定添加在每一次的请求中，不固定的前缀，将影响模型效果；“歌词：”作为样本的后缀，需要固定添加在每一次的请求中，不固定的后缀，将影响模型效果；

**示例输入：**

```

{
  "text": "歌名：夏日的海边；歌词：",
  "max_gen_len": "64"
}

```

**示例返回：**

```

{
  "log_id": "123456",
  "result": [{
    "content": "故事的小黄花 从出生那年就飘着 童年的荡秋千",
    "is_truncate": 0
  }]
}

```

**示例样本2：以旅行问答场景为例：**

- 用户输入：“十月去青海应该带什么？”，则入参text字段为：“问题是：十月去青海应该带什么？答案是：”
- 输出：“带个男朋友”，则content为“带个男朋友”

其中，“问题是：”是前缀，“答案是：”是后缀，前缀避免改为“问题：”或“题目是：”等相关词组，后缀避免改为“答案：”或“回答：”等相关词组；前后缀都需要固定添加在每一次的请求中，否则将影响模型效果。

**示例输入：**

```
{
  "text": "问题是：十月去青海应该带什么？答案是：",
  "max_gen_len": "64"
}
```

示例返回：

```
{
  "log_id": "123456",
  "result": [{
    "content": "带个男朋友",
    "is_truncate": 0
  }]
}
```

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

error\_code：错误码。

error\_msg：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内提交工单反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	文本大小不合法，目前支持文本文件类型为支持txt，文本文件大小限制长度最大512 UTF-8字符。
336005	解码失败	文本编码错误（不是utf-8），目前支持文本文件类型为支持txt。如果遇到请重试，如反复失败，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336006	缺失必要参数	未上传文本文件
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## EasyDL 语音使用说明

### EasyDL语音介绍

#### ☞ 功能介绍

EasyDL语音，包含语音识别和声音分类两种训练能力，零代码自助训练语音识别语言模型，声音分类模型。提升业务领域专有名词识别准确率，区分不同声音类别，广泛适用于行业数据采集录入、语音指令、呼叫中心、声音类型检测等应用场景。

- **科学评估，提供多维报告**

上传业务场景音频和标注文本，系统自动评估语音识别基础模型得到基线准确率，输出字准、句准、核心词准等多维度评估结果报告

- **上传语料，深度训练模型**

选择基础模型上传业务场景相关文本训练语料即可自助训练语言模型，支持词汇、长文本等多种训练方式



- **迭代优化，获取最佳模型**

可多次上传文本数据迭代训练，每次训练后系统自动评估训练结果，训练效果精准提升，直观可视

- **自动上线，模型专属使用**

训练效果满意后，无需复杂操作，通过申请流程即可将模型上线使用，模型专属使用

## 特色优势

### 零门槛操作

一站式自动化训练，上传文件即可最快10分钟训练优化语言模型。

### 高精度评估

系统自动评估多种基础模型，推荐最优模型进行训练。训练前后均提供字准、句准、核心词准等多维度评估结果报告，

### 强训练效果

预置百度超大规模预训练模型，多个基础模型支持多行业多业务场景。支持词汇、长段文本等多种训练方式。支持多次上传训练文本，迭代训练不断优化模型，平均5%-25%识别准确率提升。

### 超灵活部署

模型通过申请流程即可自动上线，用户账号专属使用。支持在线API，websocket API，多种操作系统的SDK，适配多种终端的使用需求

## 应用场景

训练语音识别模型可以在如下的应用场景中获得更好的识别效果

- **语音对话**：APP语音助手，金融、医疗、航空公司智能机器人对话等短语音交互场景，使用领域中的专业术语进行训练，提高对话精准度
- **语音指令**：智能硬件语音控制、app内语音搜索关键词、语音红包等场景，训练固定搭配的指令内容，让控制更精确
- **语音录入**：农业采集、工业质检、物流快递单录入、餐厅下单、电商货品清点等业务信息语音录入场景，训练业务中的常用词，录入的结果更加有效
- **电话客服**：运营商、金融、地产销售等电话客服业务场景，使用领域中的专业术语进行训练，提高对话精准度

训练声音分类模型可以在如下的应用场景中定制区分不同的声音类型

- **安防监控**：定制识别不同的异常或正常的声音，进而用于突发状况预警
- **科学研究**：定制识别同一物种的不同个体的声音、或者不同物种的声音，协助野外作业研究

## 快速使用

语音技术下任一接口进行付费，即可免费训练语音识别模型，不收取额外的训练费用。

## 语音识别

### 语音识别介绍

Hi，您好，欢迎使用EasyDL语音识别。

原语音自训练平台即已结束公测正式上线，品牌升级更名为“EasyDL语音识别”，平台和语音识别通用接口全面打通，语音技术下任一接口开通付费即可免费训练语音识别模型，无需额外费用。

---

如果您在调用通用语音识别模型时遇到如下困难：

- 1、在垂直业务领域下通用语音识别模型准确率不满足需求，语音识别应用的场景专业词汇较集中，如医疗词汇、金融词汇、教育用语、交通地名、人名等，识别结果存在“同音不同字”的情况。例如“虹桥机场”识别为“红桥机场”；“债券”识别为“在劝”。
- 2、语音识别结果不准带来更高的后处理成本，并且语音识别模型针对性优化训练存在技术门槛、成本高、训练周期长。

欢迎使用EasyDL语音识别，可以通过自助训练语言模型的方式有效提升您业务场景下的识别准确率。

## 使用流程概述

平台使用的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快一天内即可获得专属模型。



1、**创建模型**：选择您需要训练的语音识别接口，目前支持训练**短语语音识别-中文普通话**、**短语语音识别极速版**、**实时语音识别-中文**、**呼叫中心语音解决方案**接口。填写基础信息为您的模型进行命名和功能描述，并留下您的联系方式以便于我们和您联系。

2、**系统评估**：上传您业务场景中的**真实音频和对应的正确标注文本**（尽可能覆盖全部的场景），**客观科学地评估基础模型的识别率**。根据评估结果，系统自动推荐最佳的基础模型，您可以选择任一基础模型进行训练。

3、**训练模型**：上传您**业务场景中出现的高频词汇或者是长句文本**，可以有效提升业务用语的识别率；并可以迭代训练，持续优化。

5、**上线模型**：得到满意的训练模型即可申请上线，审批通过自动上线模型。模型上线后，在语音识别的接口中配置模型参数即可使用训练后的效果。

开始使用平台前，先了解以下您需要提前准备的物料及准备建议：

1、【**测试集**（包括业务音频+准确100%的标注文本）】，用于评估基础模型识别率和训练后模型识别率，相当于准备一份“标准答案”。如果模型使用业务范围较广（例如某行业领域模型），建议测试集在1000-3000条左右评估会相对客观；如果是针对某些特定场景训练，可只提供该场景的音频测试集几十条-几百条均可，包含希望评估的业务内容即可。

2、【**训练集**（投入平台进行训练的文本）】，用于语言模型训练，建议文本要和测试集的内容强相关。训练文本可以放置希望提升识别效果的词汇，如业务上的固定搭配和业务关键词等，或者可以将某个词汇放在不同句式的句子中，高频出现。**\*\*影响训练效果的关键因素为“文本出现的频率”和“上下文的句意理解”等\*\***。无需重复提交大量文本，少量关键文本即可有训练效果。

进入**EasyDL语音识别**

输入用户名及密码，点击“登录”，进入EasyDL语音识别。可以看到整体训练流程，点击创建模型可以直接进行模型创建，点击模型中心可以进入到模型列表页面。



整体训练流程将按照目录栏的顺序依次操作即可。



下面将详细介绍每一步的操作方式和注意事项。若遇到的问题在此文档没有找到答案，可以加入官方QQ群（群号: 686267521）咨询群管。

## 创建模型

在导航栏【模型中心】-【我的模型】页中可以点击【创建模型】按钮；也可以直接点击左侧导航菜单中的【创建模型】进入创建模型步骤。目前一个账号下支持创建10个模型，模型可删除。

在创建模型步骤中，需要进行“基础信息填写”“上传测试集”“选择基础模型”三个环节完成创建。

测试集的作用是通过上传音频和正确的标注文本评估基础模型的识别率，根据基础模型识别率选择最合适的基础模型进行训练。等模型训练后系统自动使用该测试集评估得到训练后模型的识别率，可以直观的查看训练提升效果。



1. 基础信息：包括接口类型、模型名称、公司/个人、所属行业、应用场景、应用设备、功能描述、邮箱地址、联系方式

- **接口类型**：包括短语音识别（支持16K采样率音频）、实时语音识别（支持16K采样率音频）和呼叫中心场景（支持8K采样率音频）3种，用户可以基于应用场景和音频采样率来进行选择。
- **模型名称**：用户可自行填写模型名称，可支持中文、英文、数字、下划线.+#\*()^-
- **公司/个人**：模型归属企业则需要填写企业名称，归属个人则不需填写
- **所属行业**：企业业务或个人应用所属的行业信息
- **应用场景**：语音识别模型应用落地的业务场景
- **应用设备**：业务中使用语音技术的录音设备终端
- **功能描述**：描述模型应用的场景，有助于上线审核哦
- **邮箱地址**：填写联系人的邮箱地址，用于模型上线等信息的通知
- **联系方式**：第一个模型需要用户填写联系方式，后面的模型系统自动复制第一个模型的联系方式（可修改）其中，公司/个人、所属行业、应用场景、应用设备、功能描述、邮箱地址、联系方式在第一个模型中的填写信息会重复使用，后面创建的模型不用重复填写，但可修改信息

2. 上传测试集：包括填写测试集名称、上传语音文件、上传标注文件

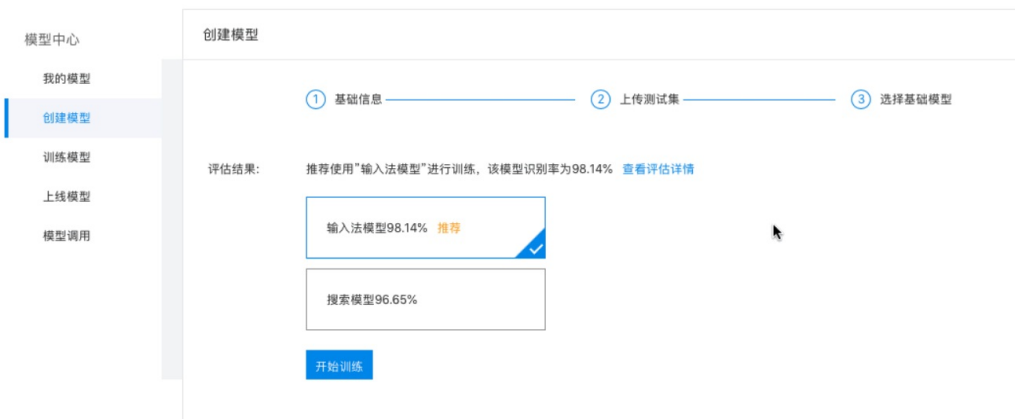
- **上传测试集**：用户可自行填写测试集名称，可支持中文、英文、数字、下划线.+#\*()^-
- **上传语音文件**：上传音频压缩zip文件（**请将所有音频文件直接压缩，请勿将音频存放在文件夹内再压缩**），格式要求：
  - 16k 16bit单声道pcm/wav文件
  - 8k 16bit 单声道pcm/wav文件（客服场景）；
  - 音频文件名请不要包含中文、特殊符号、空格等字符；
  - 所有音频需直接打包压缩为zip文件格式后上传，zip大小不超过100M，解压后单个音频大小不超过150M
- **上传标注文件**：上传音频的标注文本txt文件，格式要求：
  - 标注文件内容应与音频文件相对应的内容一致(单条音频对应文本长度不超过5000字)；
  - 标注文件格式应为txt格式，**GBK编码**；

- 标注txt文本中，由音频名称、标注内容两部分构成，用"tab"区隔，带后缀或不带后缀均可；

上传完语音文件及标注文件，点击【开始评估】，后台进入评估状态，此时弹窗提示评估完毕时间，并自动跳转回【我的模型】。一个账号只能同时评估一个模型。待模型评估完毕后通过【我的模型】可以点击进入“选择基础模型”



3. **选择基础模型**：系统根据基础模型的识别率自动推荐适合训练的基础模型，基础模型识别率超过50%才可选择进行训练。



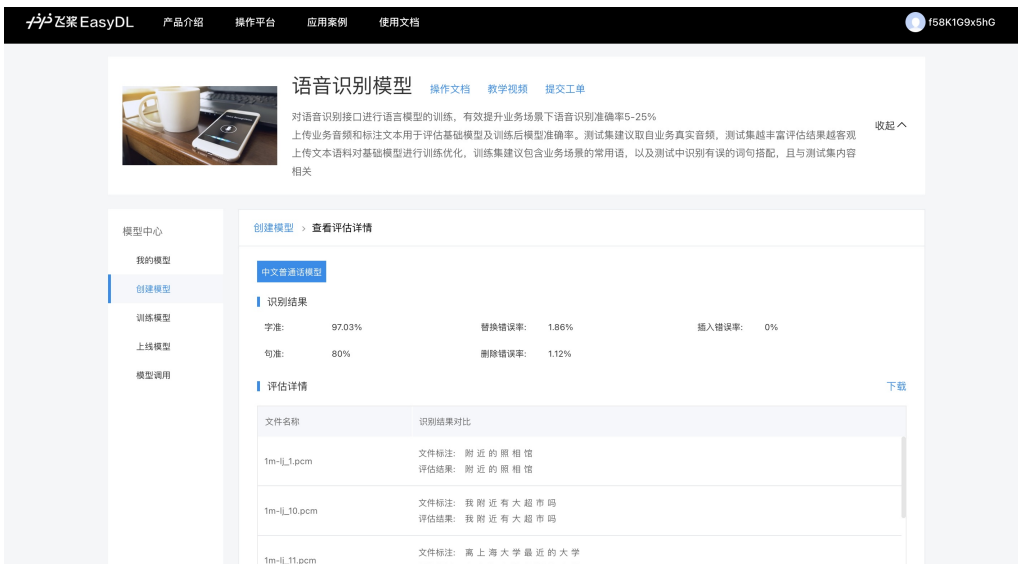
若基础模型识别率未达到50%，请检查语音文件和标注文件内容是否匹配，若不匹配，训练结果无意义。若检查标注文件无误后识别率仍旧过低，可以加入官方QQ群进行咨询：686267521

- 短语音识别产品类型中目前支持对**短语音识别极速版**进行训练
- 实时语音识别产品类型中目前支持对**实时语音识别**的中文普通话模型进行训练
- 呼叫中心产品类型中目前支持对**呼叫中心语音解决方案**进行训练

选择基础模型后点击【开始训练】即可在该模型上进行模型训练。

点击“查看评估详情”可以查看测试集在基础模型上的具体识别结果，评估详情包括：字准确率，句准确率，插入错误，删除错误，替换错误5个指标，以及在该测试集上的具体识别结果与标注结果的对比，根据识别错误信息可以更加精准地准备训练文本。





在“查看评估详情”页点击“返回上一步”或“创建模型”可返回选择基础模型

## 训练模型

可以在【创建模型】-“选择基础模型”页点击【开始训练】按钮进入【训练模型】

也可以在【我的模型】列表页选择已创建完成的模型点击操作栏中的“开始训练”进入【训练模型】；

也可以直接在左侧导航栏中点击【训练模型】，进入【训练模型】

在训练模型步骤中，选择需要训练的模型，并上传训练文本。目前有两种训练方式可以选择，可以上传热词，或者是长段文本，也可以两种均上传进行训练。



**热词文本格式要求：**热词训练支持上传热词txt文件进行训练，每个词之间需要换行，txt格式要求gbk编码，大小不超过5M

**句篇文本格式要求：**句篇训练支持上传多行单句或一整段篇章（一段文字且需要符号）txt文件进行训练，txt格式要求gbk编码，大小不超过5M

建议您上传与您所需模型内容相关度较高的文本或关键词，以便最大程度提高您的模型识别率

上传训练文本成功之后点击【开始训练】，后台进入模型训练状态，此时弹窗提示评估完毕时间，并自动跳转回【我的模型】。一个账号下同时只能训练一个模型。待模型训练完毕后生成新的模型版本，在【我的模型】列表页可以查看模型训练结果。



在【我的模型】列表，

模型ID	模型名称	当前版本	训练状态	基础模型准确率及模型效果	操作
1019	qwert	V1	训练完成	输入法模型:98.14% 当前版本:99.26%	<a href="#">历史版本</a> <a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">下载</a> <a href="#">删除</a>

[训练结果详情](#)

可以查看基础模型的识别率，和当前版本的识别率，了解训练提升效果

- **训练结果详情**：可以查看训练后模型在测试集上的识别详情，包括：字准确率，句准确率，插入错误，删除错误，替换错误5个指标，以及在测试集上的具体识别详情。可以进行操作
  - **历史版本**：可以查看历史训练的所有记录并进行操作
  - **申请上线**：对当前模型训练结果较为满意，可以点击申请上线，跳转至上线模型步骤
  - **迭代训练**：当前模型训练结果不满意，可以在当前版本基础上或者基础模型上继续添加新的训练语料，进行迭代训练获得新的模型版本
  - **下载**：可以下载评估模型上传的测试集和训练模型的训练集
  - **删除**：可以删除整个模型（包括所有历史版本），删除后不能恢复

## 上线模型

可以在【我的模型】选择要上线的模型，在操作栏点击“申请上线”

或者在左侧导航栏中点击【上线模型】，选择要上线的模型和版本进行上线（只有模型训练成功生成版本号才可上线）

当前版本	训练状态	基础模型准确率	当前版本准确率	操作
V4	训练完成	87.84%	92.37%	<a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">下载</a> <a href="#">训练结果详情</a>
V3	训练完成	87.84%	92.37%	<a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">下载</a> <a href="#">训练结果详情</a>
V2	训练完成	87.84%	92.46%	<a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">下载</a> <a href="#">训练结果详情</a>
V1	训练完成	87.84%	92.55%	<a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">下载</a> <a href="#">训练结果详情</a>

一个账号下最多只能上线3个模型。申请上线后需要后台管理员进行审核，1-3天内会有审核结果，可在【我的模型】中查看审核状态。若对审核过程有任何问题可以加入官方QQ群（群号：686267521）咨询群管。

- **审核中**：可以查看历史版本训练情况；可以取消申请，取消后方可继续训练

2529	模型名称1	V2	上线审核中 ?	搜索模型: 75.66% 当前版本: 75.66% <a href="#">训练结果详情</a>	<a href="#">历史版本</a> <a href="#">取消申请</a>
------	-------	----	---------	--	---

- **审核失败**：问号可查看审核失败原因；可以查看历史版本训练情况；可以迭代训练或重新训练；可以重新申请上线

2530	模型名称模型名...	V2	上线审核失败 ?	输入法模型: 75.66% 当前版本: 75.66% <a href="#">训练结果详情</a>	<a href="#">历史版本</a> <a href="#">申请上线</a> <a href="#">迭代训练</a> <a href="#">重新训练</a> <a href="#">下载</a> <a href="#">删除</a>
------	------------	----	----------	---	---

- **审核通过**：审核通过则后台自动上线，上线时间需要1-3天，上线过程中模型不可以做任何操作

2580	模型名称3	V2	上线中	输入法模型: 75.66% 当前版本: 75.66% <a href="#">训练结果详情</a>	正在上线中
------	-------	----	-----	---	-------

- **上线完成**：上线完成的模型可以正式调用

252925299	模型名称1	V2	已上线	输入法模型: 75.66% 当前版本: 75.66% <a href="#">训练结果详情</a>	<a href="#">历史版本</a> <a href="#">SDK下载</a> <a href="#">删除</a>
-----------	-------	----	-----	---	---

## 模型调用

上线通过的模型，在【我的模型】可以点击“模型调用”，查看如何使用模型

也可以在左侧导航栏中点击【模型调用】

中文普通话模型:87.18%

V1 已上线 实时语音识别 当前版本:90.67% [历史版本](#) [模型调用](#) [下载](#) [删除](#)

[训练结果详情](#)

模型中心

我的模型

创建模型

训练模型

上线模型

**模型调用**

模型调用

选择模型: 实时来一个

第一步: 创建语音技术应用(若已创建可直接使用), 获取鉴权参数AppID, API Key, Secret Key, [立即创建](#)

第二步: 获取专属模型参数 模型ID: 4690 基础模型pid: 1537

第三步: 根据业务情况, 选择合适的调用方式, 配置鉴权参数和专属模型参数即可使用。若您已经使用语音识别的服务, 只需要在您的接口中补充模型参数即可实现训练后的识别效果。

该基础模型支持在以下产品中使用: [实时语音识别](#), 该产品目前为邀测状态, 请点击下方合作咨询进行咨询业务开通

产品类型	基础模型	调用方式	鉴权参数	专属模型参数	操作
实时语音识别	中文普通话模型	WebSocket API	API Key, Secret Key	lm_id=4690, dev_pid=1537	<a href="#">技术文档</a>
		Android SDK			<a href="#">SDK下载</a> <a href="#">技术文档</a>
		iOS SDK	AppID, API Key, Secret Key	LM_ID=4690, PID=1537	<a href="#">SDK下载</a> <a href="#">技术文档</a>
		Linux SDK			<a href="#">SDK下载</a> <a href="#">技术文档</a>

选择您需要上线的模型 (训练完成的模型才可申请上线)

#### 短语音识别 :

step1 : 创建语音技术应用(若已创建可直接使用), 获取鉴权参数AppID, API Key, Secret Key。 [立即创建](#)

step2 : 获取专属模型参数 模型ID: xxxx 基础模型pid: xxxx

step3 : 配置鉴权参数和专属模型参数即可使用

短语音识别极速版支持API方式调用, 具体使用方法详见[技术文档](#)

#### 实时语音识别 :

step1 : 创建语音技术应用(若已创建可直接使用), 获取鉴权参数AppID, API Key, Secret Key。 [立即创建](#)

step2 : 获取专属模型参数 模型ID: xxxx 基础模型pid: xxxx

step3 : 根据业务情况, 选择合适的调用方式, 配置鉴权参数和专属模型参数即可使用

实时语音识别支持Websocket API, Android、iOS、Linux SDK方式调用 该接口目前处于邀测阶段, 权限开通请点击[合作咨询](#)或者加入官方QQ群: 588369236 获取技术文档和demo下载地址。

#### 呼叫中心模型-V1 :

step1 : 创建智能呼叫中心应用(若已创建可直接使用), 获取鉴权参数AppID, API Key, Secret Key。 [立即创建](#)

step2 : 获取专属模型参数 模型ID: xxxx

step3 : 下载SDK, 配置鉴权参数和专属模型参数即可使用

呼叫中心模型-V1支持C++ SDK、JAVA SDK、MRCP server三种调用方式, 下载百度智能呼叫中心SDK即可使用。 [立即下载](#) 具体使用方法详见[技术文档](#)

#### 呼叫中心模型-V2 :

step1 : 创建语音技术应用(若已创建可直接使用), 获取鉴权参数AppID, API Key, Secret Key。 [立即创建](#)

step2 : 获取专属模型参数 模型ID: xxxx

step3 : 下载SDK, 配置鉴权参数和专属模型参数即可使用



呼叫中心模型-V2支持MRCP server调用方式，具体使用方法详见[技术文档](#)

语音自训练平台公测期间，为了帮助客户验证线上效果，每个账号支持上线3个模型，每个账号累计有50000次免费调用量。正式商用后，免费资源可能会有所调整。公测期QPS限额：个人未认证账户2QPS；个人认证账户3QPS；企业认证账户5QPS。公测期间如需更多资源，欢迎[商务合作](#)咨询。

## 声音分类

### 声音分类整体说明

#### 什么是声音分类模型

声音分类是指可以定制识别出当前音频是哪一种声音，或者是什么状态/场景的声音。

EasyDL声音分类可以定制模型更多可以区分出不同物种发出的声音，如果希望定制声纹识别模型（如区分出当前音频是谁的声音），目前用EasyDL声音分类暂时无法解决。

目前声音分类使用EasyDL支持对最长15s左右的音频进行处理，在正式使用EasyDL声音分类模型之前，需要将已有的数据进行分段处理。



#### 声音分类的典型应用场景

- **安防监控**：定制识别不同的异常或正常的声音，进而用于突发状况预警。比如监控在工业生产场景中监控是否出现了异常噪音，从而辅助人工测试的时候判断是否出现bug。
- **科学研究**：定制识别同一物种的不同个体的声音、或者不同物种的声音，协助野外作业研究。比如动物研究机构从野外采集的声音，借助于EasyDL声音分类模型，判断当前音频属于什么物种。
- **其他**：尽情脑洞大开，训练你希望实现的声音分类模型。

#### 定制声音分类模型的整体流程

定制声音分类模型基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快几分钟即可获得定制模型。



#### 分析业务需求

这里我们详细介绍下，在使用EasyDL平台之前首先需要分析业务需求。这一步主要将实际业务需求转换为模型设计，在声音分类场景中，首先需要明确的问题为业务场景可能出现的全部声音类型有哪些？，这里很多企业开发者往往会主要关注业务场景中需要重点识别出的异常声音分类，而忽略了正常的声音也是一种分类。

以某服务商接到项目，需要判断出小区附近是否存在较大噪音为例，综合考虑小区附近可能有的声音类型，在这个场景需要定制声音分类模型能有效区分正常无噪音、正常噪音如救护车、警车声音、异常噪音，如汽车大声按喇叭等三类状态。那么在后续的准备数据阶段，也需要能有效准备这三类声音。



## 数据准备

### 创建数据集

在训练之前需要在数据中心【创建数据集】

The screenshot displays the EasyDL 'My Data Overview' page. On the left, a navigation menu includes 'Data Overview' (数据总览), which is highlighted. The main content area shows a table of datasets:

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	操作
V1	309243	0	● 已完成	音频分类	0% (0/0)	多人标注 导入 删除
V1	274020	885	● 已完成	音频分类	100% (885/885)	查看 多人标注 导入 标注 ...
V1	273870	171	● 已完成	音频分类	100% (171/171)	查看 多人标注 导入 标注 ...

The 'Create Dataset' form below includes:

- 数据集名称: 限制50个汉字内
- 数据类型: 音频
- 数据集版本: V1
- 标注类型: 音频分类
- 标注模板: 短音频单标签
- 完成按钮

On the right, a '常见问题' (FAQ) section lists five questions related to design categories, audio format requirements, and upload procedures.

### 设计分类

- 每个标签就是对这个音频希望识别出的全部结果。标签的上限为1000种。
- 标签名由数字、中英文、中/下划线组成，长度上限256字符。

### 音频的具体格式要求

- 训练集音频需要和实际场景要识别的音频环境一致，举例：如果实际场景要识别的音频都是手机摄录的，那训练的音频也需要同样的场景获得，而不要采用网上随便下载的音频。
- 每个标签的音频需要覆盖实际场景里面的可能性，如不同环境下，训练集覆盖的场景越多，模型的泛化能力越强。
- 如果需要寻求第三方数据采集团队协助数据采集，可以加入官方QQ群（群号:679517246）联系群管咨询了解。
- 音频支持mp3, m4a, wav格式，单个音频大小在4M内且时长小于15s。

### 上传数据集

#### 上传数据要求说明

这里我们对上传数据的要求不仅是格式上的要求，更重要的是介绍怎样的数据可以更有效提升模型效果

#### 设计分类

首先想好分类如何设计，每个分类为你希望识别出的一种结果，如要识别猫狗的叫声，则可以以“猫”、“狗”等分别作为一个分类；如果安防监控通过声音判断是否出现异常状态，可以以“正常”“不异常”设计为两类，或者“正常”“异常原因一”、“异常原因二”、“异常原因三”……设计为多类。

注意：目前单个模型的上限为1000类，如果要超过这个量级请在百度云控制台内[提交工单](#)反馈

### 准备数据

基于设计好的分类准备音频数据，每个分类需要准备50个音频文件以上，如果想要较好的效果，建议100个起音频文件，如果某些分类的声音具有相似性，需要增加更多音频。

音频的基本格式要求：目前支持音频文件类型为支持wav,mp3,m4a，音频文件大小限制在4M以内。一个模型的音频总量限制10万个音频文件。

注意1：训练集音频需要和实际场景要识别的音频环境一致，举例：如果实际场景要识别的声音都是手机采集的，那训练的音频文件也需要同样的场景获得，而不要采用网上随便下载的音频

注意2：考虑实际应用场景可能有的种种可能性，每个分类的音频需要覆盖实际场景里面可能有的可能性，如噪音干扰、多种可能的采集设备，训练集覆盖的场景越多，模型的泛化能力越强。

注意3：如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单](#)反馈

你可能会有的问题：如果训练音频数据无法全部覆盖实际场景要识别的音频，怎么办？

答：本身模型算法会有一定的泛化能力，尽可能覆盖即可。

### 导入未标注数据

#### 本地数据

##### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

上传压缩包 [↑ 上传压缩包](#)

已有数据集 支持选择百度云BOS导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他语音类模型的数据集

##### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式 

- 本地导入
- BOS目录导入**
- 分享链接导入
- 平台已有数据集

### 导入已标注数据

#### 本地数据

##### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式  

- [上传压缩包](#)
- API导入

已有数据集 支持选择百度云BOS导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他语音类模型的数据集

**导入数据**

数据标注状态

 无标注信息  有标注信息

导入方式

请选择 ^

- 本地导入
- BOS目录导入
- 分享链接导入
- 平台已有数据集

**数据管理API**

本文档主要说明当您线下已有大量的已经完成分类整理的音频数据，如何通过调用API完成音频数据的便捷上传和管理。

**数据集创建API****接口描述**

该接口可用于创建数据集。

**接口鉴权**

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

**请求说明****请求示例**

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/create>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

**请求参数**

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

**返回说明****返回参数**

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
dataset_id	否	number	创建的数据集ID

### 查看数据集列表API

#### 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态, 包括shared、smart和空值, 分别表示共享中、智能标注中、非特殊状态

### 查看分类 (标签) 列表API

#### 接口描述

该接口可用于查看分类 (标签)。返回分类 (标签) 的名称、包含数据量等信息。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法 : POST

请求URL : <https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下 :

参数	值
Content-Type	application/json

Body中放置请求参数, 参数详情如下 :

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型, 可包括: IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应: 图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号, 默认0
num	否	number	数量, 默认20, 最多100

若查看声音分类的全部分类, 在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API :

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
append_label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION时，填入图片/声音的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；文本分类10000个汉字</b>
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称（由数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

若上传声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 数据集删除API

##### 接口描述

该接口可用于删除数据集。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID

若删除声音分类数据集，在type参数应传「SOUND\_CLASSIFICATION」

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

#### 分类（标签）删除API

##### 接口描述

该接口可用于删除分类（标签）。

##### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

##### 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数



字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

若删除声音分类的子类，在type参数应传「SOUND\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

## 模型训练

### 🔗 声音分类训练操作说明

数据提交后，可以在导航中找到【训练模型】，按以下步骤操作，启动模型训练：

**注意：**启动训练前请确保数据已经标注完成，否则无法启动训练



### step1：选择模型

选择此次训练的模型。

### step2：训练配置

#### 部署方式

- 可选择「公有云部署」、「EasyEdge本地部署」

#### 选择设备

- 如果您选择了「EasyEdge本地部署」，请根据实际部署设备选择设备。

#### 选择算法

- 当前语音分类仅支持默认算法，点击可查看[算法性能及适配硬件](#)。

#### 高级训练配置

- 如果您选择了「EasyEdge本地部署」，选择【同步支持公有云部署】，训练完成后模型可部署到百度云上进行使用。

### step3：添加数据

- 先选择数据集，再按分类选择数据集里的音频，可从多个数据集选择音频
- 声音分类模型至少需要选择2个及以上分类

### step4：训练模型

点击「开始训练」，训练模型。

- 训练时间与数据量大小有关，1000个音频文件大约可以在30min内训练完成
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面，在模型训练完毕即可收到短信通知。

平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有声音分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

**注意：**如果遇到模型超过2天停留在训练中的状态，或者遇到训练失败的情况，请在百度云控制台内[提交工单](#)反馈

## 声音分类模型效果评估报告说明

### 模型评估报告内容说明

模型训练完成后我们可以在模型列表中看到模型效果及查看模型评估报告的入口。进入模型评估报告页面，我们可以看到整体报告内容中包含以下几个区域内容：

### 模型基本信息

在这个部分可以选择应用类型（声音分类目前仅支持云服务）、训练版本、相应版本提交的音频数量、相应版本提交的分类数量。

### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、准确率、F1-score、精确率、召回率。这部分模型效果的结果内容是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如音频数量低于100个），参与评估的数据可能不超过30个音频，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

**注意：**若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口测试批量音频数据获取更准确的模型效果。

### 详细评估

在这个部分可以看到上述训练效果背后的原始评估数据。以及不同top结果的准确率效果情况，下面为相关名词解释。



### 准确率

准确率含义为正确分类的样本数与总样本数之比，这里指的总样本是指从总训练数据中随机抽取部分数据参与模型评估的总样本，在上文截图中，参与训练的音频数200个，实际参与评估的音频为下面详细评估预测表现表格的数据总和，即50个。那么准确率为参与评估的正确数量46/50，结果为92.0%

### F1-Score

F1-score是指对某类别而言为精确率和召回率的调和平均数，此处为各类别F1-score的平均数。就某类而言，精确率和召回率体现了该分类的精确率及召回率的平衡情况：

$$F_1 = 2 \cdot \frac{\textit{precision} \cdot \textit{recall}}{\textit{precision} + \textit{recall}}$$

### 精确率

精确率是对某类别而言为正确预测为该类别的样本数与预测为该类别的总样本数之比，此处为各类别精确率的平均数。如果精确率比较低，有一定可能因为不同类别效果存在差异所致，请检查下不同类型样本量是否均衡。

### 召回率

召回率是指对某类别而言为正确预测为该类别的样本数与该类别的总样本数之比，此处为各类别召回率的平均数。

### top1、top2、top...5

是指对于每一个评估的音频文件，模型会根据置信度高低，依次给出top1-top5的识别结果，其中top1置信度最高，top5的置信度最低。那么top1的准确率值是指对于评估标准为“top1结果识别为正确时，判定为正确”给出准确率。top2准确率值是指对于评估标准为“top1或者top2只要有一个命中正确的结果，即判定为正确”给出的准确率。……以此类推。

### 如何解读模型效果

在看模型评估报告结果中，首要需要关注下详细评估中的预测表现，这里可以看到所有评估报告的数据是基于什么量级进行计算的。当整体参与评估的数量较少时，所有数值可能无法真实反映模型效果。

#### 评估样本具体数据情况

随机测试集	
预测表现	正确数量：217
	错误数量：3

在查看模型评估结果可能需要思考在当前业务场景精确率与召回率更关注哪个指标，是更希望减少误识别，还是更希望减少误召回。前者更需要关注召回率的指标，后者更需要关注精确率的指标。同时F1-SCORE可以有效关注精确率和召回率的平衡情况，对于希望召回与识别效果兼具的场景，F1-Score越接近1效果越好。

### 🔗 声音分类训练时长说明

训练时长与数据量、所选算法紧密相关。

目前声音分类的训练时长主要影响因素为数据量，以下为内部测试的数据量与训练时长的对应关系，供参考：

数据量	训练时长
数十个音频	60min左右
数百个音频	90min左右
数千个音频	120min左右
数万个音频	150min以上

### 🔗 如何提升模型效果

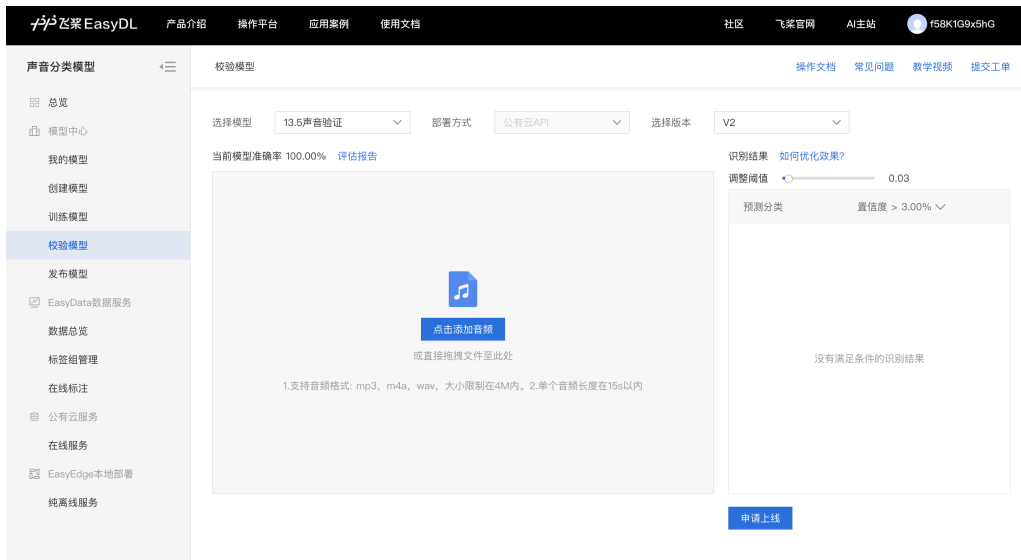
#### 如何充分测试模型效果

##### 模型校验

在查看模型评估报告基础上，首先使用模型校验功能测试未参与过训练的音频数据进行模型训练，在这一步尽量上传不同类别的数据充分测试，并在测试过程中线下记录识别错误的音频。在测试过程中需要关注以下内容：

1. 不同分类的准确率是否存在明显差异
2. 识别错误的音频是否存在一些共性？比如设备相似、音调相似、环境相似等等

### 3. 识别错误的音频人耳是否能明显分辨



#### 发布模型为API通过调用接口充分测试

将声音分类模型发布为API后，调用接口进行批量测试，在测试过程中同样重点关注上述三点内容。

#### 如何提升模型效果

在充分测试模型效果基础上，如果发现模型效果欠佳，建议根据以下顺序分析并提升模型效果。

#### 检查并优化训练数据

首先检查目前欠佳的模型是否存在训练数据过少的情况，建议每个类别的音频量不少于200个，如果低于这个量级建议扩充。

在扩充数据中需要一并检查不同类别的数据量是否均衡，建议不同分类的数据量级相同，并尽量接近，如果有的类别数据量很高，有的类别数据量较低，那么可能会存在不同类别的准确率不同，同时低准确率的分类会拉低整体模型效果。

另外需要检查测试模型的音频数据与训练数据采集来源是否一致，如果设备不一致、或者采集的环境不一致（录音室环境及实际生产环境的差异），那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致

最后也请确认识别错误的音频人耳是否能清晰分辨，模型效果很难超越人耳的识别精度效果，这种情况，请在百度云控制台内[提交工单](#)反馈。

#### 联系EasyDL团队提升模型效果

在完成上述检查并优化训练数据的工作后，仍然发现了显著的模型效果低的情况，请在百度云控制台内[提交工单](#)反馈。

## 模型发布

### 🔗 声音分类模型发布整体说明

训练完成后，可将模型部署在公有云服务器，发布成为在线的restful API接口，参考示例文档通过HTTP请求的方式进行调用。或将模型通过服务器端SDK部署在私有服务器，

#### 公有云API

将模型发布为API后，将获得一部分免费调用次数，超出免费额度将根据调用次数进行收费。

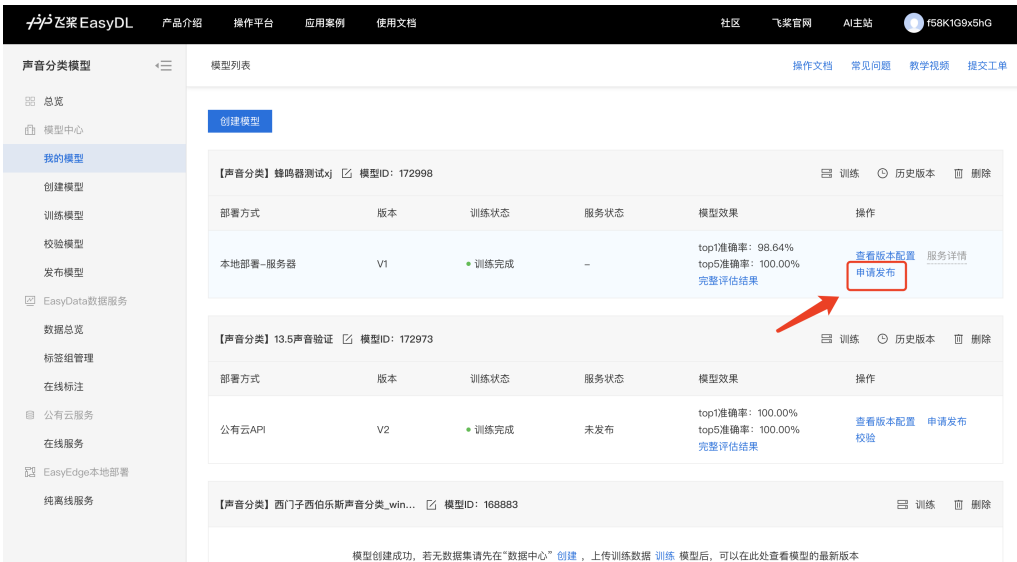
#### 私有服务器部署

可将训练完成的模型部署在私有CPU/GPU服务器上，在内网/无网环境下使用模型，确保数据隐私。

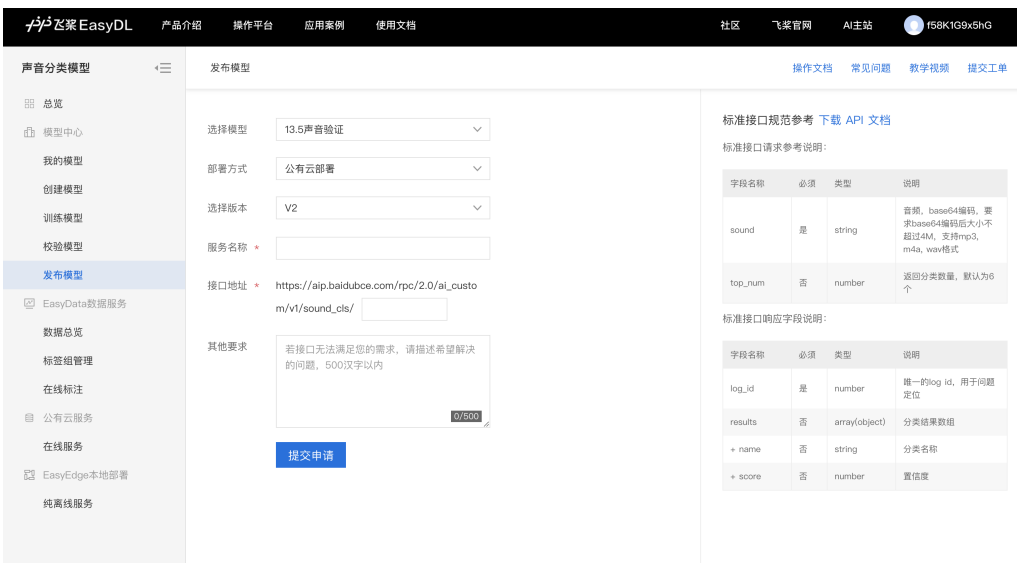
### 🔗 公有云部署

### 🔗 声音分类模型如何发布为API

声音分类模型训练完毕后就可以在左侧目录栏中找到【发布模型】，发布模型表单页面需要自定义接口地址后缀及服务名称，即可申请发布。



申请发布后，通常的审核周期为T+0，即申请当天可以审核完成，如遇到周末及其他法定假日，将顺延至节后第一天完成审核，如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。



模型发布成功后，可以在模型列表跟踪审核状态，在审核通过后，审核发布中预计等待15分钟左右时间，API将完成上线，并在模型列表相应模型位置查看到【服务详情】入口，点击可查看接口地址，调用API请参考[API调用文档](#)。

🔗 声音分类API调用文档

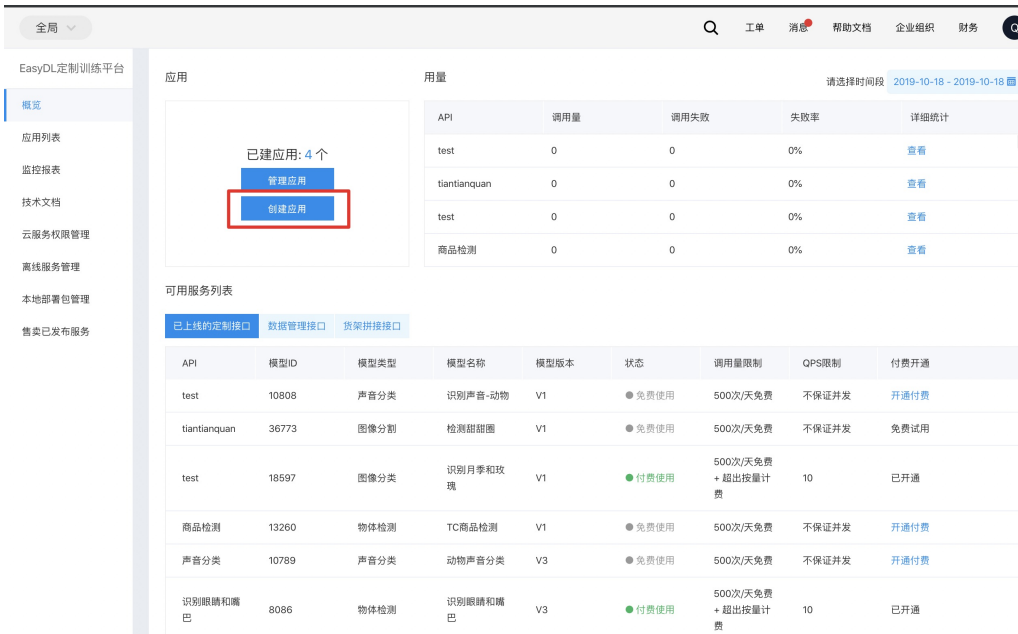
**接口描述**

声音分类模型完成模型发布后可以获得定制的声音分类API，实现基于定制的声音分类模型，调用API输入15s以内的音频数据，返回自定义的分类结果。

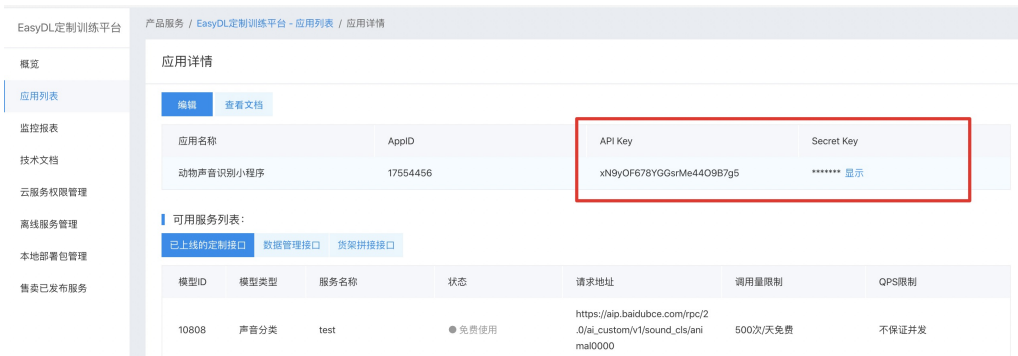
**接口鉴权**

**创建应用**

进入[EasyDL控制台](#)概览页，在已上线的定制接口下，可以看到已发布上线的声音分类API。在正式调用之前，首先点击[创建应用](#)，定义应用名称、应用类型、应用描述等信息，完成应用创建。



应用创建后，在应用详情页获取API Key、Secret Key。

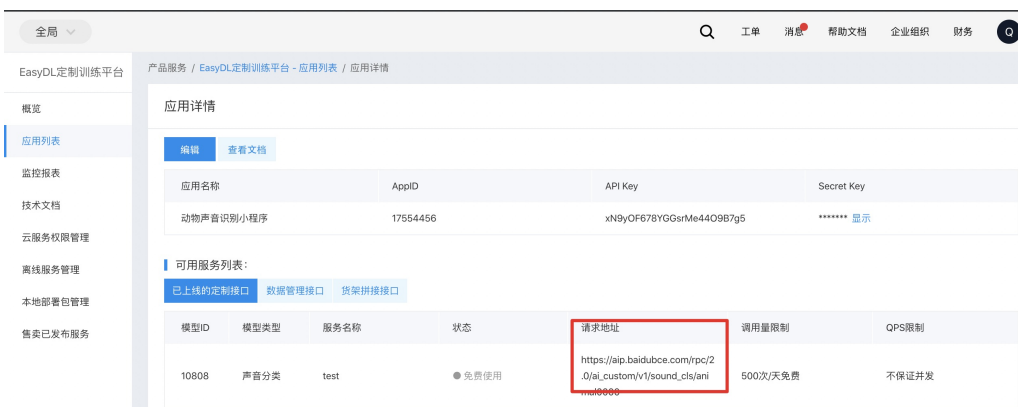


### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请在应用详情页面获取接口地址，与发布模型时定义的url一致。



URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json



Body请求示例：

```
{
  "sound": "<base64数据>",
  "top_num": 6
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
sound	是	string	-	音频，base64编码，要求base64编码后大小不超过4M，支持mp3、m4a、wav格式，单个文件时间长度不超过15s 注意需要去掉编码头后再进行urlencode。
top_num	否	number	-	返回分类数量，默认为6个

#### 请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、音频地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

#### PHP代码示例

```
<?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = "", $param = "")
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_HEADER, 0);
    // 要求结果为字符串且输出到屏幕上
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, 1);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    // post提交方式
    curl_setopt($curl, CURLOPT_POST, 1);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
    // 运行curl
    $data = curl_exec($curl);
    curl_close($curl);

    return $data;
}

$token = '[调用鉴权接口获取的token]';
$url = '[接口地址]?access_token=' . $token;
$bodys = '{"sound": "\sfasq35sadvsvqr5q..."}'
$res = request_post($url, $bodys);

var_dump($res);
```

#### Python3代码示例

```

"""
EasyDL 声音分类 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
    pip freeze | grep requests
若返回值为空，则安装该库
    pip install requests
"""

**目标音频的 本地文件路径，支持mp3/m4a/wav格式**
SOUND_FILEPATH = "【您的测试音频地址，例如./example.mp3】 "

**可选的请求参数**
**top_num: 返回的分类数量，不声明的话默认为 6 个**
PARAMS = {"top_num": 3}

**服务详情 中的 接口地址**
MODEL_API_URL = "【您的API地址】 "

**调用 API 需要 ACCESS_TOKEN。若已有 ACCESS_TOKEN 则于下方填入该字符串**
**否则，留空 ACCESS_TOKEN，于下方填入 该模型部署的 API_KEY 以及 SECRET_KEY，会自动申请并显示新 ACCESS_TOKEN**
ACCESS_TOKEN = "【您的ACCESS_TOKEN】 "
API_KEY = "【您的API_KEY】 "
SECRET_KEY = "【您的SECRET_KEY】 "

print("1. 读取目标音频 '{}'.format(SOUND_FILEPATH)")
with open(SOUND_FILEPATH, 'rb') as f:
    base64_data = base64.b64encode(f.read())
    base64_str = base64_data.decode('UTF8')
print("将 BASE64 编码后音频的字符串填入 PARAMS 的 'sound' 字段")
PARAMS["sound"] = base64_str

if not ACCESS_TOKEN:
    print("2. ACCESS_TOKEN 为空，调用鉴权接口获取TOKEN")
    auth_url = "https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials\
        "&client_id={}&client_secret={}".format(API_KEY, SECRET_KEY)
    auth_resp = requests.get(auth_url)
    auth_resp_json = auth_resp.json()
    ACCESS_TOKEN = auth_resp_json["access_token"]
    print("新 ACCESS_TOKEN: {}".format(ACCESS_TOKEN))
else:
    print("2. 使用已有 ACCESS_TOKEN")

print("3. 向模型接口 'MODEL_API_URL' 发送请求")
request_url = "{}?access_token={}".format(MODEL_API_URL, ACCESS_TOKEN)
response = requests.post(url=request_url, json=PARAMS)
response_json = response.json()
response_str = json.dumps(response_json, indent=4, ensure_ascii=False)
print("结果:\n{}".format(response_str))

```

#### JAVA代码示例

```
package com.baidu.ai.aip;

import com.baidu.ai.aip.utils.HttpUtil;
import com.baidu.ai.aip.utils.GsonUtils;

import java.util.*;

/**
 * easydl声音分类
 */
public class EasydlSoundClassify {

    /**
     * 重要提示代码中所需工具类
     * FileUtil,Base64Util,HttpUtil,GsonUtils请从
     * https://ai.baidu.com/file/658A35ABAB2D404FBF903F64D47C1F72
     * https://ai.baidu.com/file/C8D81F3301E24D2892968F09AE1AD6E2
     * https://ai.baidu.com/file/544D677F5D4E4F17B4122FBD60DB82B3
     * https://ai.baidu.com/file/470B3ACCA3FE43788B5A963BF0B625F3
     * 下载
     */
    public static String easydlSoundClassify() {
        // 请求url
        String url = "【接口地址】";
        try {
            Map<String, Object> map = new HashMap<>();
            map.put("sound", "sfasq35sadvsvqwr5q...");

            String param = GsonUtils.toJson(map);

            // 注意这里仅为了简化编码每一次请求都去获取access_token，线上环境access_token有过期时间，客户端可自行缓存，过期后重新获取。
            String accessToken = "[调用鉴权接口获取的token]";

            String result = HttpUtil.post(url, accessToken, "application/json", param);
            System.out.println(result);
            return result;
        } catch (Exception e) {
            e.printStackTrace();
        }
        return null;
    }

    public static void main(String[] args) {
        EasydlSoundClassify.easydlSoundClassify();
    }
}
```

#### C++代码示例

```

**include <iostream>**
**include <curl/curl.h>**

// libcurl库下载链接：https://curl.haxx.se/download.html
// jsoncpp库下载链接：https://github.com/open-source-parsers/jsoncpp/
const static std::string request_url = "【接口地址】";
static std::string easydlSoundClassify_result;
/**
 * curl发送http请求调用的回调函数，回调函数中对返回的json格式的body进行了解析，解析结果储存在全局的静态变量当中
 * @param 参数定义见libcurl文档
 * @return 返回值定义见libcurl文档
 */
static size_t callback(void *ptr, size_t size, size_t nmemb, void *stream) {
    // 获取到的body存放在ptr中，先将其转换为string格式
    easydlSoundClassify_result = std::string((char *) ptr, size * nmemb);
    return size * nmemb;
}
/**
 * easydl声音分类
 * @return 调用成功返回0，发生错误返回其他错误码
 */
int easydlSoundClassify(std::string &json_result, const std::string &access_token) {
    std::string url = request_url + "?access_token=" + access_token;
    CURL *curl = NULL;
    CURLcode result_code;
    int is_success;
    curl = curl_easy_init();
    if (curl) {
        curl_easy_setopt(curl, CURLOPT_URL, url.data());
        curl_easy_setopt(curl, CURLOPT_POST, 1);
        curl_slist *headers = NULL;
        headers = curl_slist_append(headers, "Content-Type:application/json;charset=UTF-8");
        curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDS, "{\"sound\":\"sfasq35sadvsvqwr5q...\"}");
        result_code = curl_easy_perform(curl);
        if (result_code != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n",
                curl_easy_strerror(result_code));
            is_success = 1;
            return is_success;
        }
        json_result = easydlSoundClassify_result;
        curl_easy_cleanup(curl);
        is_success = 0;
    } else {
        fprintf(stderr, "curl_easy_init() failed.");
        is_success = 1;
    }
    return is_success;
}

```

### 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

## 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	音频大小超出限制，请根据接口文档检查入参格式，音频文件大小应控制在4M以内，有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336005	音频解码失败	音频解码失败失败，音频文件非所限定的mp3, m4a, wav格式
336006	缺失必要参数	未上传音频文件，请补充必要参数后重新请求
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

☞ 私有服务器部署

☞ LinuxSDK集成文档

## 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法，适用于 EasyDL 和 BML。

EasyDL 通用版：

- 网络类型支持：图像分类，物体检测，图像分割，声音分类，表格预测
- 硬件支持：
  - Linux x86\_64 CPU (基础版，加速版)
  - Linux x86\_64 Nvidia GPU (基础版，加速版)
- 语言支持：Python 3.5, 3.6, 3.7

BML：

- 网络类型支持：图像分类，物体检测，声音分类
- 硬件支持：
  - Linux x86\_64 CPU (基础版)
  - Linux x86\_64 Nvidia GPU (基础版)
- 语言支持：Python 3.5, 3.6, 3.7

Release Notes

时间	版本	说明
2022.10.27	1.3.5	新增华为Atlas300、飞腾Atlas300 Python SDK，支持图像分类、物体检测、人脸检测、实例分割
2022.09.15	1.3.3	EasyDL CPU普通版新增支持表格预测
2022.05.27	1.3.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2021.12.22	1.2.7	声音分类模型升级
2021.10.20	1.2.6	CPU基础版、CPU加速版、GPU基础版推理引擎优化升级
2021.08.19	1.2.5	CPU基础版、CPU无损加速版、GPU基础版新增支持EasyDL小目标检测
2021.06.29	1.2.4	CPU、GPU新增EasyDL目标跟踪支持；新增http server服务启动demo
2021.03.09	1.2.2	EasyDL CPU加速版新增支持分类、高性能检测和均衡检测的量化压缩模型
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.03.12	1.1.14	新增声音识别python sdk
2020.02.12	1.1.13	新增口罩模型支持
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持 SDK 加速版
2019.12.04	1.1.10	支持图像分割
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.08.29	1.1.8	CPU 加速版支持
2019.07.19	1.1.7	提供模型更新工具
2019.05.16	1.1.3	NVIDIA GPU 支持
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】序列号的配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。
- 使用声音分类SDK需要安装额外依赖 \* pip 安装 `resampy pydub six librosa` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已基于sdk中无需额外安装，linux系统需要手动安装）
- 使用表格预测SDK需要安装额外依赖 `pip安装 brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3 catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2 psutil==5.7.2 pyppmm==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0` 另外ml算法安装（目前只支持python3.7） `pip install BaiduAI_TabularInfer-0.0.0-cp37-cp37m-linux_x86_64.whl` 安装 **paddlepaddle**
- 使用x86\_64 CPU 基础版 预测时必须安装（目标跟踪除外）：

```
python -m pip install paddlepaddle==2.2.2 -i https://mirror.baidu.com/pypi/simple
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU 基础版预测时必须安装（目标跟踪除外）：

```
python -m pip install paddlepaddle-gpu==2.2.2.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA10.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2 -i https://mirror.baidu.com/pypi/simple #CUDA10.2的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post110 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.0的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post111 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post112 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.2的PaddlePaddle
```

不同cuda版本的环境，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持安装 OpenVINO 使用x86\_64 CPU 加速版 SDK 预测时必须安装。

1) 请参考 [OpenVINO toolkit 文档](#)安装 2021.4版本, 安装时可忽略Configure the Model Optimizer及后续部分

2) 运行之前，务必设置环境变量

```
source /opt/intel/openvino_2021/bin/setupvars.sh
```

安装 cuda、cudnn

- 使用Nvidia GPU 加速版预测时必须安装。依赖的版本为 cuda9.0、cudnn7。版本号必须正确。

安装 pytorch (torch >= 1.7.0)

- 目标跟踪模型的预测必须安装pytorch版本1.7.0及以上（包含：Nvidia GPU 基础版、x86\_64 CPU 基础版）。
- 目标跟踪模型Nvidia GPU 基础版 还需安装依赖cuda、cudnn。

关于不同版本的pytorch和CUDA版本的对应关系：[pytorch官网](#) 目标跟踪模型还有一些列举在requirements.txt里的依赖（包括torch >= 1.7.0），均可使用pip下载安装。

```
pip3 install -r requirements.txt
```

2. 安装 easyedge python wheel 包 安装说明

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。安装说明：[华为 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Atlas300-{版本号}-cp36-cp36m-linux_x86_64.whl
```

安装说明：[飞腾 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Phytium.Atlas-{版本号}-cp36-cp36m-linux_aarch64.whl
```

3. 使用序列号激活



**纯离线服务说明**

发布纯离线服务，将训练完成的模型部署在本地，离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体方案专项适配硬件上。通过API、SDK进一步集成，灵活适应不同业务场景。

[发布并服务](#) [控制台](#)

服务栈 通用小型设备 专项适配硬件

SDK API

## 获取序列号

此处发布、下载的SDK为未授权SDK，需要前往控制台[获取序列号](#)激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标test	134318-V1 <a href="#">查看性能报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		英特尔GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
		基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>	

## 修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

## 4. GPU 加速版 使用 GPU 加速版，在安装完 whl 之后，必须：

1. 从[这里](#)下载 TensorRT7.0.0.11 for cuda9.0，并把解压后的 lib 放到 C++ SDK 的 lib 目录或系统 lib 目录
2. 运行时，必须在系统库路径中包含 C++ SDK 下的lib目录。如设置LD\_LIBRARY\_PATH

```
cd ${SDK_ROOT}
```

### \*\*1. 安装 python wheel 包\*\*

```
tar -xzf python/*.tar.gz
pip install -U {对应 Python 版本的 wheel 包}
```

### \*\*2. 设置 LD\_LIBRARY\_PATH\*\*

```
tar -xzf cpp/*.tar.gz
export EDGE_ROOT=$(readlink -f $(ls -h | grep "baidu_easyedge_linux_cpp"))
export LD_LIBRARY_PATH=$EDGE_ROOT/lib
```

### \*\*3. 运行 demo\*\*

```
python3 demo.py {RES文件夹路径} {测试图片路径}
```

如果是使用 C++ SDK 自带的编译安装的 OpenCV，LD\_LIBRARY\_PATH 还需要包括 C++ SDK 的 build 目录下的 `thirdparty/lib` 目录

如果没有正确设置 LD\_LIBRARY\_PATH，运行时可能报错：

```
ImportError: libeasyedge.so.0.4.3: cannot open shared object file: No such file or directory
ImportError: libopencv_core.so.3.4: cannot open shared object file: No such file or directory
```

## 5. 测试 Demo

### 5.1 图片预测

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



### 5.2 视频预测（适用于目标跟踪）

输入对应的模型文件夹（默认为RES）和测试视频文件路径 / 摄像头id / 网络视频流地址，运行：

```

**video_type: 输入源类型 type:int**
**1 本地视频文件**
**2 摄像头的index**
**3 网络视频流**
**video_src: 输入源地址, 如视频文件路径、摄像头index、网络流地址 type: string**
python3 demo.py {model_dir} {video_type} {video_src}

```

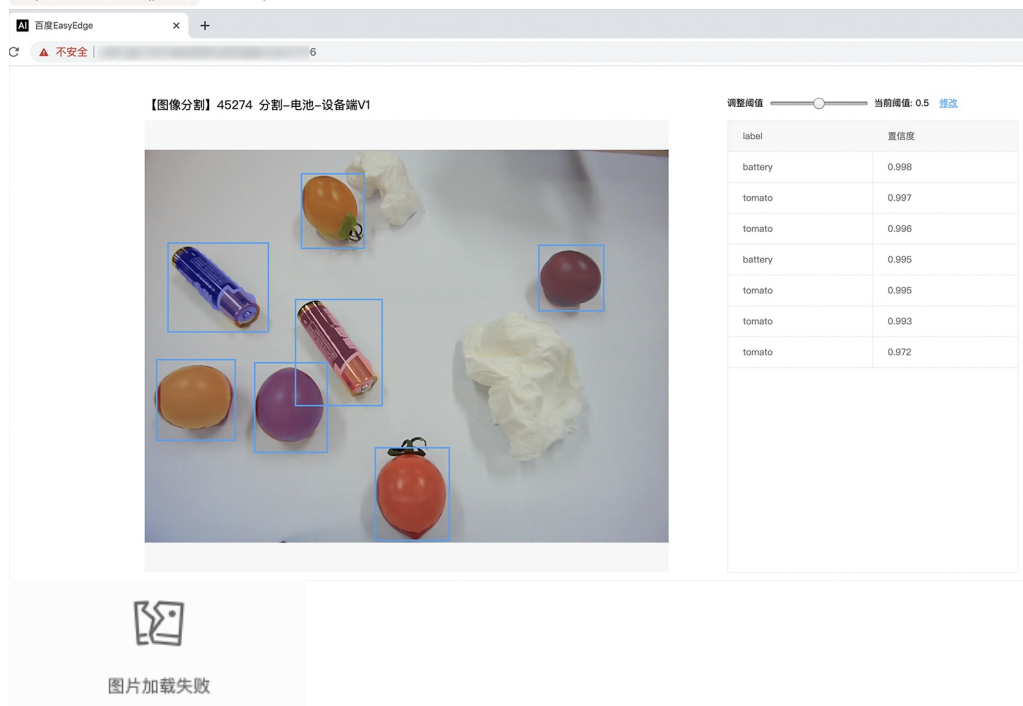
6. 测试Demo HTTP 服务 输入对应的模型文件夹（默认为RES）、序列号、设备ip和指定端口号，运行：

```
python3 demo_serving.py {model_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

后，会显示：

```
Running on http://0.0.0.0:24401/
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里 `http_client_test()`函数请求http服务进行推理。



## 使用说明

使用流程 `demo.py`

```

import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir={RES文件夹路径}, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image((numpy.ndarray的图片))
pred.close()

```

`demo_serving.py`

```

import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir={RES文件夹路径}, license=serial_key)
**请参考同级目录下demo.py里:**
**pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)**
**对以下参数device\device_id和engine进行修改**
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)

```

## 初始化

- 接口

```
def init(self,
    model_dir,
    device=Device.CPU,
    engine=Engine.PADDLE_FLUID,
    config_file='conf.json',
    preprocess_file='preprocess_args.json',
    model_file='model',
    params_file='params',
    label_file='label_list.txt',
    infer_cfg_file='infer_cfg.json',
    device_id=0,
    thread_num=1
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device, 比如: Device.CPU
        engine: BaiduAI.EasyEdge.Engine, 比如: Engine.PADDLE_FLUID
        config_file: str
        preprocess_file: str
        model_file: str
        params_file: str
        label_file: str 标签文件
        infer_cfg_file: 包含预处理、后处理信息的文件
    device_id: int 设备ID
        thread_num: int CPU的线程数

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success
    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

## 预测图像

- 接口

```
def infer_image(self, img,
                threshold=0.3,
                channel_order='HWC',
                color_format='BGR',
                data_type='numpy'):
    """

    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测, 矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测, 矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

### 预测视频（目前仅限目标跟踪模型调用）

- 接口

```
def infer_frame(self, frame, threshold=None):
    """
    视频推理(抽帧之后)
    :param frame:
    :param threshold:
    :return:
    """
```

- 返回格式dict

字段	类型	说明
pos	dict1	当前帧每一个类别的追踪目标的像素坐标(tlwh)
id	dict2	当前帧每一个类别的追踪目标的id
score	dict3	当前帧每一个类别的追踪目标的识别置信度
label	dict4	class_idx(int)与label(string)的对应关系
class_num	int	追踪类别数

### 预测声音

- 使用声音分类SDK需要安装额外依赖 `pip 安装 resampy pydub` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已集成在sdk中无需额外安装，linux系统需要手动安装）

- 接口

```
def infer_sound(self, sound_binary,
                threshold=0.3):
    """
    Args:
        sound_binary: sound_binary
        threshold: confidence
    Returns:
        list
    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类的置信度
label	string		分类的类别
index	number		分类的类别

**升级模型** 适用于经典版升级模型，执行`bash update_model.sh`，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

## FAQ

**Q: EasyDL 离线 SDK 与云服务效果不一致，如何处理？** A: 后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**Q: 运行时报错 "非法指令" 或 "illegal instruction"** A: 可能是 CPU 缺少 `avx` 指令集支持，请在[paddle官网](#) 下载 `noavx` 版本覆盖安装

**Q: NVIDIA GPU预测时，报错显存不足：** A: 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过`export FLAGS_fraction_of_gpu_memory_to_use=0.3`来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 `allocate` 更多的显存。

**Q: 我想使用多线程预测，怎么做？** 如果需要多线程预测，可以每个线程启动一个Program实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的`thread_num`参数。

## Q: 运行SDK报错 Authorization failed

**情况一：日志显示 `Http perform failed: null respond`** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受`HTTP_PROXY` 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示`failed to get/check device id(xxx)`或者`Device fingerprint mismatch(xxx)`** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

情况三：Atlas Python SDK日志提示 `ImportError: libavformat.so.58: cannot open shared object file: No such file or directory` 或者其他类似so找不到 可以在 `LD_LIBRARY_PATH` 环境变量加上 `libs` 和 `thirdpartylibs` 路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内 `libs` 和 `thirdpartylibs` 路径的方法如下(以华为Atlas300 SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas300 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## Windows SDK集成文档

### 简介

本文档介绍声音分类服务器端Windows SDK的使用方法。

- 操作系统支持
  - 64位 Windows 7 及以上
  - 64位Windows Server 2012及以上
- 环境依赖 (必须安装以下版本)
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级，下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 优化模型算法；GPU CUDA9.0 CUDA10.0 标记为待废弃状态 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | GPU基础版推理引擎优化升级；GPU加速版支持自定义模型文件缓存路径；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | 修复已知问题 | | 2021-08-19 | 1.3.2 | 新增支持EasyDL小目标检测，新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | 修复已知问题 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020-12-18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020-10-29 | 1.1.19 | 修复已知问题 | | 2020-09-17 | 1.1.18 | 支持更多模型 | | 2020.08.11 | 1.1.17 | 支持专业版更多模型 | | 2020.06.23 | 1.1.16 | 支持专业版更多模型 | | 2020.05.15 | 1.1.15 | 更新加速版tsortr版本，支持高精度检测 | | 2020.03.13 | 1.1.14 | 支持声音分类 | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | 支持物体检测高精度算法的CPU加速版，EasyDL 专业版支持 SDK 加速版 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版！ |

### 快速开始

#### 1. 安装依赖

##### 安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

##### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

### 如果使用GPU版SDK，请安装CUDA + cuDNN

<https://developer.nvidia.com/cuda>

<https://developer.nvidia.com/cudnn>

### 声音分类依赖

#### · 安装six

打开cmd，进入sdk包所在目录。执行EasyEdge-win-mXXX-x86-nvidia-gpu\python37\python.exe -m pip install -U six -i <http://mirrors.aliyun.com/pypi/simple/> --trusted-host mirrors.aliyun.com

#### · 安装librosa

打开cmd，进入sdk包所在目录。执行EasyEdge-win-mXXX-x86-nvidia-gpu\python37\python.exe -m pip install -U librosa -i <http://mirrors.aliyun.com/pypi/simple/> --trusted-host mirrors.aliyun.com

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，SDK默认使用cuda9版本，如果需要cuda10请运行EasyEdge CUDA10.0.bat切换到cuda10版本，之后打开EasyEdge.exe，选

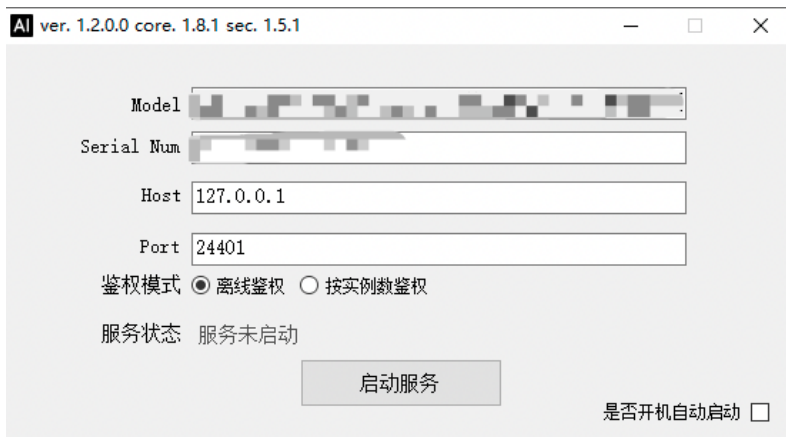
择鉴权模式，输入Serial Num  点击"启动服务"，等待数秒即可启动成功，本地服务默认运行在

图片加载失败

<http://127.0.0.1:24401/>

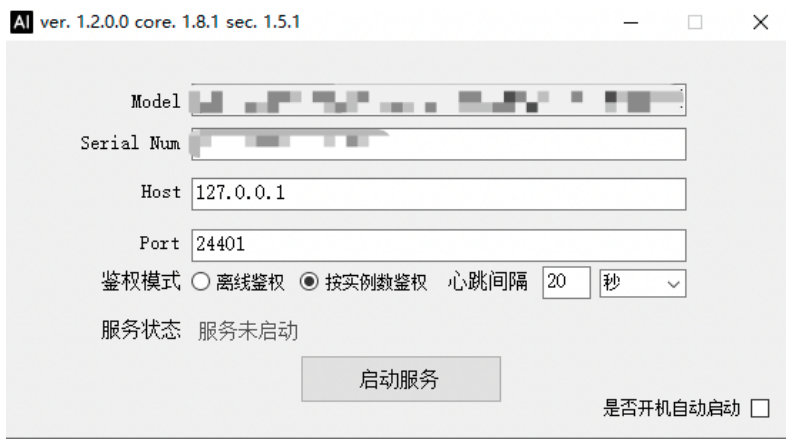
其他任何语言只需通过HTTP调用即可。

### 2.1 离线鉴权（默认鉴权模式） 首次联网激活，后续离线使用



### 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间





基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

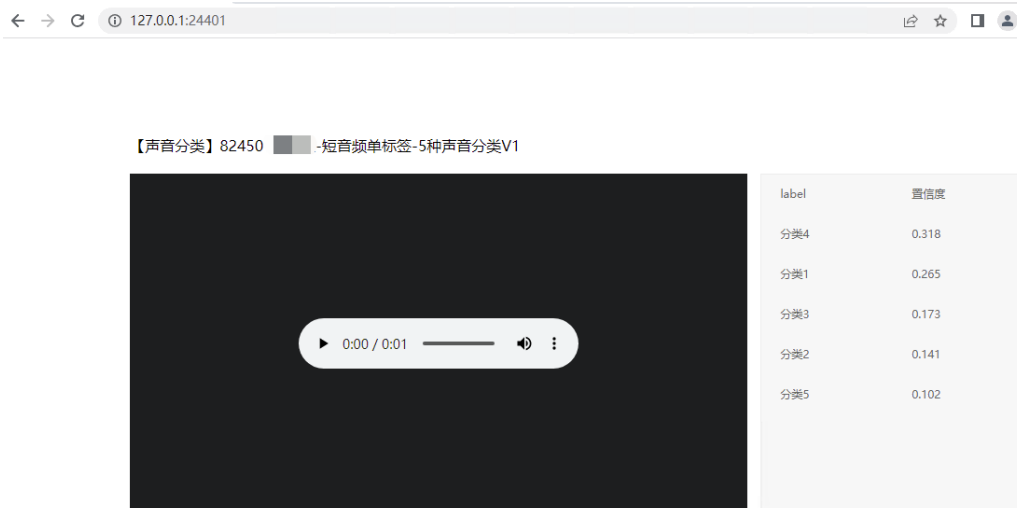
```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

### 3. Demo示例

服务运行成功，此时可直接在浏览器中输入<http://127.0.0.1:24401>，在h5中测试模型效果。



使用说明

调用说明

Python 使用示例代码如下

```
import requests

with open('./1.mp3', 'rb') as f:
    audio = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=audio).json()
```

C# 使用示例代码如下

```

    FileStream fs = new FileStream("./audio.mp3", FileMode.Open);
    BinaryReader br = new BinaryReader(fs);
    byte[] audio = br.ReadBytes((int)fs.Length);
    br.Close();
    fs.Close();
    string url = "http://127.0.0.1:8402?threshold=0.1";
    HttpRequest request = (HttpRequest)HttpRequest.Create(url);
    request.Method = "POST";
    Stream stream = request.GetRequestStream();
    stream.Write(audio, 0, audio.Length);
    stream.Close();

    HttpResponseMessage response = request.GetResponse();
    StreamReader sr = new StreamReader(response.GetResponseStream());
    Console.WriteLine(sr.ReadToEnd());
    sr.Close();
    response.Close();

```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./audio.mp3";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送声音二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | | ----- | ----- | ---- | ----- | | confidence | float | 0~1 | 分类的置信度 | | label | string | | 分类的类别 | | index | number | | 分类的类别 |

## 集成指南

### 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

### 基于c++ dll集成

#### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

#### 集成方法

参考src目录中的CMakeLists.txt进行集成

### 基于c# dll集成

#### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

#### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下：*.NET Framework 4.5* Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

2. 服务调用时返回为空，怎么处理？ 调用输入的音频不为空。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

☞ 通用小型设备部署

☞ Windows集成文档

## 简介

本文档介绍Windows CPU SDK的使用方法。

- 网络类型支持：图像分类，物体检测，图像分割，声音分类
- 硬件支持：
  - Intel CPU 普通版 \* x86\_64
  - CPU 加速版 - Intel Xeon with AVX2 and AVX512 - *Intel Core Processors with AVX2* - Intel Atom Processors with SSE \* - AMD Core Processors with AVX2
  - Intel Movidius Myriad2/Myriad X (仅支持Win10)
- 操作系统支持
  - 普通版：64位 Windows 7 及以上，64位Windows Server2012及以上
  - 加速版：64位 Windows 10，64位Windows Server 2019及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015-2019
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

\*intel 官方合作，拥有更好的适配与性能表现

**Release Notes** | 时间 | 版本 | 说明 | |-----| ----- |-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | 优化模型算法 | | 2022-09-15 | 1.7.0 | 新增支持表格预测 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | CPU基础版推理引擎优化升级；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | CPU加速版推理引擎优化升级 | | 2021-08-19 | 1.3.2 | 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版！ |

## 快速开始

### 1. 安装依赖

必须安装：

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

### Visual C++ Redistributable Packages for Visual Studio 2015-2019

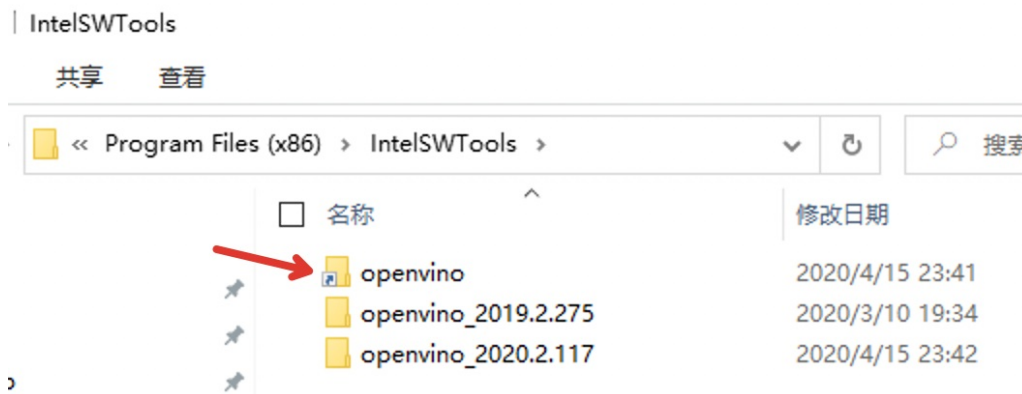
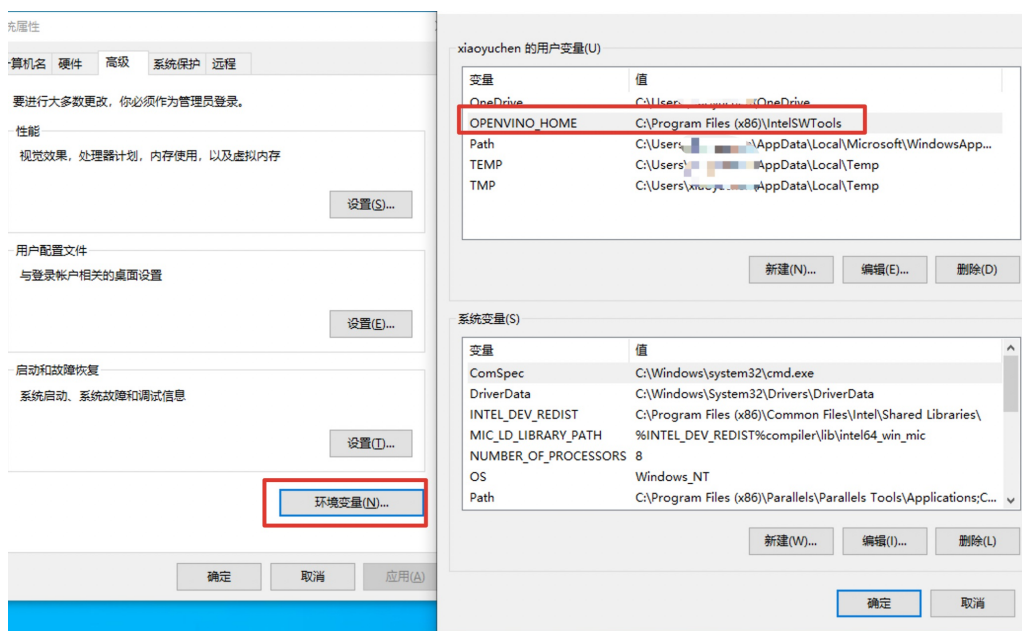
<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-160>

可选安装：

#### Openvino (仅使用Intel Movidius必须)

- 使用 OpenVINO™ toolkit 安装，请参考 [OpenVINO toolkit 文档](#)安装 2020.3.1LTS (必须) 版本, 安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装，请参考 [Openvino Inference Engine文档](#)编译安装 2020.3.1LTS (必须) 版本。

安装完成后，请设置环境变量OPENVINO\_HOME为您设置的安装地址，默认是C:\Program Files (x86)\IntelSWTools，并确保文件夹下的openvino的快捷方式指向了2020.3.1LTS版本。



#### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

#### 2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num，选择鉴权模式，点击"启动服务"，等待数秒即可启动成功，本地服务默认运行在

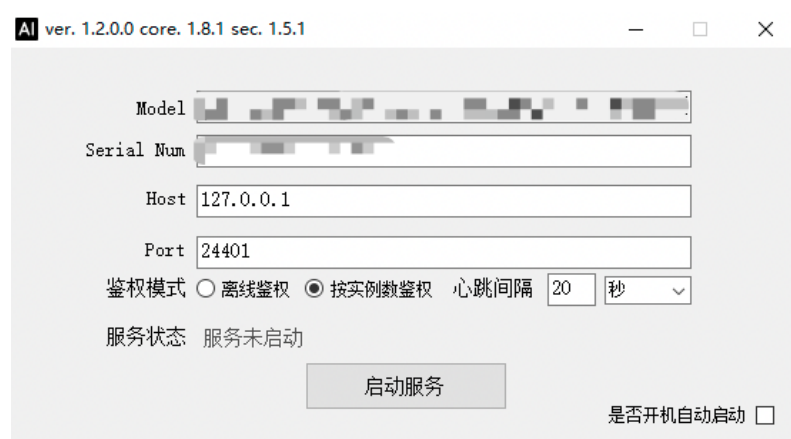
```
http://127.0.0.1:24401/
```

其他任何语言只需通过HTTP调用即可。

## 2.2 离线鉴权（默认鉴权模式） 首次联网激活，后续离线使用



2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间



基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

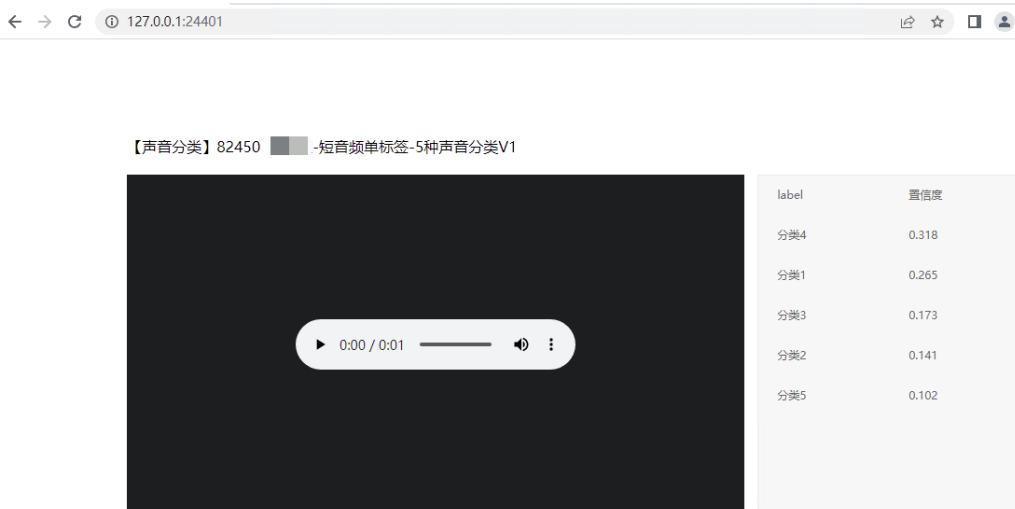
```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

## 3. Demo示例

服务运行成功，此时可直接在浏览器中输入<http://127.0.0.1:24401>，在h5中测试模型效果。



如果上传音频文件后，弹窗报错 500 internal server error，可参看该解决方案：[点击这里跳转](#)

## 使用说明

### 声音服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.mp3', 'rb') as f:
    audio = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post("http://127.0.0.1:24401/", params={"threshold": 0.1},
                       data=audio).json()
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./audio.mp3", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] audio = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(audio, 0, audio.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**

**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./audio.mp3";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        struct curl_slist* headers = NULL;
        // 根据您的音频格式，选择对应的Content-Type
        //MP3: audio/mpeg
        //Ogg: audio/ogg
        //Mav: audio/mav
        headers = curl_slist_append(headers, "Content-Type: audio/mpeg");
        //set headers
        curl_easy_setopt(curl, CURLOPT_HTTPHEADER, headers);
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送声音二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | | ----- | ----- | ---- | ----- | | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 |

### 集成指南



## 基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

## 基于c++ dll集成

### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

### 集成方法

参考src目录中的CMakeLists.txt进行集成

## 基于c# dll集成

### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下：*.NET Framework 4.5* Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是CPU加速版，需额外确保Openvino安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

### 4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

### 7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

### 8. 勾选“开机自动启动”后，程序闪退

一般是写注册表失败。

可以确认下HKEY\_CURRENT\_USER下Software\Microsoft\Windows\CurrentVersion\Run能否写入（如果不能写入，可能被杀毒软件等工具管制）。也可以尝试基于bin目录下的easyedge\_serving.exe命令行形式的二进制，自行配置开机自启动。

### 9.浏览器打开webui上传文件提示 500 internal server error

详细描述：



查看根目录下EasyEdge.log

1、如果报错日志为：“ModuleNotFoundError: No module named 'librosa”

解决方案：

- 1) 打开cmd命令行终端，进入SDK根目录下 python37文件夹
- 2) 使用python37文件夹内自带python解释器安装缺失的依赖项

```
python.exe -m pip install librosa
```

2、如果报错：Microsoft Visual C++ 14.0 is required. Get it with “Microsoft Visual C++ Build Tools”

或者 Failed to build soxr ERROR: Could not build wheels for soxr which use PEP 517 and cannot be installed directly

解决方案：本质原因是缺少数学库的头文件，需下载Microsoft Visual C++ Build Tools：

[https://blog.csdn.net/qq\\_42685893/article/details/129459771](https://blog.csdn.net/qq_42685893/article/details/129459771)

其他问题 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## EasyDL 视频使用说明

### EasyDL视频介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL视频

目前EasyDL视频支持训练以下模型：

- 视频分类

分析短视频的内容，识别出视频内人体做什么动作，物体/环境发生了什么变化

- 目标跟踪

对视频流中的特定运动对象检测识别，获取目标的运动参数，从而实现对后续视频帧该对象的运动预测（轨迹、速度等），实现对运动目标的行为理解

#### 可视化操作

无需深度学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

#### 操作步骤

##### Step 1 创建模型

确定模型名称，记录希望模型实现的功能

##### Step 2 上传并标注数据

前往数据总览页面上传数据，并在线标注数据

##### Step 3 训练模型并校验效果

选择部署方式与算法，用上传的数据一键训练模型

模型训练完成后，可在线校验模型效果

#### Step 4 发布模型

根据训练时选择的部署方式，将模型以云端API的方式发布使用

更详细的操作指导，请参考各类模型下的技术文档

## 视频分类

### 视频分类介绍

#### 简介

EasyDL视频分类是针对视频内容识别推出的一个定制化训练平台。定制视频分类模型，可以用于分析短视频的内容，识别出视频内人体做的是什  
么动作，物体/环境发生了什么变化。

#### 产品功能

- **AI模型训练平台**  
专门用于训练视频内容分析相关业务场景下的高精度AI模型
- **定制化识别**  
客户可自定义要识别的场景，充分满足客户定制化需求
- **全可视化操作**  
所有模型训练相关的操作都可以在网页上进行，无需编程，仅需四步即可部署定制化AI模型

#### 产品优势

- **可即用**  
业务流程极简，全可视化界面操作，无需深度学习基础，仅需四步即可部署定制化AI模型
- **更轻快**  
训练数据每类仅需50个短视频文件，最快15分钟可训练完毕
- **高精度**  
超过三分之二的模型准确率>90%
- **强安全**  
数据加密与隔离，完善的服务调用鉴权，为客户的数据和模型提供企业级安全保障

#### 适用场景

- **人体动作监控**  
定制监控人体特殊动作，比如特殊手势，工地/后厨人员行为等
- **环境变化监控**  
定制监控环境变化，比如山体塌方，泥石流等
- **视频内容分析**  
快速分析视频内容，可用于短视频APP和直播平台中
- **物体状态变化监控**  
定制识别特定物体的移动方向、形态变化等
- **其他**  
尽情脑洞大开，训练你希望实现的视频分类模型

#### 如何访问EasyDL视频分类

- **产品首页**

请访问：[EasyDL视频分类](#)

- 模型训练

登录百度云后，请访问：[视频分类模型训练](#)，可进行模型训练和部署。

- 控制台

登录百度云后，请访问：[百度云控制台](#)，可进行应用创建和模型云服务管理

## 创建模型

### 进入创建模型页面

在[EasyDL视频分类产品主页](#)点击【开始训练】按钮进入到[模型训练页](#)，下面会出现两种情况：

第一种，如果您没有登录百度云，则会跳转到百度云登录页面，没有百度账户的客户请先[注册百度账户](#)。登录后，会跳转到[模型概览页](#)，点击「视频分类」卡片上的「点击前往」按钮，会跳转模型训练页面的创建模型页。

第二种，如果您已登录，会直接进入[到我的模型页](#)，该页面能够管理已经创建的模型，点击左侧列表中的[创建模型](#)进入创建模型页面。

**创建模型** 进入创建模型页面后你会看到如下图中展示的内容

The screenshot shows the 'Create Model' page in the EasyDL interface. The page has a sidebar on the left with navigation options like 'Overview', 'Model Center', 'My Models', 'Create Model', 'Train Model', 'Verify Model', 'Publish Model', 'EasyData Data Service', 'Data Overview', 'Label Management', 'Online Annotation', 'Public Cloud Services', and 'Online Services'. The main content area is titled 'Model List > Create Model'. It contains a form with the following fields:

- Model Category: 视频分类
- Model Name: 手语识别
- Model Affiliation: 公司 (selected) / 个人
- Email Address: z\*\*\*\*\*@baidu.com
- Contact Information: 135\*\*\*\*\*919
- Function Description: (text area with 0/500 character limit)

A blue '完成' (Complete) button is located at the bottom of the form.

需要填写的项目如下：

- 模型名称  
模型的名称
- 模型归属  
模型是属于公司的，还是属于个人的，如果是前者，请填写公司名称
- 选择行业  
请选择您公司所属的行业
- 应用场景  
请选择模型将会被应用于的业务场景
- 邮箱地址  
用于联系到您的邮箱地址
- 联系方式  
有效的联系方式将有助于后续模型上线的人工快速审核，以及更快的百度官方支持，推荐填写个人手机号码
- 功能描述  
描述该模型将要应到的业务场景，详细的描述，在获取官方支持时，能帮助我们为您提供准确的使用建

如下图所示，完成所有填写项后点击【完成】按钮完成模型创建，创建完成后会跳转到[我的模型](#)页面。

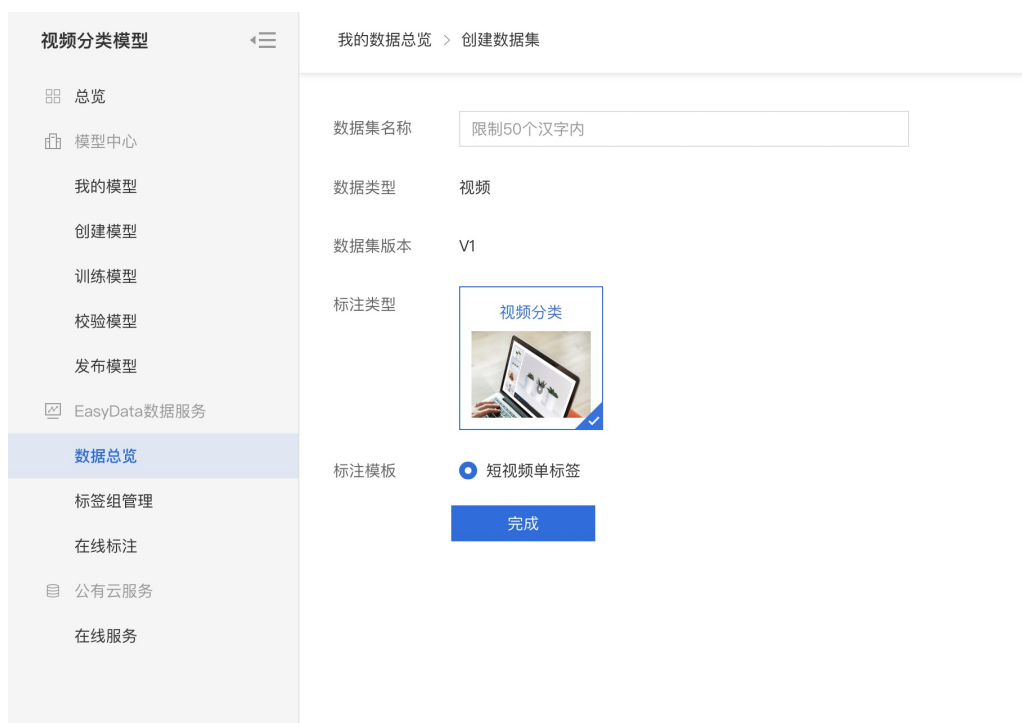
## 数据准备

### 创建数据集

**创建数据集** 完成[模型创建](#)后，会跳转到[我的模型](#)页面，这时您会看到如下图展示的内容，由于模型还未训练，所以模型列表中没有显示模型的效果，在训练模型前，需要先完成创建数据集。

部署方式	版本	训练状态	服务状态	模型效果	操作
公有云API	V3	训练完成	未发布	top1准确率: 70.59% top5准确率: 100.00% <a href="#">完整评估结果</a>	<a href="#">查看版本配置</a> <a href="#">申请发布</a> <a href="#">校验</a>

点击模型列表内的上传或是左侧栏数据中心下的[创建数据集](#)可以进入到创建、导入数据集页面，如下图所示

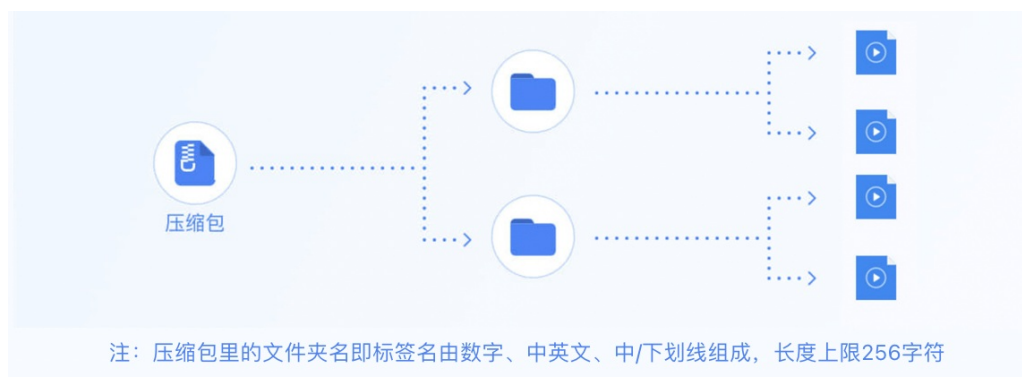


## 上传数据集

### 上传视频分类数据集

已标注数据上传 目前支持本地导入、BOS目录导入、分享链接导入、平台已有数据集导入，4种导入方式。支持的标注格式有文件夹命名分类和json平台通用两种。

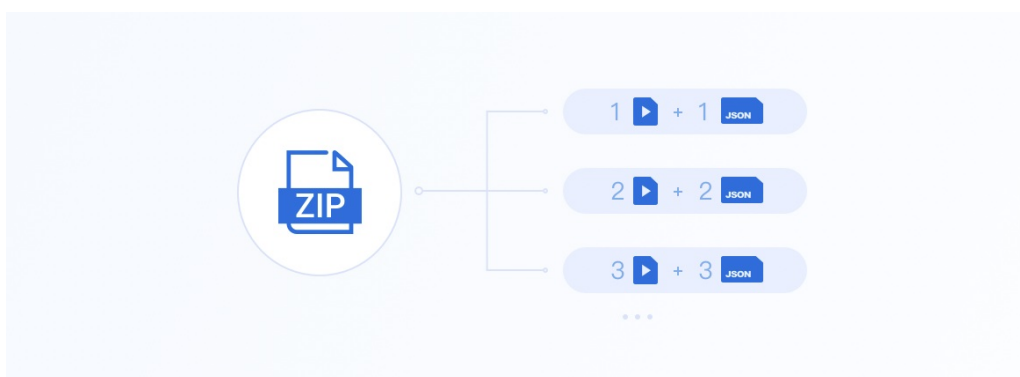
- 以文件夹命名分类



导入要求如下：

1. 上传已标注文件要求格式为zip格式压缩包，同时压缩前源文件大小在5GB以内
2. 压缩包内支持一个文件夹的名称作为标签，文件夹下的所有视频文件作为样本
3. 标签由数字、中英文、中/下划线组成，长度上限256字符
4. 单视频文件类型要求为mp4/mov，单文件大小限制在4M内，视频码率不超过3Mbps，长度限制10s以内
5. 您的账户下数据集数量限制为20G视频，如果需要提升数据额度，可在平台[提交工单](#)

- json平台通用



导入要求如下：

1. 上传已标注文件要求格式为zip格式压缩包，同时压缩前源文件大小在5GB以内
2. 压缩包内需要包括视频源文件（mp4/mov）及同名的json格式标注文件
3. 标注文件中标签由数字、中英文、中/下划线组成，长度上限256字符
4. 单视频文件类型要求为mp4/mov，单文件大小限制在4M内，视频码率不超过3Mbps，长度限制10s以内
5. 您的账户下数据集数量限制为20G视频，如果需要提升数据额度，可在平台[提交工单](#)

**视频内容要求：**

1. 训练视频和实际场景要识别的视频拍摄环境一致，举例：如果实际要识别的视频是摄像头俯拍的，那训练视频就不能用网上下载的目标正面视频
2. 每个视频需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

**未标注数据上传** 目前支持本地导入、BOS目录导入、分享链接导入、平台已有数据集导入，4种导入方式。导入数据的要求为：

1. 压缩包仅支持zip格式，压缩前源文件大小限制5GB以内
2. 单视频文件类型要求为mp4/mov，单次上传限制10个文件
3. 单个视频文件大小限制在4M内，视频码率不超过3Mbps，长度限制120min
4. 分辨率大于1080P的视频会被压缩至1080P，编码格式不是h264格式的视频会被转为h264格式
5. 您的账户下数据集数量限制为20G视频，如果需要提升数据额度，可在平台[提交工单](#)

导入完成后可在平台完成在线数据标注。如有任何问题，请[提交工单](#)联系我们

**视频内容要求：**

1. 训练视频和实际场景要识别的视频拍摄环境一致，举例：如果实际要识别的视频是摄像头俯拍的，那训练视频就不能用网上下载的目标正面视频
2. 每个视频需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

## 🔗 视频分类数据集管理API

目录

1. [数据集管理API介绍](#)
2. [数据集创建API](#)
3. [查看数据集列表API](#)
4. [查看分类（标签）列表API](#)
5. [添加数据API](#)
6. [数据集删除API](#)
7. [分类（标签）删除API](#)

## 8. 错误码

## 数据集管理API介绍

本文档主要说明当您线下已有大量的已经完成分类整理的视频数据时，如何通过调用API完成视频数据的便捷上传和管理。EasyDL数据集管理API在管理不同模型数据类型之间是通用的。上传不同模型类型数据，只是在部分接口入参存在差异，使用及接口地址完全一致。

## 数据集创建API

## 接口描述

该接口可用于创建数据集。

## 接口鉴权

同发布模型后获取的API鉴权方式：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

## 请求说明

## 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/dataset/create

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

## 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
dataset_name	是	string	数据集名称，长度不超过20个utf-8字符

若上传视频分类数据集，在type参数应传「VIDEO\_CLASSIFICATION」

## 返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
dataset_id	否	number	创建的数据集ID

## 查看数据集列表API

## 接口描述

该接口可用于查看数据集列表。返回数据集的名称、类型、状态等信息。



## 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

## 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
start	否	number	起始序号，默认为0
num	否	number	数量，默认20，最多100

若查看视频分类数据集，在type参数应传「VIDEO\_CLASSIFICATION」

## 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	数据集总数
results	否	array(object)	数据集列表
+dataset_id	否	number	数据集ID
+dataset_name	否	string	数据集名称
+type	否	string	数据集类型
+status	否	string	数据集状态
+special_status	否	string	数据集特殊状态，包括shared、smart和空值，分别表示共享中、智能标注中、非特殊状态

## 查看分类（标签）列表API

### 接口描述

该接口可用于查看分类（标签）。返回分类（标签）的名称、包含数据量等信息。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/label/list>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
dataset_id	是	number	数据集ID
start	否	number	起始序号，默认0
num	否	number	数量，默认20，最多100

若查看视频分类的全部分类，在type参数应传「VIDEO\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
total_num	否	number	标签总数
results	否	array(object)	标签列表
+label_id	否	string	标签/分类ID
+label_name	否	string	标签/分类名称
+entity_count	否	number	图片/声音/文本数量

### 添加数据API

#### 接口描述

该接口可用于在指定数据集添加数据。

#### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
dataset_id	是	number	数据集ID
append_label	否	boolean	确定添加标签/分类的行为：追加(true)、替换(false)。默认为追加(true)。
entity_content	是	string	type为IMAGE_CLASSIFICATION/OBJECT_DETECTION/IMAGE_SEGMENTATION/SOUND_CLASSIFICATION/VIDEO_CLASSIFICATION时，填入图片/声音/视频的base64编码；type为TEXT_CLASSIFICATION时，填入utf-8编码的文本。 <b>内容限制为：图像分类base64前10M；物体检测base64前10M；图像分割base64前10M；声音分类base64前4M，声音时长1~15秒；视频分类base64前4M，时长10秒内；文本分类10000个汉字</b>
entity_name	是	string	文件名
labels	是	array(object)	标签/分类数据
+label_name	是	string	标签/分类名称（由数字、字母、中划线、下划线组成），长度限制20B
+left	否	number	物体检测时需给出，标注框左上角到图片左边界的距离(像素)
+top	否	number	物体检测时需给出，标注框左上角到图片上边界的距离(像素)
+width	否	number	物体检测时需给出，标注框的宽度(像素)
+height	否	number	物体检测时需给出，标注框的高度(像素)

若上传视频分类数据集，在type参数应传「VIDEO\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 数据集删除API

### 接口描述

该接口可用于删除数据集。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
dataset_id	是	number	数据集ID

若删除视频分类数据集，在type参数应传「VIDEO\_CLASSIFICATION」

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

### 分类（标签）删除API

### 接口描述

该接口可用于删除分类（标签）。

### 接口鉴权

同模型上线后获取的API：

- 1、在[EasyDL——控制台](#)创建应用
- 2、应用详情页获取API Key和Secret Key

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/label/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

字段	必选	类型	说明
type	是	string	数据集类型，可包括：IMAGE_CLASSIFICATION, OBJECT_DETECTION, IMAGE_SEGMENTATION, SOUND_CLASSIFICATION, VIDEO_CLASSIFICATION, TEXT_CLASSIFICATION 分别对应：图像分类、物体检测、图像分割、声音分类、视频分类、文本分类
dataset_id	是	number	数据集ID
label_name	是	string	标签/分类名称

若删除视频分类的子类，在type参数应传「VIDEO\_CLASSIFICATION」

返回说明

返回参数

字段	必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

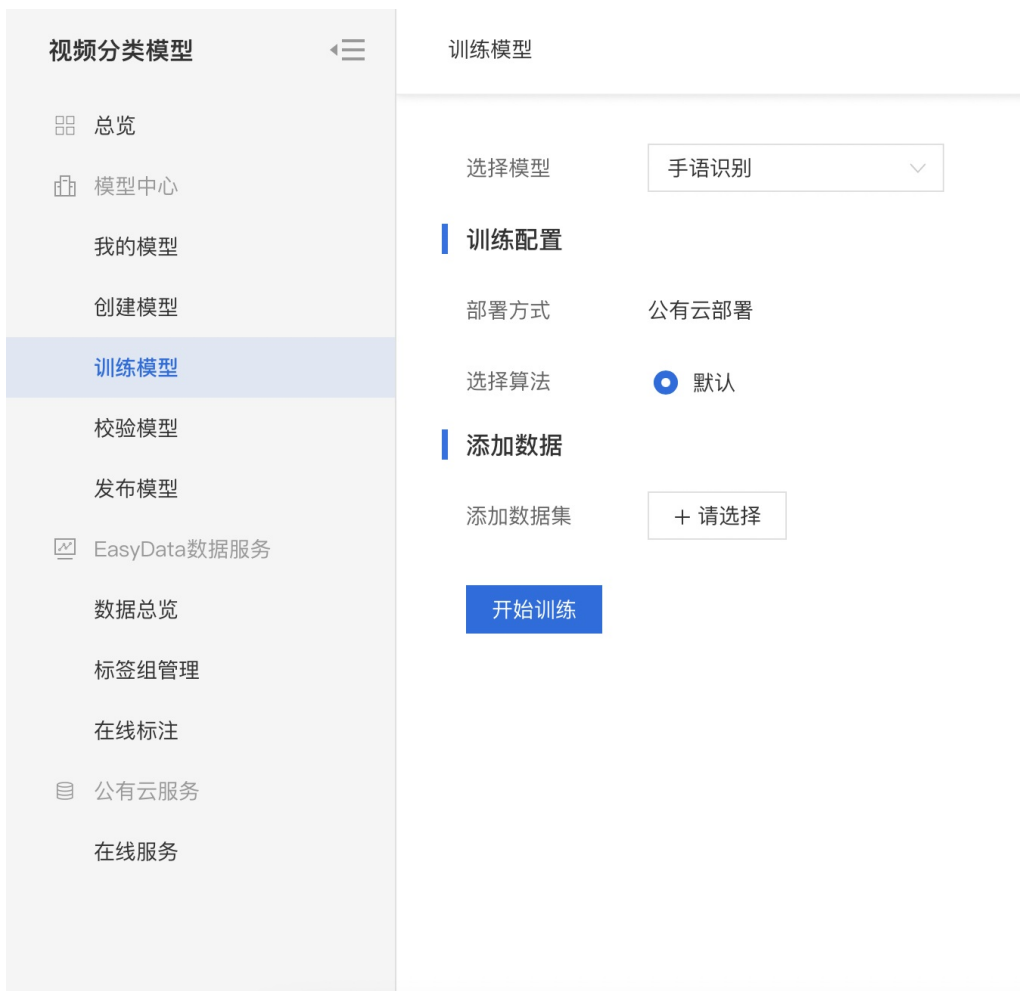
需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	参数xx不合法，请检查相关参数
406002	dataset not exist	数据集不存在
406003	dataset already exists	数据集已存在
406004	dataset can not be modified temporarily	数据集暂不可修改
406005	label not exist	标签/分类不存在
406006	no permission to modify the dataset	没有修改数据集的权限
406007	dataset cannot be modified while smart annotation is running	智能标注期间不可修改数据集
406008	quota exceeded	配额超限

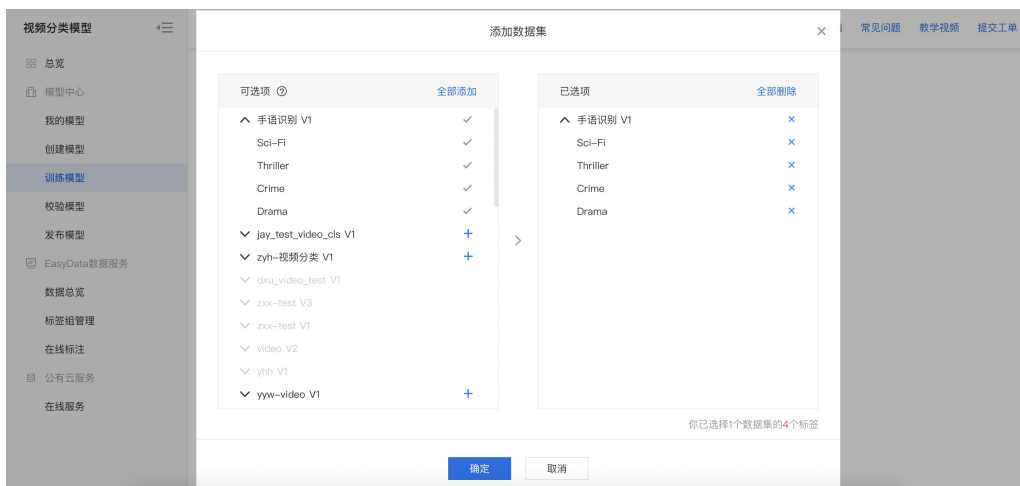
## 模型训练

### 🔗 模型训练操作说明

在完成[上传数据集](#)后，在左侧列表中点击[训练模型](#)进入训练模型页面，如下图所示



选择上传的数据集和相应的标签，如下图所示



完成添加后，如下图所示。



点击**开始训练**后自动跳转到**我的模型**开始训练，训练过程中推荐打开短信通知，如下图所示，这样模型训练好后我们将第一时间以短信的方式告



训练完成后，如下图所示，可以查看模型的完整评估报告，上传一些视频在线**校验模型**。



平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有视频分类操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

## 🔗 模型效果评估报告

简介

模型训练完成后，模型列表中可以看到模型的结果，包括两个指标：top1准确率和top5准确率，也可以点击「完整评估报告」查看更为详细的模型表现，包括准确率、F1-score、精确率和召回率，本文档会介绍如何解读模型的各项指标。

### 模型训练结果

模型的训练结果是如何得到的？

所有训练数据中，系统会随机抽取70%的标注数据作为训练数据，剩余的30%作为测试数据，训练数据训练出的模型去对测试数据进行检测，检测得到的结果跟人为标注的结果进行比对，得到准确率、F1-score、精确率和召回率。

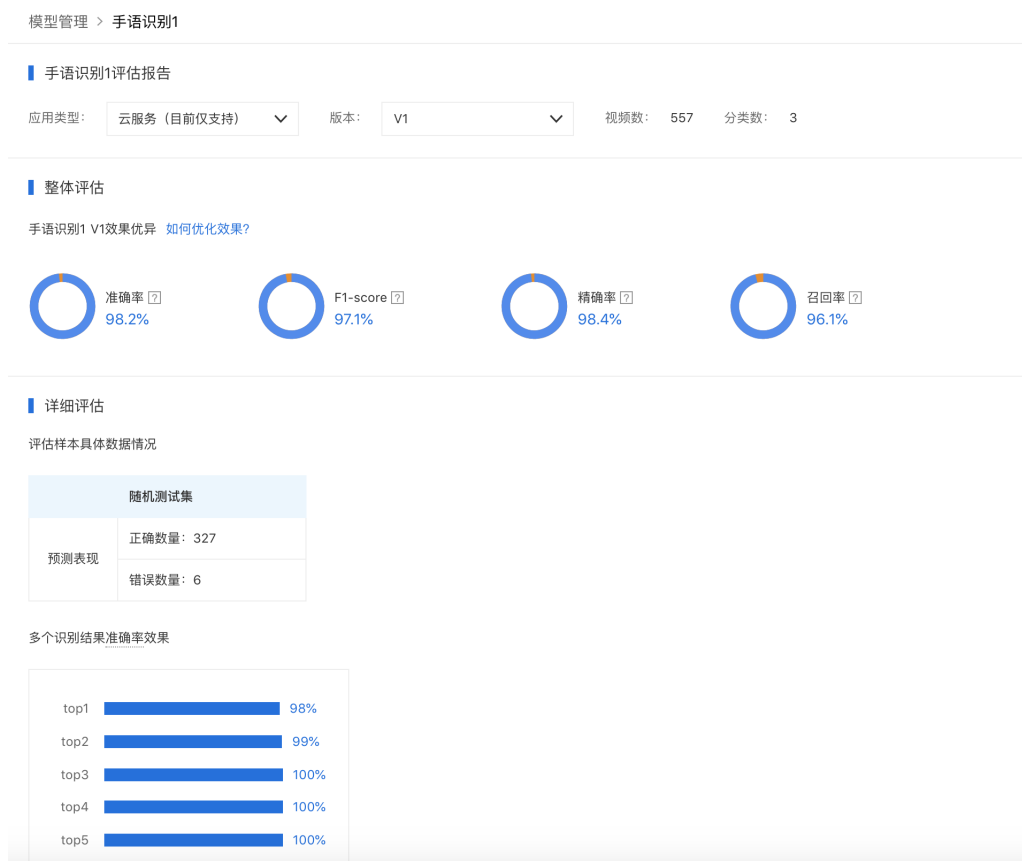
提示：训练数据，即上传的视频越接近真实业务里需要预测的视频，模型训练结果越具有参考性。

在查看模型评估结果可能需要思考在当前业务场景精确率与召回率更关注哪个指标，是更希望减少误识别，还是更希望减少误召回。前者更需要关注召回率的指标，后者更需要关注精确率的指标。同时F1-SCORE可以有效关注精确率和召回率的平衡情况，对于希望召回与识别效果兼具的场景，F1-Score越接近1效果越好。

### 完整评估报告

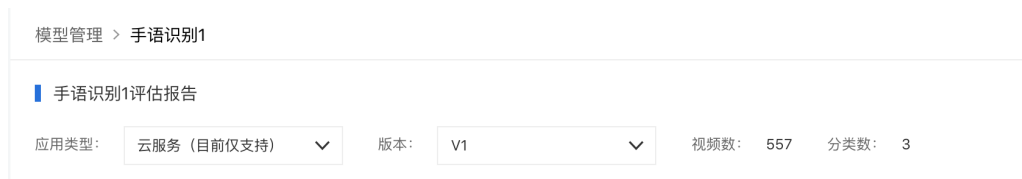
如果需要了解更为详细的模型效果表现，可以在模型列表中点击三项指标下方的「完整评估报告」，完整评估报告页面如下图所示：





## 评估报告

如下图所示：



在这部分可以选择模型的版本，以及看到每个版本参与训练的视频数和分类数。

## 整体评估

如下图所示：



在这部分，四项指标的含义如下：

### • 准确率

准确率 Accuracy = 模型正确预测所有分类的数量/所有分类客观存在的数据总数

对于一个分类模型而言，准确率表示这个模型中所有分类的综合识别效果。如果准确率为1，说明所有分类在测试数据中都被正确识别

### • F1-score

F1-score是模型中一个分类的精确率和召回率的调和平均数，对于希望召回与识别效果兼具的场景，F1-Score越接近1效果越好。

### • 精确率

精确率 Precision = 模型正确预测为该分类的数量/模型预测为该分类的总数

对于一个分类而言，精确率越高，说明模型识别出是这个分类的所有结果中，正确数量的占比越高。如果精确率为1，说明识别出的所有结果都

是对的，但不说明该分类全部都被识别出来了，可能会存在漏识别。

- 召回率

召回率 Recall = 模型正确预测为该分类的数量 / 该分类客观存在的数据总数

对于一个分类而言，召回率越高，说明模型越完整地识别出这个分类。如果召回率为1，说明这个分类全部都别模型识别出来了，但不表示识别出是这个分类的结果都是对的，可能会存在误识别。

如果对模型的效果有疑问，可以点击「如何优化效果」查看模型效果不佳的原因，如下图所示：

咨询如何优化效果✕

\* 问题名称: 部分分类效果较差 ▼

提交咨询前，建议先查看 [效果优化参考文档](#)

其他要求: 请输入其他要求

0/2000

提交

将您想咨询的问题描述填写后提交，我们线下会有专员联系到您帮您解决问题。

### 详细评估

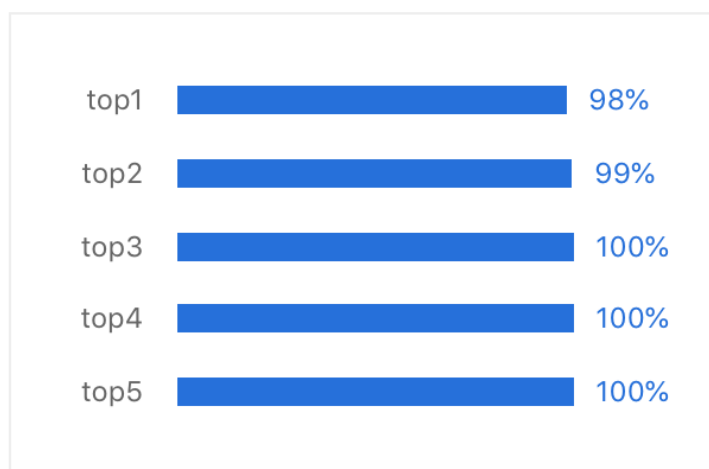
如下图，这里可以看到所有评估报告的数据是基于什么量级的数据进行计算的，当整体参与评估的数量较少时，所有数值可能无法真实反映模型效果。同时，可以看到模型多个识别结果时的准确率。

## 详细评估

评估样本具体数据情况

随机测试集	
预测表现	正确数量：327
	错误数量：6

多个识别结果准确率效果



### 模型发布

#### 🔗 模型发布整体说明

训练完成后，可将模型部署在公有云服务器上，通过API进行调用。

#### 公有云API

- 训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

#### 相关费用

将模型发布为API后，将获得一部分免费调用次数，如需更多调用量，请在百度云控制台内[提交工单](#)反馈。

#### 🔗 发布为API

#### 🔗 如何发布为API

在完成[模型训练](#)后，可将训练好的模型发布为可调用的服务API。

点击模型列表内对应模型「操作」列中的「申请发布」，或是在左侧导航栏点击[发布模型](#)可以进入发布模型页面，如上图所示。在对应选项中选择和输入相应内容发起模型发布的申请：

#### 1. 选择模型（必选）

选择需要发布的模型，**只能选择已经完成训练的模型**

#### 2. 选择服务

视频分类仅支持发布为云服务API

#### 3. 选择版本（必选）

选择需要发布的模型版本，**只能选择完成训练且没有发布过的版本**

#### 4. 服务名称（必填）

为发布的服务命名，**服务名称不得多于20个字符**

#### 5. 接口地址（必填）

自定义服务的API URL，**接口地址需要多于5个字符但不能超过20个字符，仅限英文**

#### 6. 其他要求

如果有其他要求可以输入要求描述

填写完上述信息后，点击「提交申请」完成发布模型申请。提交申请后，模型列表内该模型的申请状态和服务状态为有以下几种情况：

申请状态	服务状态	状态描述
审核中	未发布	服务刚申请发布，模型在审核中
审核成功	发布中	服务通过审核，进入系统自动发布阶段
审核成功	已发布	服务发布成功
审核失败	未发布	服务未通过审核，通常为模型训练结果mAP < 0.6，如需申诉，请在百度云控制台内 <a href="#">提交工单</a> 反馈

提示：第一次申请发布的模型需要人工审核，通常4小时内完成，如果希望加急上线，请在百度云控制台内[提交工单](#)反馈。非第一次申请发布的模型，如果模型训练结果mAP>0.6，训练数据量大于20，则会自动通过审批。审批完成后，大约需要5分钟左右自动完成发布。

发布成功后，可以点击模型列表内「操作」列中的「服务详情」获取服务API URL，点击后弹出下图所示窗口：

## 服务详情



服务名称: silence手语识别01

模型版本: V1

接口地址: https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/video\_cls/silence\_shouyu1

服务状态: 已发布

立即使用

查看API文档

点击「查看API文档」可以快速跳转至API文档，参考文档调用API获取视频分类AI能力。

## 🔗 视频分类API调用文档

## 简介

本文档主要说明视频分类模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)，咨询问题类型请选择人工智能服务
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

## 接口鉴权

## 1、在EasyDL——控制台创建应用

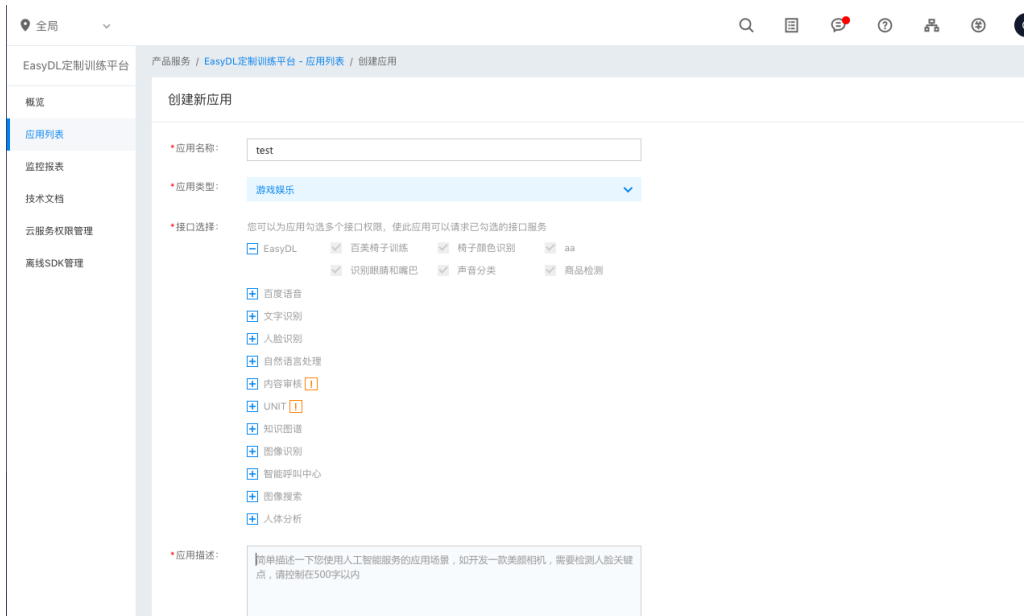
The screenshot shows the 'EasyDL定制训练平台 - 概览' (EasyDL Custom Training Platform - Overview) page. The main content area is divided into two sections: '应用' (Applications) and '可用服务列表' (Available Services List).

**应用 (Applications):** This section shows a summary of applications. A message indicates '已建应用: 0 个' (0 applications built). Below this, there are two buttons: '管理应用' (Manage Applications) and '创建应用' (Create Application), with the latter highlighted by a red box. To the right, a table shows usage statistics for various APIs.

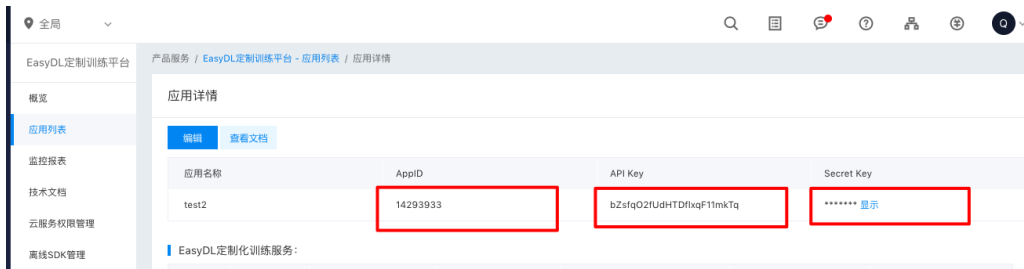
API	调用量	调用失败	失败率	详细统计
百美椅子训练	0	0	0%	<a href="#">查看</a>
椅子颜色识别	0	0	0%	<a href="#">查看</a>
aa	0	0	0%	<a href="#">查看</a>
识别眼睛和嘴巴	0	0	0%	<a href="#">查看</a>
声音分类	0	0	0%	<a href="#">查看</a>

**可用服务列表 (Available Services List):** This table lists the available services for training.

API	模型ID	模型类型	模型名称	模型版本	状态	调用量限制	QPS限制
百美椅子训练	230	图像分类	百美椅子训练	V1	● 免费使用	500次/天免费	不保证并发
椅子颜色识别	2598	图像分类	椅子颜色识别	V1	● 免费使用	500次/天免费	不保证并发
aa	3025	图像分类	test0208	V3	● 暂未配置权限	500次/天免费	不保证并发
识别眼睛和嘴巴	8086	物体检测	识别眼睛和嘴巴	V3	● 免费使用	500次/天免费	不保证并发
声音分类	10789	声音分类	动物声音分类	V3	● 暂未配置权限	500次/天免费	不保证并发



## 2、应用详情页获取AK SK



### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先在[定制视频分类训练平台](#)进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">Access Token获取</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001和336002的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "video": "<base64数据>",
  "top_num": 6
}
```

Body中放置请求参数，参数详情如下：

### 请求参数

参数	是否必选	类型	可选值范围	说明
video	是	string	-	视频，base64编码，建议视频码率不超过3Mbps，长度不超过10s，base64编码后大小不超过4M，支持mp4, mov格式
top_num	否	number	-	返回分类数量，默认为6个

### 请求代码示例

提示一：使用示例代码前，请记得替换其中的示例Token、视频地址或Base64信息。

提示二：部分语言依赖的类或库，请在代码注释中查看下载地址。

```
Python3

"""
EasyDL 视频分类 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

**目标视频的 本地文件路径，支持mp4, mov格式**
VIDEO_FILEPATH = "【您的测试视频地址，例如：./example.mp4】"
```

### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
results	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、视频base64编码错误等等，可检查下视频编码、代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336003	Base64解码失败	图片/音频/文本/视频格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336004	输入文件大小不合法	视频超出大小限制，图片限20M以内，请根据接口文档检查入参格式，有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336005	视频解码失败	视频编码错误（非MP4/MOV视频格式），请检查并修改视频格式
336006	缺失必要参数	video字段缺失（未上传视频）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 常见问题

### 🔗 训练相关问题

#### 数据处理失败或者状态异常怎么办？

请检查已上传的分类命名是否正确，例如是否存在中文命名、或者有空格，如果自查没有发现问题请在百度云控制台内[提交工单](#)反馈。

#### 模型训练失败怎么办？



如果遇到模型训练失败的情况，请在百度云控制台内[提交工单](#)反馈。

### 已经上线的模型还可以继续优化吗？

已经上线的模型依然可以持续优化，操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练，然后在模型列表中更新线上服务，完成模型的优化。

### ☞ 模型效果相关问题

#### 模型效果怎么调优？

当效果不满意时，请查看您的训练数据是否和实际场景中要识别的图片一致，以及训练数据量是否太少。如果训练数据量已经达到一定丰富度，例如单个分类/标签的视频超过50个，效果却仍然不佳，请在百度云控制台内[提交工单](#)反馈。

### ☞ 模型上线相关问题

#### 希望加急上线怎么处理？

请在百度云控制台内[提交工单](#)反馈。

#### 每个账号可以上线几个模型？是否可以删除已上线的模型？

每个账号最多申请发布10个模型，已上线模型无法删除。

### ☞ 收费相关问题

#### 接口上线后是否收费？调用量不够怎么办？

目前是限量免费使用的原则，上线模型后可免费获得500次/天，qps=1的调用限额。**如有超过这个量的需求，请在百度云控制台内[提交工单](#)反馈。**费用问题不用太过担心，如果所需要的量级非常高，可能会基于实际要求适当收费。价格可以根据其他已推出的图像等能力的价格作为参考。

### ☞ 其他问题

#### 模型能否支持私有化部署？

目前我们提供的方案支持公有云API在线调用，尚不支持离线SDK和服务端的私有化部署。

#### 申请发布模型审核不通过都是什么原因？

可能为以下原因：1、当前您的模型存在一些问题，如训练数据异常、数据量不够，或者您不想再继续使用等一系列原因，我们会和您通过电话沟通，沟通达成一致后再拒绝。2、您的电话未接通且模型效果较差，会直接拒绝。如果需要申诉，请在百度云控制台内[提交工单](#)反馈。

## 目标跟踪

### 目标跟踪介绍

### ☞ 功能介绍

目标跟踪是指对视频流中的特定运动对象检测识别，获取目标的运动参数，从而实现对后续视频帧该对象的运动预测（轨迹、速度等），实现对运动目标的行为理解。

### ☞ 应用场景

目标计数：流水线上特定产品的数量统计；商场、旅游景点的人流统计等

智能化交通：人流、车流分析；行人运动轨迹预测；交通违规抓拍等

人/动物的轨迹分析：监控摄像下的行人可疑移动轨迹分析；养殖场动物移动轨迹监测等

应用示例1，飞机轨迹跟踪：



应用示例2，生猪行为分析：



开始使用

### Step 1 创建模型

确定模型名称，记录希望模型实现的功能

### Step 2 上传并标注数据

前往数据总览页面上传数据，并在线标注数据

### Step 3 训练模型并校验效果

选择部署方式与算法，用上传的数据一键训练模型

模型训练完成后，可在线校验模型效果

### Step 4 发布模型

将模型以本地部署的方式发布使用

更详细的操作指导，请参考左侧导航栏中各步骤的操作文档

## 创建模型

### 进入创建模型页面

在[EasyDL视频](#)点击【立即使用】按钮后，选择目标跟踪到[目标跟踪模型训练页](#)，下面会出现两种情况：

第一种，如果您没有登录百度云，则会跳转到百度云登录页面，没有百度账户的客户请先[注册百度账户](#)。登录后，会跳转到[模型概览页](#)，点击「视频分类」卡片上的「点击前往」按钮，会跳转模型训练页面的创建模型页。

第二种，如果您已登录，会直接进入到我的[模型](#)页，该页面能够管理已经创建的模型，点击左侧列表中的[创建模型](#)进入创建模型页面。

**创建模型** 进入创建模型页面后你会看到如下图中展示的内容

目标跟踪 <三> 模型列表 > 创建模型 [操作文档](#)

模型类别 目标跟踪

模型名称 \*

模型归属  公司  个人

请输入公司名称

所属行业 \* 请选择行业

应用场景 \* 请选择应用场景

邮箱地址 \* z\*\*\*\*\*@baidu.com

联系方式 \* 135\*\*\*\*919

功能描述 \*

0/500

完成

需要填写的项目如下：

- 模型名称  
模型的名称
- 模型归属  
模型是属于公司的，还是属于个人的，如果是前者，请填写公司名称

- 所属行业  
请选择您公司所属的行业
  - 应用场景  
请选择模型将会被应用于的业务场景
  - 邮箱地址  
用于联系到您的邮箱地址
  - 联系方式  
有效的联系方式将有助于后续模型上线的人工快速审核，以及更快的百度官方支持，推荐填写个人手机号码
  - 功能描述  
描述改模型将要应到的业务场景，详细的描述，在获取官方支持时，能帮助我们为您提供准确的使用建
- 像下图展示的一样完成所有填写项后点击【下一步】按钮完成模型创建，创建完成后会跳转到[我的模型](#)页面。

目标跟踪

模型列表 > 创建模型

模型类别 目标跟踪

模型名称 \* 车辆识别跟踪

模型归属  公司  个人

百度

所属行业 \* 交通出行

应用场景 \* 视频流中的目标移动轨迹分析

邮箱地址 \* z\*\*\*\*\*@baidu.com

联系方式 \* 135\*\*\*\*\*919

功能描述 \* 识别交通出行场景中的违章车辆并跟踪轨迹

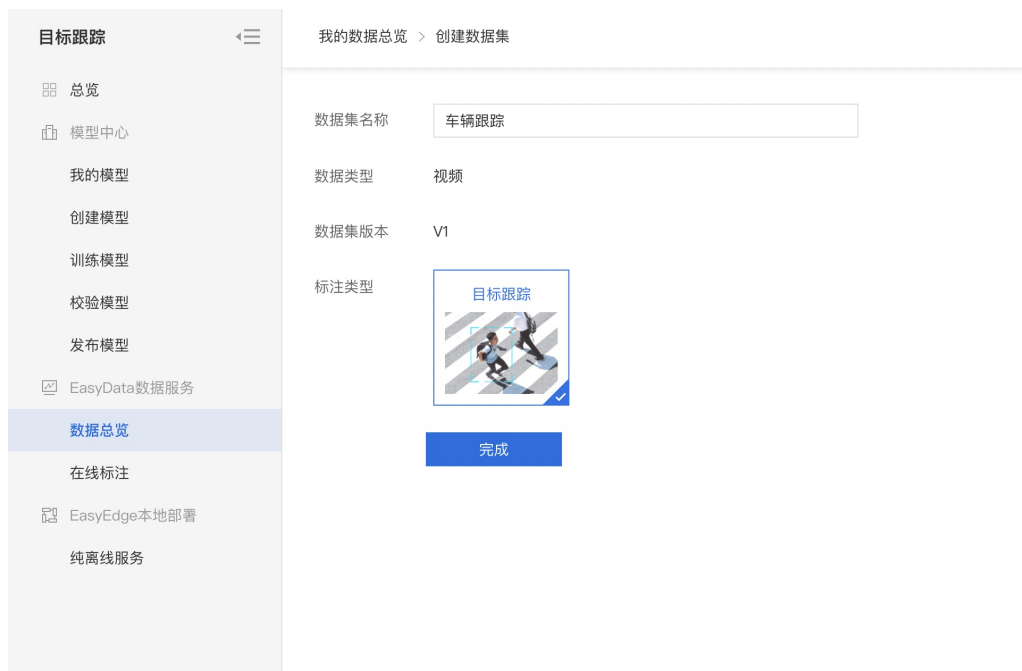
19/500

完成

## 数据准备

### 创建数据集

**创建数据集** 在训练之前需要在导航栏的「数据总览」页面【创建数据集】，如下图所示。



点击【完成】，成功创建数据集



## 上传数据集

**上传目标跟踪数据集 已标注数据上传** 基于CVAT标注好的数据以MOT1.1的数据集形式导出，上传数据压缩包：

- 压缩包仅支持zip格式，大小限制5GB以内
- 压缩包内单个视频长度限制在10分钟内，至少应上传4个视频标注压缩包
- 从CVAT导出的标注数据压缩包可多次上传一起导入数据集组。也可通过本地解压再添加到同一个文件夹后压缩上传

我的数据总览 > 1111111/V1/导入

**创建信息**

数据集ID: 4420      版本号: V1

备注: [🔗](#)

**标注信息**

标注类型	目标追踪	标注模板	目标追踪
数据总量	0	已标注	0
标签个数	0	目标数	0
待确认	0	大小	0M

**数据清洗**

暂未做过数据清洗任务

**导入数据**

数据标注状态:  有标注信息

导入方式: 本地导入

上传压缩包: [上传压缩包](#) 已上传3个文件

- MOT16-02\_split.zip
- MOT16-05\_split-0.zip
- MOT16-11\_split.zip

[确认并返回](#)

常见问题

1、如何进行数据标注?

2、已标注数据的上传压缩包格式要求

如有任何问题，请[提交工单](#)联系我们

#### 视频内容要求：

- 1、训练视频和实际场景要识别的视频拍摄环境一致，举例：如果实际要识别的视频是摄像头俯拍的，那训练视频就不能用网上下载的目标正面视频
- 2、每个视频需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

如果需要寻求第三方数据采集团队协助数据采集，请在百度云控制台内[提交工单](#)反馈

未标注数据上传 目前支持本地导入、BOS目录导入、分享链接导入、平台已有数据集导入，4种导入方式 本地导入的要求为

1. 压缩包仅支持zip格式，压缩前源文件大小限制5GB以内
2. 单视频文件类型要求为mp4/mov，单次上传限制10个文件
3. 单个视频文件大小限制在3G内，视频码率不超过3Mbps，长度限制120min
4. 分辨率大于1080P的视频会被压缩至1080P，编码格式不是h264格式的视频会被转为h264格式
5. 您的账户下数据集数量限制为20G视频，如果需要提升数据额度，可在平台[提交工单](#)

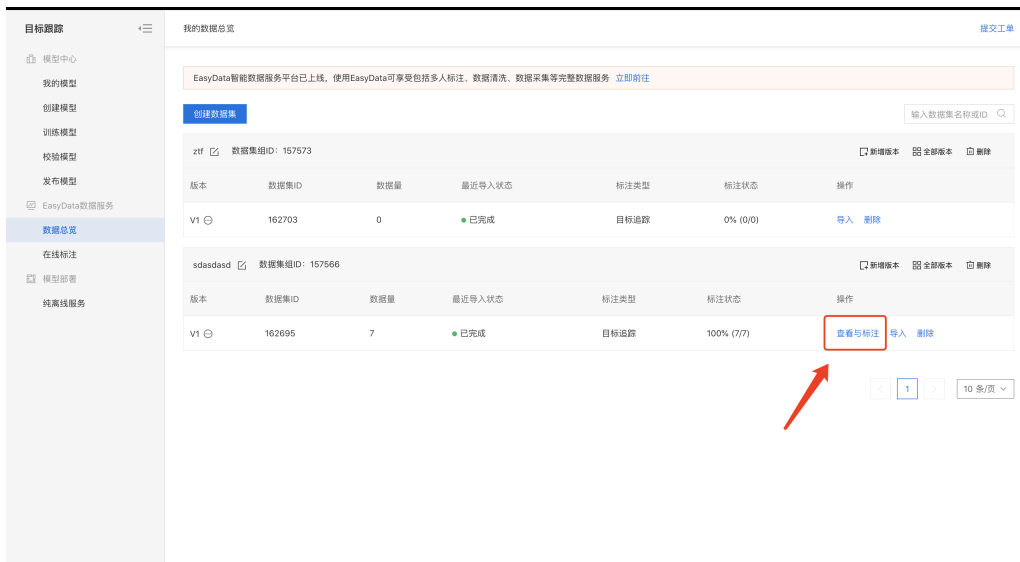
导入完成后可在平台完成在线数据标注。如有任何问题，请[提交工单](#)联系我们

#### 视频内容要求：

- 1、训练视频和实际场景要识别的视频拍摄环境一致，举例：如果实际要识别的视频是摄像头俯拍的，那训练视频就不能用网上下载的目标正面视频
- 2、每个视频需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

#### 在线标注

**目标追踪在线标注** 在创建好数据集，并导入视频数据后。可点击数据总览页面，上传数据对应的「查看与标注」操作开始标注任务。

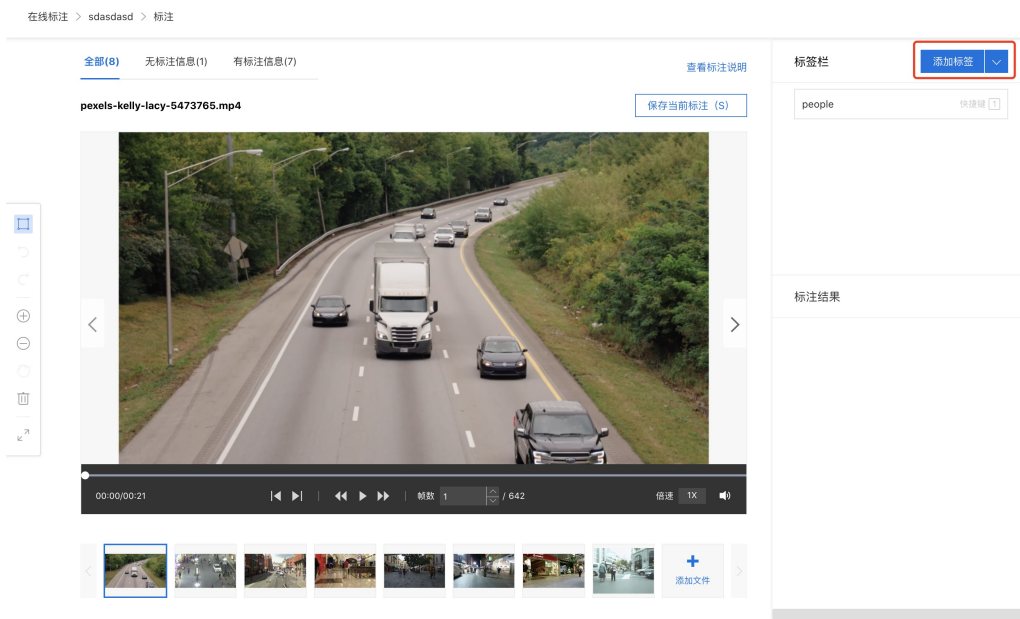


### 在线标注视频示意

### 在线标注图例

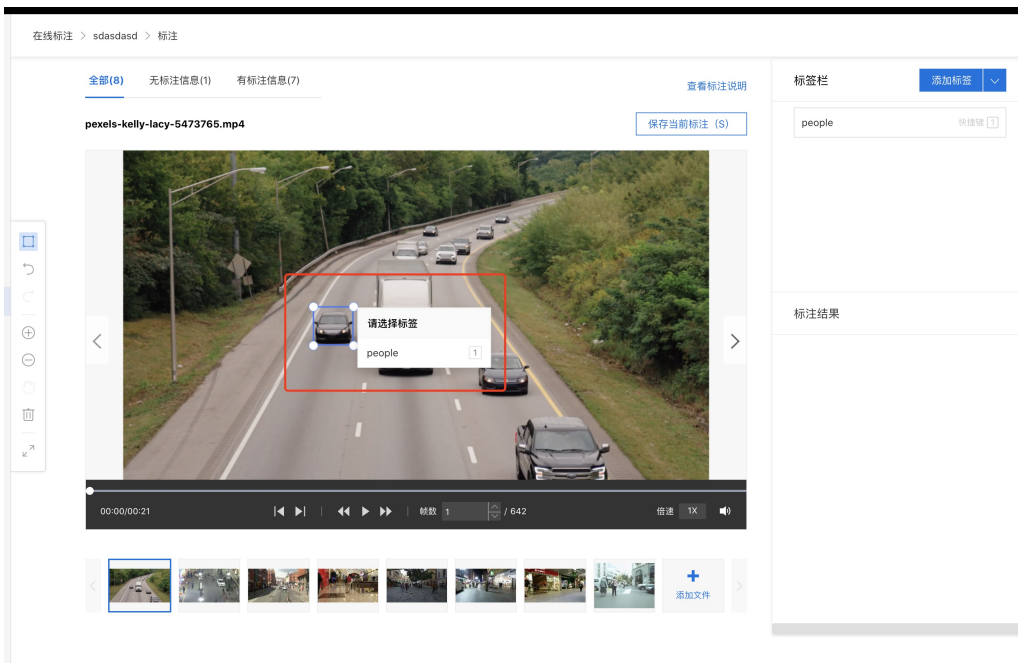
#### 1. 添加标签

点击标注页面右上角的「添加标签」，输入标签名称来添加标签。标签添加成功后可在右侧标签栏查看所添加的标签，并提供「修改标签名称」、「连续标注」的功能



#### 2. 标注目标第一关键帧

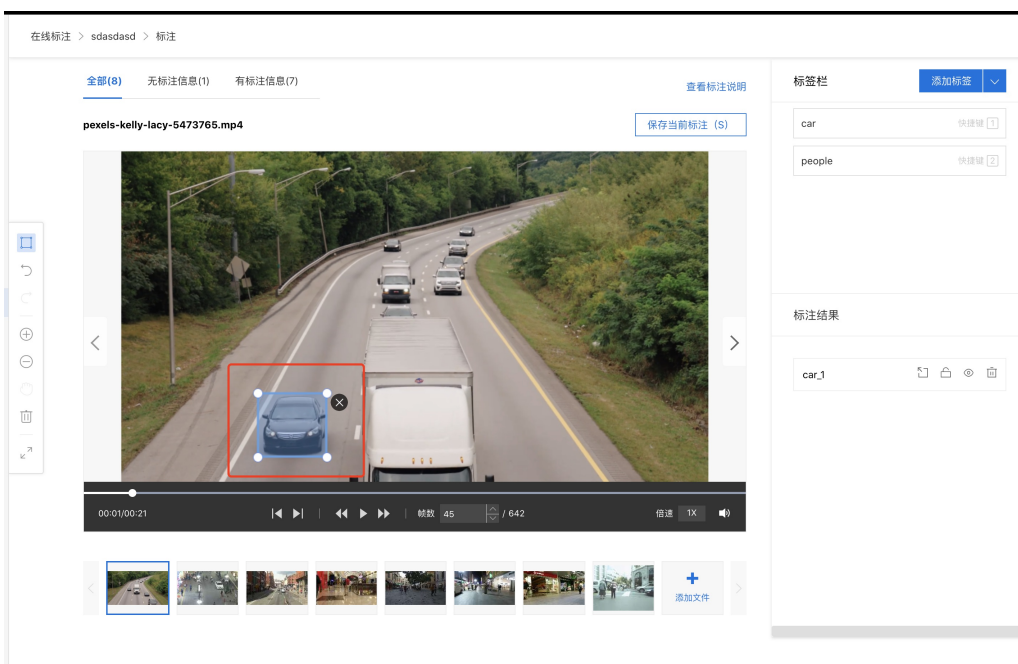
在视频画面中框选出所要识别的目标，并选择标签



### 3.标注目标第二关键帧

在标注完目标第一关键帧之后，播放视频。待目标移动一段距离后，暂停视频并选定标注框，将标注框拖动至目标当前位置，标注目标第二关键帧。此时即完成了目标第一关键帧到第二关键帧之间帧的全部标注，可拖动进度条查看。重复关键帧的标注操作，来完成目标出现在视频画面中的全部标注。

注：如目标移动轨迹不规律、过长等情况下，需要分段标注多个关键帧，来保证目标关键帧之间标注准确性

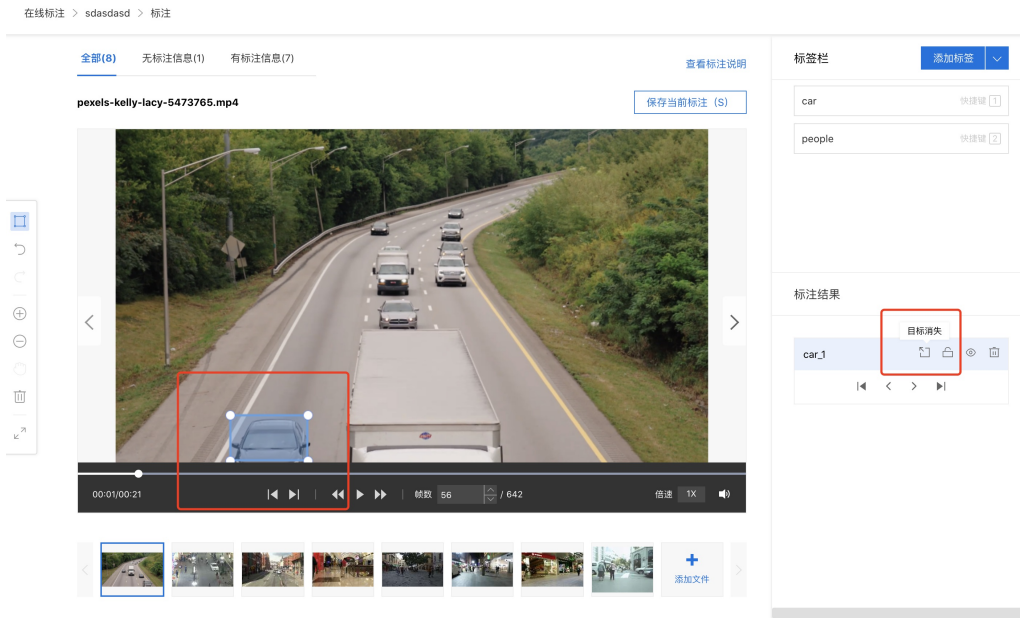


### 4.标注「目标消失帧」

目标即将消失在画面中时，需要标注目标消失帧，从而完成目标的整个标注

注：如目标一直存在在视频画面中，则无需进行目标消失操作

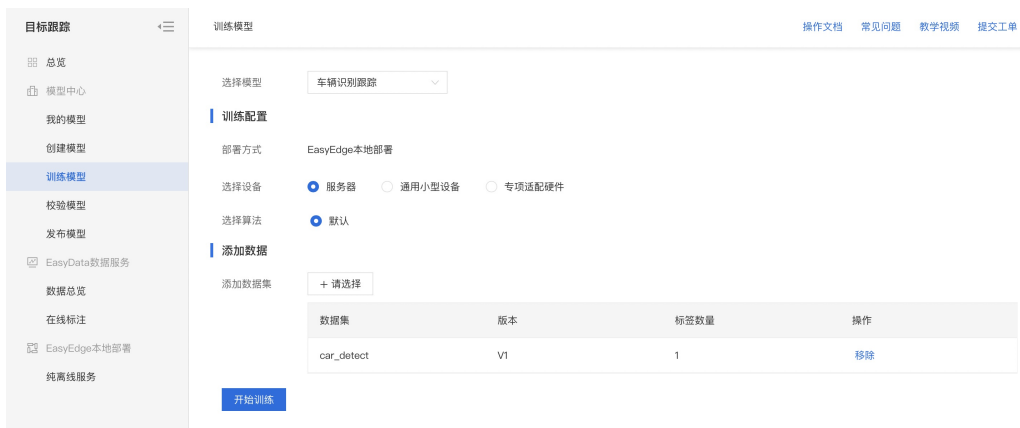




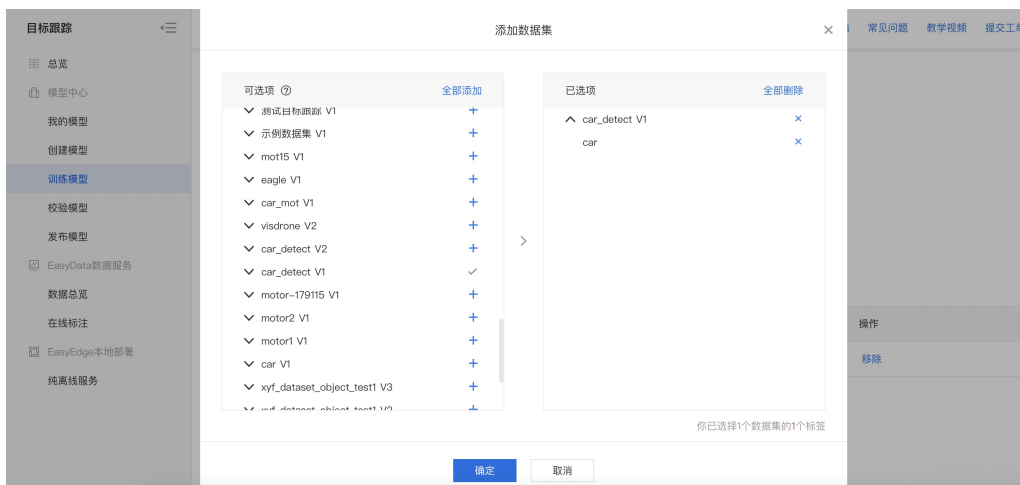
## 模型训练

### 🔗 模型训练操作说明

数据提交后，可以在导航中找到【训练模型】，选择此次训练的模型，并按以下步骤操作，启动模型训练：



添加数据：



### 训练配置

#### 部署方式

目前仅支持「EasyEdge本地部署」

#### 选择设备



目前支持「服务器部署」、「通用小型设备」以及「专项适配硬件」，其中专项适配硬件支持Jetson(Nano/TX2/Xavier)[了解不同方案](#)

## 选择算法

当前仅支持选择默认算法

## 添加数据

### 添加训练数据

添加某单个标签的不同视频数据，在视频数据量为50以上时，模型可以训练充分，得到效果不错的模型。

## 训练模型

点击「开始训练」，训练模型。

- 训练时间与数据量大小有关
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面，如下图所示



平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有目标跟踪操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

## 🔗 模型效果评估报告

### 简介

模型训练完成后，模型列表中可以看到模型的结果，包括三个指标：MOTA、MOTP、召回率，也可以点击「完整评估报告」查看更为详细的模型表现，本文档会介绍如何解读模型的各项指标。

### 模型训练结果

#### 模型的训练结果是如何得到的？

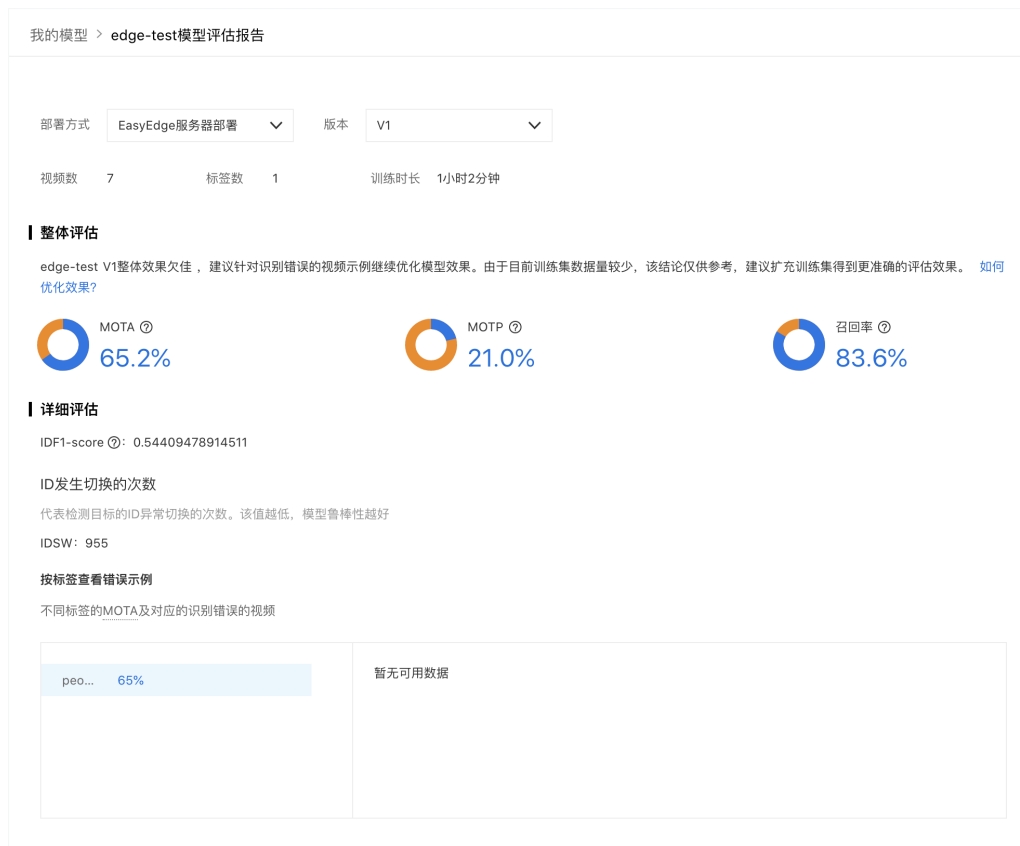
所有训练数据中，系统会随机抽取70%的标注数据作为训练数据，剩余的30%作为测试数据，训练数据训练出的模型去对测试数据进行检测，检测得到的结果跟人为标注的结果进行比对，得到MOTA、MOTP、IDF1-score和召回率。

提示：训练数据，即上传的视频越接近真实业务里需要预测的视频，模型训练结果越具有参考性。

在查看模型评估结果可能需要思考在当前业务场景MOTP与召回率更关注哪个指标，是更希望减少误识别，还是更希望减少误召回。前者更需要关注召回率的指标，后者更需要关注MOTP的指标。同时IDF1-Score可以有效关注MOTP和召回率的平衡情况，对于希望召回与识别效果兼具的场景，IDF1-Score越接近1效果越好。

## 完整评估报告

如果需要了解更为详细的模型效果表现，可以在模型列表中点击三项指标下方的「完整评估报告」，完整评估报告页面如下图所示：



## 评估报告

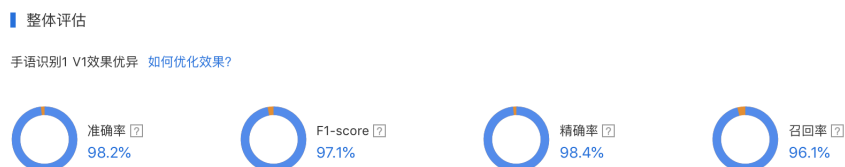
如下图所示：



在这部分可以选择模型的版本，以及看到每个版本参与训练的视频数。

## 整体评估

如下图所示：



在这部分，四项指标的含义如下：

- MOTA

目标跟踪任务中的MOTA指标，指除误报、丢失目标、ID异常切换情况的正确预测样本占有所有样本的比率

对于一个模型而言，MOTA表示这个模型中所有标签的综合识别效果。因效果较差的模型可能存在ID异常切换的情况多，大于样本总数的情况，所以MOTA的取值可能为负， $MOTA \in (-\infty, 1]$ 。如果MOTA为1，说明所有样本在测试数据中都被正确识别

- MOTP

目标跟踪任务中的MOTP指标，指各个阈值都为默认值0.5的情况下正确预测的目标数与预测目标总数之比

对于一个标签而言，MOTP越高，说明模型识别出是这个标签的所有结果中，正确数量的占比越高。如果MOTP为1，说明识别出的所有结果都是对的，但可能会存在漏识别。

• 召回率

召回率 Recall = 模型正确预测为该标签的ID数量/该标签真实存在的ID总数

召回率越高，说明模型越完整地识别出这个标签。

详细评估

• IDF1-score

IDF1-score代表该模型的综合评测效果，越高效果越好。此处为默认平均阈值为0.5时的IDF1-Score

• IDSW

代表检测目标的ID异常切换的次数。该值越低，模型鲁棒性越好

详细评估

IDF1-score ⓘ: 0.59340778488561

ID发生切换的次数

代表检测目标的ID异常切换的次数。该值越低，模型鲁棒性越好

IDSW: 753

按标签查看错误示例


不同标签的MOTA及对应的识别错误的视频

peo... 66%	<p>people的错误结果示例 <a href="#">如何解读错误示例?</a></p> 
------------	--

• 错误示例

可查看按照默认阈值下被判定为错误识别的视频片段样本示例，点击左下角的筛选项可查看正确识别、误识别、漏识别的各个情况

1
✕



■ 正确识别
 ■ 误识别
 ■ 漏识别

正确识别
  误识别
  漏识别

## 模型发布

### 🔗 模型发布整体说明

训练完成后，可将模型部署在本地服务器，通用小型设备以及专项适配硬件

#### 本地服务器部署

- 可将训练完成的模型部署在本地GPU服务器上，支持服务器SDK的集成方式
- 可在内网/无网环境下使用模型，确保数据隐私

#### 通用小型设备

- 通用小型设备部署支持将模型部署在本地的小型计算设备上，提供SDK的集成方式
- 通用小型设备SDK-纯离线服务：支持Windows操作系统，具体的系统、硬件环境支持请参考[技术文档](#)。提供相应代码包、说明文档，供企业用户/开发者二次开发
- 如存在设备无法联网，需要在纯离线的环境下激活的情况，或SDK生成失败等任何其他问题，欢迎[提交工单](#)或加入QQ群（679517246）咨询了解

#### 专项适配硬件

- 为加速开发者们落地离线AI项目，EasyDL提供多种高性价比的软硬一体方案，支持在AI市场直接购买 [了解更多](#)
- 专项硬件适配SDK和激活序列号是EasyDL软硬一体方案的软件部分。如已在其他渠道购买专项适配硬件，也可在此发布为专用SDK，并单独购买激活序列号

### 🔗 服务器端SDK

#### 🔗 目标跟踪服务器端SDK简介

本文档主要说明定制化模型发布后获得的服务器端SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

#### SDK说明

目标跟踪服务器端SDK支持Linux操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux		CPU: x86_64 NVIDIA GPU: x86_64 HUAWAI Atlas 300: x86_64

单次预测时延根据具体设备、线程数不同，数据可能有波动，请以实测为准

#### 激活&使用步骤

离线SDK的激活与使用分以下三步：

- ① 下载SDK后，在[控制台](#)获取序列号
- ② 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

#### ③ 正式使用

#### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在百度云控制台内[提交工单](#)反馈

1、激活失败怎么办？

- ①可能是当前序列号已被其他设备激活使用，请核实序列号后用未被激活的序列号重新激活
- ②序列号填写错误，请核实序列号后重新激活
- ③同一台设备绑定同一个序列号激活次数过多（超过50次），请更换序列号后重试
- ④首次激活需要联网，网络环境不佳或无网络环境，请检查网络环境后重试
- ⑤模型发布者和序列号所属账号非同一账号，如果存在这种异常建议更换账号获取有效序列号
- ⑥序列号已过有效期，请更换序列号后重试
- ⑦如有其他异常请在百度云控制台内[提交工单](#)反馈

🔗 [目标跟踪服务器端SDK集成文档-Linux-C++](#)

## 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持：图像分类，物体检测，图像分割，目标追踪
- 硬件支持：
  - CPU 基础版：- intel x86\_64 \* - AMD x86\_64 - 龙芯 loongarch64 - 飞腾 aarch64
  - CPU 加速版 - Intel Xeon with Intel®AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE - AMD Core Processors with AVX2
  - NVIDIA GPU: x86\_64 PC
  - 寒武纪 Cambricon MLU270
  - 比特大陆计算卡SC5+
  - 百度昆仑XPU K200
    - x86\_64 - 飞腾 aarch64 - 百度昆仑XPU R200
    - x86\_64 - 飞腾 aarch64
  - 华为Atlas 300
  - 海光DCU: x86\_64 PC
  - 寒武纪 MLU370 on x86\_64
- 操作系统支持：Linux

根据开发者的选择，实际下载的版本可能是以下版本之一：

- EasyDL图像
  - x86 CPU 基础版
  - x86 CPU 加速版
  - Nvidia GPU 基础版
  - Nvidia GPU 加速版
  - x86 mlu270基础版
  - x86 SC5+基础版
  - Phytium MLU270基础版

- Phytium XPU基础版
- Phytium Atlas300I基础版
- Hygon DCU基础版

性能数据参考[算法性能及适配硬件](#)

\*intel 官方合作，拥有更好的适配与性能表现。

## Release Notes

时间	版本	说明
2022.1 2.29	1.7.2	模型性能优化；推理库性能优化
2022.1 0.27	1.7.1	新增语义分割模型http请求示例；升级海光DCU SDK，需配套rocm4.3版本使用；Linux GPU基础版下线适用于CUDA10.0及以下版本的SDK；Linux GPU加速版升级推理引擎版本
2022.0 9.15	1.7.0	Linux GPU加速版升级预测引擎；Linux GPU加速版适用于CUDA9.0、CUDA10.0的SDK为deprecated，未来移除；新增实例分割高性能模型离线部署；性能优化
2022.0 7.28	1.6.0	Linux CPU普通版、Linux GPU普通/加速版、Jetson新增目标追踪模型接入实时流的demo
2022.0 5.27	1.5.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2022.0 5.18	1.5.0	GPU加速版max_batch_size参数含义变更；修复GPU加速版并发预测时部分图片结果预测错误及耗时增加问题；CPU普通版预测引擎升级；新增版本号头文件；新增飞腾Atlas300I支持，并且在EasDL新增多种加速版本；示例代码移除frame_buffer，新增更安全高效的safe_queue；新增Tensor In/Out接口和Demo
2022.0 4.25	1.4.1	EasyDL, BML升级支持paddle2模型
2022.0 3.25	1.4.0	新增支持海光服务器搭配海光DCU加速卡；
2021.1 2.22	1.3.5	GPU加速版支持自定义模型文件缓存路径；新增支持飞腾MLU270服务器、飞腾XPU服务器
2021.1 0.20	1.3.4	CPU加速版推理引擎优化升级，新增支持飞腾CPU、龙芯CPU服务器、比特大陆计算卡SC5+ BM1684、寒武纪MLU270；大幅提升EasyDL GPU加速版有损压缩加速模型的推理速度
2021.0 8.19	1.3.2	CPU、GPU普通版及无损加速版新增支持EasyDL小目标检测，CPU普通版、GPU普通版支持检测模型的batch预测
2021.0 6.29	1.3.1	CPU普通版、GPU普通版支持分类模型的batch预测，CPU加速版支持分类、检测模型的batch预测；GPU加速版支持CUDA11.1；视频流解析支持调整分辨率；预测引擎升级
2021.0 5.13	1.3.0	新增视频流接入支持；模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告
2021.0 3.09	1.2.1	GPU新增目标追踪支持，http server服务支持图片通过base64格式调用，EasyDL高性能检测模型和均衡检测模型CPU加速版新增量化压缩模型
2021.0 1.27	1.1.0	EasyDL经典版分类高性能模型升级；部分SDK不再需要单独安装OpenCV
2020.1 2.18	1.0.0	1.0版本发布！安全加固升级、性能优化、引擎升级、接口优化等多项更新
2020.1 1.26	0.5.8	EasyDL经典版分类模型CPU加速版里新增量化压缩模型
2020.1 0.29	0.5.7	新增CPU加速版支持：EasyDL经典版高精度、超高精度物体检测模型和EasyDL经典版图像分割模型
2020.0 9.17	0.5.6	性能优化，支持更多模型

2020.0 8.11	0.5.5	提升预测速度；支持百度昆仑芯片
2020.0 5.15	0.5.3	优化性能，支持专业版更多模型
2020.0 4.16	0.5.2	支持CPU加速版；CPU基础版引擎升级；GPU加速版支持多卡多线程
2020.0 3.12	0.5.0	x86引擎升级；更新本地http服务接口；GPU加速版提速，支持批量图片推理
2020.0 1.16	0.4.7	ARM引擎升级；增加推荐阈值支持
2019.1 2.26	0.4.6	支持海思NNIE
2019.1 1.02	0.4.5	移除curl依赖；支持自动编译OpenCV；支持EasyDL专业版 Yolov3；支持EasyDL经典版高精度物体检测模型升级
2019.1 0.25	0.4.4	ARM引擎升级，性能提升30%；支持EasyDL专业版模型
2019.0 9.23	0.4.3	增加海思NNIE加速芯片支持
2019.0 8.30	0.4.2	ARM引擎升级；支持分类高性能与高精度模型
2019.0 7.25	0.4.1	引擎升级，性能提升
2019.0 7.25	0.4.0	支持Xeye, 细节完善
2019.0 6.11	0.3.3	paddle引擎升级；性能提升
2019.0 5.16	0.3.2	新增NVIDIA GPU支持；新增armv7l支持
2019.0 4.25	0.3.1	优化硬件支持
2019.0 3.29	0.3.0	ARM64 支持；效果提升
2019.0 2.20	0.2.1	paddle引擎支持；效果提升
2018.1 1.30	0.1.0	第一版！

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE含义变更。变更前：预测输入图片数不大于该值均可。变更后：预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理，在输入图片数小于该值时依然可正常运行，但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值，尽可能保持最小。

2020-12-18: 【接口升级】 参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。【关于SDK包与RES模型文件夹配套使用的说明】我们强烈建议用户使用部署tar包中配套的SDK和RES。更新模型时，如果SDK版本号有更新，请务必同时更新SDK，旧版本的SDK可能无法正确适配新发布出来部署包中的RES模型。

## 快速开始

SDK在以下环境中测试通过

- x86\_64, Ubuntu 16.04, gcc 5.4

- x86\_64, Ubuntu 18.04, gcc 7.4
- Tesla P4, Ubuntu 16.04, cuda 9.0, cudnn 7.5
- x86\_64, Ubuntu 16.04, gcc 5.4, XTCL r1.0
- aarch64, Kylin V10, gcc 7.3
- loongarch64, Kylin V10, gcc 8.3
- Bitmain SC5+ BM1684, Ubuntu 18.04, gcc 5.4
- x86\_64 MLU270 , Ubuntu 18.04, gcc 7.5
- phytium MLU270 , Kylin V10 , gcc 7.3.0
- phytium XPU , Kylin V10 , gcc 7.3.0
- hygon DCU, CentOS 7.8 gcc 7.3.0
- XPU K200, x86\_64, Ubuntu 18.04
- XPU K200 aarch64, Ubuntu 18.04
- XPU R200, x86\_64, Ubuntu 18.04
- XPU R200 aarch64, Ubuntu 18.04
- MLU370, x86\_64, Centos7.6.1810

#### 依赖包括

- cmake 3+
- gcc 5.4 (需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.11 (可选)
- cuda9.0\_cudnn7 (使用NVIDIA-GPU时必须)
- XTCL 1.0.0.187 (使用昆仑服务器时必须)
- Rocm4.3, Miopen 2.14(使用海光DCU服务器时必须)

#### 1. 安装依赖

以下步骤均可选，请开发者根据实际运行环境选择安装。

##### (可选) 安装cuda&cudnn

**在NVIDIA GPU上运行必须(包括GPU基础版，GPU加速版)**

对于GPU基础版，若开发者需求不同的依赖版本，请在[PaddlePaddle官网](#) 下载对应版本的libpaddle\_fluid.so或参考其文档进行编译，覆盖lib文件夹下的相关库文件。

##### (可选) 安装TensorRT

**在NVIDIA GPU上运行GPU加速版必须**

下载包中提供了对应 cuda9.0、cuda10.0、cuda10.2、cuda11.0+四个版本的 SDK，cuda9.0 和 cuda10.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.0.0.11，cuda10.2 及以上的 SDK 默认依赖的 TensorRT 版本为 TensorRT8.4，请在[这里](#)下载对应 cuda 版本的 TensorRT，并把其中的lib文件拷贝到系统lib目录，或其他目录并设置环境变量。

**(可选) 安装XTCL 使用昆仑服务器及对应SDK时必须** 请安装与1.0.0.187版本兼容的XTCL。必要时，请将运行库路径添加到环境变量。

##### (可选) 安装Rocm、Miopen

**使用海光DCU服务器对应SDK时必须**

海光DCU SDK依赖Rocm 4.3和Miopen 2.14版本，推荐使用easyedge镜像

(registry.baidubce.com/easyedge/hygon\_dcu\_infer:1.0.2.room4.3)，SDK镜像内运行，镜像拉取方式(wget https://aipe-easyedge-



public.bj.bcebos.com/dcu\_docker\_images/hygon\_dcu\_rocm4.3.tar.gz && docker load -i hygon\_dcu\_rocm4.3.tar.gz)，关于海光DCU使用更多细节可参考[paddle文档](#)

**2. 使用序列号激活** 在控制台获取的序列号请通过参数配置结构体EdgePredictorConfig的成员函数set\_config(easyedge::params::PREDICTOR\_KEY\_SERIAL\_NUM, "this-is-serial-num")设置。

具体请参考SDK自带的Demo.cpp文件的使用方法。

### 3. 测试Demo

模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；在Intel CPU上运行CPU加速版，如果thirdparty里包含openvino文件夹的，必须在编译或运行demo程序前执行以下命令：source \${cpp\_kit位置路径}/thirdparty/openvino/bin/setupvars.sh 或者执行 source \${cpp\_kit位置路径}/thirdparty/openvino/setupvars.sh(openvino-2022.1+)

请在官网获取序列号，填写在demo.cpp中



图片加载失败

部分SDK中已经包含预先编译的二进制，如 bin/easyedge\_demo, bin/easyedge\_serving，配置LD\_LIBRARY\_PATH后，可直接运行：  
LD\_LIBRARY\_PATH=./lib ./bin/easyedge\_serving

编译运行：

```
cd src
mkdir build && cd build
cmake .. && make
./easyedge_image_inference (模型RES文件夹) (测试图片路径)
**如果是NNIE引擎，使用sudo运行**
sudo ./easyedge_image_inference (模型RES文件夹) (测试图片路径)
```

如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEGE_BUILD_OPENCV=ON
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

demo运行效果：



图片加载失败

```
> ./easyedge_image_inference ../../../../RES 2.jpeg
2019-02-13 16:46:12.659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit 0.2.1(20190213)
2019-02-13 16:46:14.083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14.326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

对于支持批量预测的模型和SDK，可在使用前修改demo\_image\_inference或demo\_batch\_inference里的batch\_size再编译、执行。

详情请参考下方[使用说明-其他配置部分](#)

#### 4. 测试Demo HTTP 服务

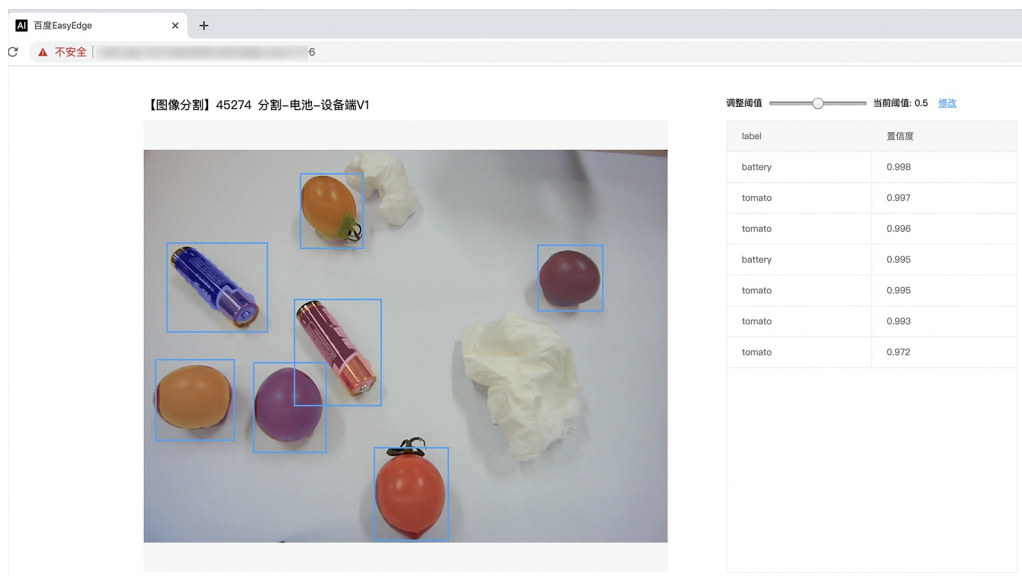
编译demo完成之后，会同时生成一个http服务 运行

```
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

后，日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



对于目标追踪的模型，请选择一段视频，并耐心等待结果



图片加载失败

同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

#### 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

#### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型资源目录
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ; 在这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

输入图片不限制大小

**SDK参数配置** SDK的参数通过 `EdgePredictorConfig::set_config`和`global_controller()->set_config`配置。 `set_config`的所有key在`easyedge_xxxx_config.h`中。其中

- `PREDICTOR`前缀的key是不同模型相关的配置，通过`EdgePredictorConfig::set_config`设置
- `CONTROLLER`前缀的key是整个SDK的全局配置，通过`global_controller()->set_config`设置

以序列号为例，KEY的说明如下：

```

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

使用方法如下：

```

EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");

```

具体支持的运行参数配置列表可以参考开发工具包中的头文件的详细说明。

相关配置均可以通过环境变量的方法来设置，对应的key名称加上前缀`EDGE_`即为环境变量的key。如序列号配置的环境变量key为`EDGE_PREDICTOR_KEY_SERIAL_NUM`，如指定CPU线程数的环境变量key为`EDGE_PREDICTOR_KEY_CPU_THREADS_NUM`。注意：通过代码设置的配置会覆盖通过环境变量设置的值。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image, std::vector<std::vector<EdgeResultData>>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测、图像分割时才有意义
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割的模型, 该字段才有意义
    // 请注意: 图像分割时, 以下两个字体会比较大, 使用完成之后请及时释放EdgeResultData
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask

    // 目标追踪模型, 该字段才有意义
    int trackid; // 轨迹id
    int frame; // 处于视频中的第几帧
    EdgeTrackStat track_stat; // 跟踪状态
};
```

## 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

$y_2$  \* 图片高度 = 检测框的右下角的纵坐标

### 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域，0代表非目标区域
```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding，此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

classVideoDecoding :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

```
struct VideoConfig
```

```
/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type; // 输入源类型
    std::string source_value; // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0}; // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false}; // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0}; // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false}; // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path; // frame存储为视频文件的路径
    bool save_all{false}; // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};
```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。input\_fps：用于抽帧前设置fps。resolution：设置摄像头采样的分辨率，其值请参考easyedge\_video.h中的定义，注意该分辨率调整仅对输入源为摄像头时有效。conf：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的demo\_video\_inference。

#### 设置序列号

请在网页控制台中申请序列号，并在init初始化前设置。LinuxSDK 首次使用需联网授权。

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_SERIAL_NUM, "this-is-serial-num");
```

#### 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

#### http服务

1. 开启http服务 http服务的启动可以参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

## 2. 请求http服务

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片或视频来进行测试。

### http 请求方式一：无额外编码

- 图片测试：不使用图片base64格式

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例 (针对非语义分割模型)

```

import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img_data).json()

```

Python请求示例 (针对语义分割模型)

```

import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    res = requests.post("http://127.0.0.1:24401/",
        data=img_data)
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果

```

### Java请求示例

- 视频测试

Python请求示例 (注意：区别于图片预测，需指定Content-Type；否则会调用图片推理接口)

```
import requests

with open('./1.mp4', 'rb') as f:
    video_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        headers={'Content-Type': 'video'},
        data=video_data).json()
```

http 请求方法二：图片使用base64格式 HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 分类网络：body 中请求示例

```
{
  "image": "<base64数据>"
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 检测和分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms，不含网络交互时间

请求示例 (针对非语义分割模型)



```

import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        result = requests.post("http://{服务ip地址}:24401/", json={
            "image": base64.b64encode(f.read()).decode("utf8")
        })
    # print(result.request.body)
    # print(result.request.headers)
    print(result.content)

if __name__ == '__main__':
    main()

```

请求示例 (针对语义分割模型)

```

import base64
import requests

def main():
    with open("1.jpg 【图片路径】", 'rb') as f:
        res = requests.post("http://{服务ip地址}:24401/", json={"image": base64.b64encode(f.read()).decode("utf8")})
    with open("gray_result.png", "wb") as fb:
        fb.write(res.content) # 语义分割模型是像素点级别输出，可将api返回结果保存为灰度图，每个像素值代表该像素分类结果

if __name__ == '__main__':
    main()

```

返回示例

```

{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}

```

其他配置

### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



## 2. CPU线程数设置

CPU线程数可通过 `EdgePredictorConfig::set_config`配置

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_CPU_THREADS_NUM, 4);
```

## 3. 批量预测设置

```
int batch_size = 2; // 使用前修改batch_size再编译、执行
while (get_next_batch(imgs, img_files, batch_size, start_index)) {
    ...
}
```

**GPU 加速版 预测接口** GPU 加速版 SDK 除了支持上面介绍的通用接口外，还支持图片的批量预测，预测接口如下：

```
/**
 * @brief
 * GPU加速版批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& result
) = 0;

/**
 * @brief
 * GPU加速版批量图片推理接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE`，其含义见下方参数配置接口的介绍。

**运行参数选项** 在上面的内容中我们介绍了如何使用 `EdgePredictorConfig` 进行运行参数的配置。针对GPU加速版开发工具包，目前 `EdgePredictorConfig` 的运行参数所支持的Key包括如下项：

```
/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成最大 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";
```

```

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产生的max_batch_size不相等时，则重新编译模型（推荐）
 * 2: 无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名，这在多进程加载同一个模型的时候是有用的。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产生的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时显

存。

`PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE`：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数需等于此值。

`PREDICTOR_KEY_DEVICE_ID`：设置需要使用的 GPU 卡号。

`PREDICTOR_KEY_GTURBO_COMPILE_LEVEL`：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制。当此值为 0 时，表示忽略当前设置的 `max_batch_size` 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 `max_batch_size` 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

`PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY`：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference 和 multi predictor。

`PREDICTOR_KEY_GTURBO_FP16`：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式。目前已知不支持fp16的模型包括：图像分类高精度模型。

**多线程预测** GPU 加速版 SDK 的多线程分为单卡多线程和多卡多线程两种。单卡多线程：创建一个 predictor，并通过

`PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY` 控制单卡所支持的最大并发量，只需要 init 一次，多线程调用 infer 接口。多卡多线程：多卡的支持是通过创建多个 predictor，每个 predictor 对应一张 GPU 卡，predictor 的创建和 init 的调用放在主线程，通过多线程的方式调用 infer 接口。

**已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时，部分结果错误** A：EasyDL图像分类高精度模型在有些显卡上可能存在此问题，可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

**2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object** A：部分显卡存在此问题，如果遇到此问题，请确认没有频繁调用 init 接口，通常调用 infer 接口即可满足需求。

**3. 开启 fp16 后，预测结果错误** A：不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括：图像分类高精度模型。目前不支持的将在后面的版本陆续支持。

**昆仑服务器** 昆仑服务器SDK支持将EasyDL的模型部署到昆仑服务器上。SDK提供的接口风格一致，简单易用，轻松实现快速部署。Demo的测试可参考上文中的测试Demo部分。

**参数配置接口** 在上面的内容我们介绍了如何使用EdgePredictorConfig进行运行参数的配置。针对昆仑服务器开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```
/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * 使用哪张加速卡
 * 值类型：int
 * 默认值：0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 设置需要同时预测的图片数量
 * 值类型：int
 * 默认值：1
 */
static constexpr auto PREDICTOR_KEY_KUNLUN_BATCH_SIZE = "PREDICTOR_KEY_KUNLUN_BATCH_SIZE";
```

`PREDICTOR_KEY_DEVICE_ID`：设置需要使用的加速卡的卡号。

`PREDICTOR_KEY_KUNLUN_BATCH_SIZE`：设置单次预测可以支持的图片数量。

使用方法：

```
int batch_size = 1;
config.set_config(easyedge::params::PREDICTOR_KEY_KUNLUN_BATCH_SIZE, batch_size);
```

**模型调优** 通过设置如下环境变量，可以在初始化阶段对模型调优，从而让预测的速度更快。

```
export XPU_CONV_AUTOTUNE=5
```

## FAQ

### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3`

方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

### 2. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

### 3. NVIDIA GPU预测时，报错显存不足 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请根据显存大小和模型配置。调整合适的初始 fraction\_of\_gpu\_memory。参数的含义参考[这里](#)。

### 4. 如何将我的模型运行为一个http服务？目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

### 5. 运行NNIE引擎报permission denied 日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

### 6. 运行SDK报错 Authorization failed

**情况一：**日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受 HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：**日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更

- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

### 7. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

### 8. 运行二进制时，提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

9. 运行二进制时提示 libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 同上面8的问题类似，没有正确设置动态库的查找路径，可通过设置LD\_LIBRARY\_PATH为sdk的thirdparty/opencv/lib文件夹解决

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:~/thirdparty/opencv/lib
(tips: 上面冒号后面接的thirdparty/opencv/lib路径以实际项目中路径为准，比如也可能是../thirdparty/opencv/lib)
```

10. 编译时报错：file format not recognized 可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中，再解压缩、编译

11. 进行视频解码时，报错符号未找到、格式不支持、解析出的图片为空、无法设置抽帧 请确保安装OpenCV时，添加了-DWITH\_FFMPEG=ON选项（或者GStream选项），并且检查OpenCV的安装日志中，关于Video I/O段落的说明是否为YES。

```
-- Video I/O:
-- DC1394:          YES (ver 2.2.4)
-- FFMPEG:          YES
-- avcodec:         YES (ver 56.60.100)
-- avformat:        YES (ver 56.40.101)
-- avutil:          YES (ver 54.31.100)
-- swscale:         YES (ver 3.1.101)
-- avresample:      NO
-- libv4l/libv4l2:  NO
-- v4l/v4l2:        linux/videodev2.h
```

如果为NO，请搜索相关解决方案，一般为依赖没有安装，以apt为例：

```
apt-get install yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
```

12. GPU加速版运行有损压缩加速的模型，运算精度较标准模型偏低 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

## 通用小型设备

### 目标跟踪WindowsSDK集成文档

#### 简介

本文档介绍目标跟踪通用小型设备Windows SDK的使用方法。

#### 硬件支持：

- Intel CPU 普通版 \* x86\_64
- CPU 加速版 - Intel Xeon with AVX2 and AVX512 - Intel Core Processors with AVX2 - Intel Atom Processors with SSE
- Intel Movidius Myriad2/Myriad X (仅支持Win10)

- 操作系统支持
  - 普通版：64位 Windows 7 及以上
  - 加速版：64位 Windows 10
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- 协议
  - HTTP

**Release Notes** | 时间 | 版本 | 说明 | |-----|-----|-----| | 2023-08-30 | 1.8.3 | 新增支持按实例数鉴权 | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级，下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 优化模型算法；GPU CUDA9.0 CUDA10.0 标记为待废弃状态 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | GPU基础版推理引擎优化升级；GPU加速版支持自定义模型文件缓存路径；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | 修复已知问题 | | 2021-08-19 | 1.3.2 | 新增支持EasyDL小目标检测，新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版 | |

## 快速开始

### 1. 安装依赖

必须安装：

#### 安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

#### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

#### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

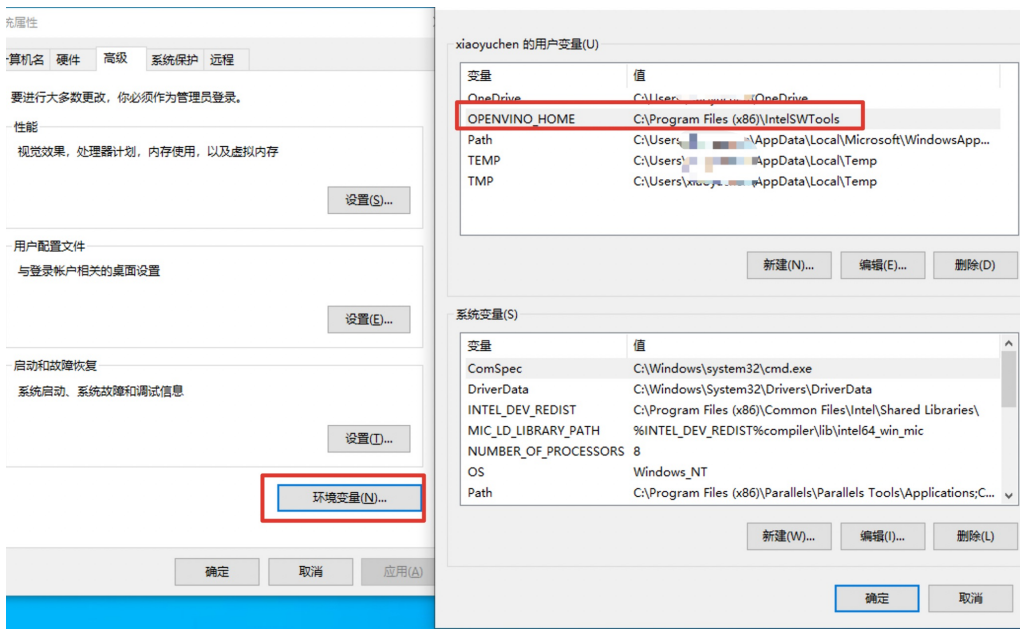
可选安装：

#### Openvino (仅使用Python Intel Movidius必须)

- 使用 OpenVINO™ toolkit 安装，请参考 [OpenVINO toolkit 文档](#) 安装 2020.3.1LTS (必须) 版本，安装时可忽略Configure the Model Optimizer及后续部分。
- 使用源码编译安装，请参考 [Openvino Inference Engine文档](#) 编译安装 2020.3.1LTS (必须) 版本。

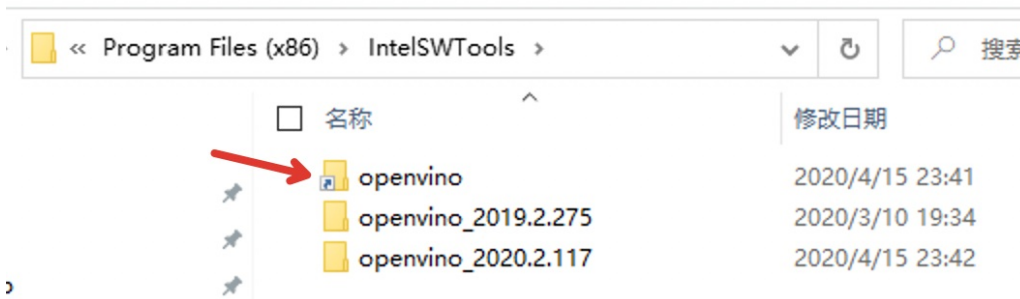
安装完成后，请设置环境变量OPENVINO\_HOME为您设置的安装地址，默认是C:\Program Files (x86)\IntelSWTools，并确保文件夹下的openvino的快捷方式指向了2020.3.1LTS版本。





### IntelSWTools

共享 查看



### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验”，点击安装，安装之后重启即可。

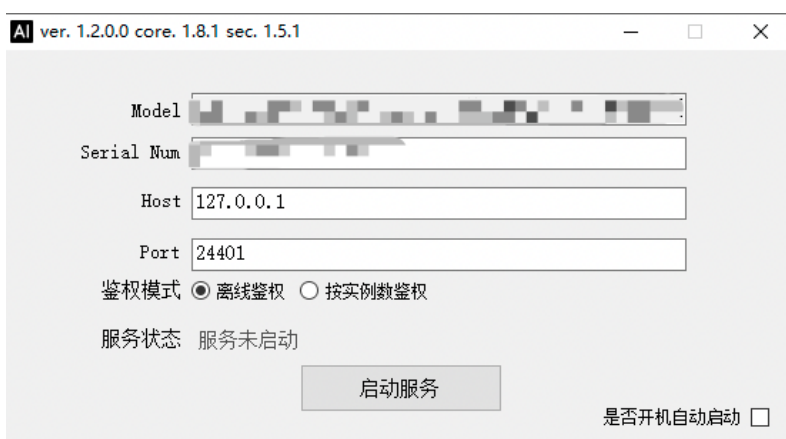
### 2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num，选择鉴权模式，点击“启动服务”，等待数秒即可启动成功，本地服务默认运行在

http://127.0.0.1:24401/

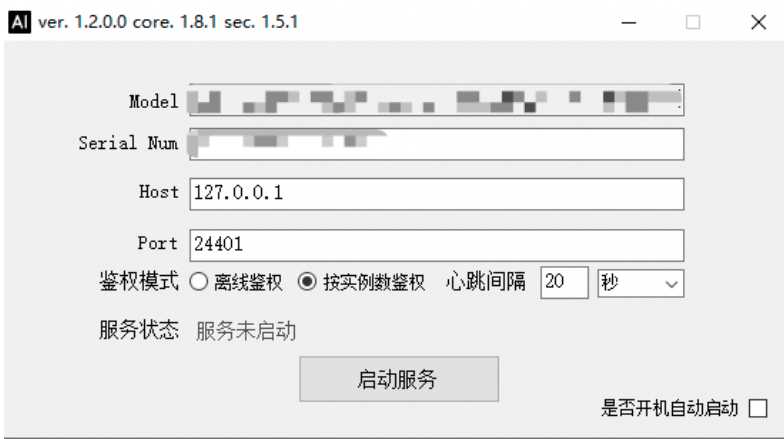
其他任何语言只需通过HTTP调用即可。

### 2.1 离线鉴权（默认鉴权模式）首次联网激活，后续离线使用



### 2.2 按实例数鉴权 周期性联网激活，离线后会释放所占用鉴权，启动时请确保心跳间隔小于等于生成序列号时填写的定期确认时间





基于源码集成时，若需要按实例数鉴权，需要通过代码指定使用按实例数鉴权

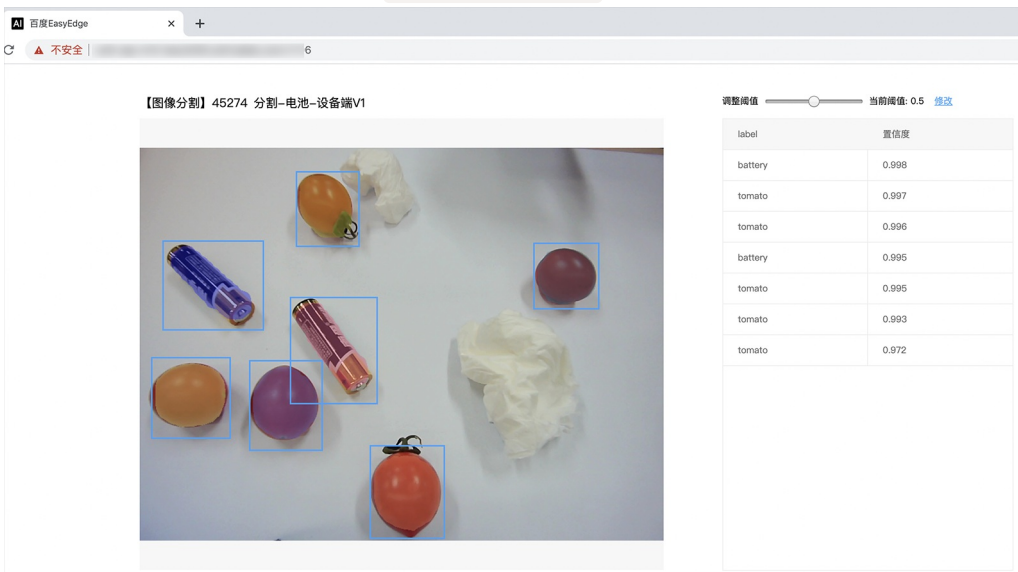
```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_AUTH_MODE, 2);
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL, 20);
```

或通过环境变量指定

```
set EDGE_CONTROLLER_KEY_AUTH_MODE=2
set EDGE_CONTROLLER_KEY_INSTANCE_UPDATE_INTERVAL=20
```

### 3. Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入 `http://127.0.0.1:24401`，在h5中测试模型效果。



使用说明

图像服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.mp4', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img).json()
```

C# 使用示例代码如下

```
    FileStream fs = new FileStream("./1.mp4", FileMode.Open);
    BinaryReader br = new BinaryReader(fs);
    byte[] img = br.ReadBytes((int)fs.Length);
    br.Close();
    fs.Close();
    string url = "http://127.0.0.1:8402?threshold=0.1";
    HttpRequest request = (HttpRequest)HttpRequest.Create(url);
    request.Method = "POST";
    Stream stream = request.GetRequestStream();
    stream.Write(img, 0, img.Length);
    stream.Close();

    HttpResponse response = request.GetResponse();
    StreamReader sr = new StreamReader(response.GetResponseStream());
    Console.WriteLine(sr.ReadToEnd());
    sr.Close();
    response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./1.mp4";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

结果 获取的结果存储在response字符串中。 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|----|-----| | confidence | float | 0~1 | 追踪的置信度 | | label | string | | 追踪的类别 | | index | number | | 追踪的类别 | | x1, y1 | float | 0~1 | 矩形的左上角坐标 (相对长宽的比例值) | | x2, y2 | float | 0~1 | 矩形的右下角坐标 (相对长宽的比例值) | | trackId | int | | 轨迹id | | frame | int | | 帧号 |

关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

$x_2$  \* 图片宽度 = 检测框的右下角的横坐标

$y_2$  \* 图片高度 = 检测框的右下角的纵坐标

## FAQ

### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
.NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是Python Intel Movidius版，需额外确保Opencv安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

### 4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL? 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted? Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

### 7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

**其他问题** 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

🔗 软硬一体方案

🔗 目标跟踪Jetson专用SDK集成文档

## 简介

本文档介绍EasyEdge/EasyDL的Jetson SDK的使用方法。Jetson SDK支持的硬件包括Jetson nano，Jetson TX2，Jetson AGX Xavier和Jetson Xavier NX。您可在[AI市场](#)了解Jetson相关系列产品，同时可以在[软硬一体方案](#)了解部署方案。

## 模型支持：

- EasyDL图像：图像分类高精度，图像分类高性能，物体检测高精度，物体检测均衡，物体检测高性能，目标跟踪单标签模型。
- BML：
  - 公开数据集预训练模型：SSD-MobileNetV1，YOLOv3-DarkNet，YOLOv3-MobileNetV1，ResNet50，ResNet101，SE-ResNeXt50，SE-ResNeXt101，MobileNetV2，EfficientNetB0\_small，EfficientNetB4，MobileNetV3\_large\_x1\_0，ResNet18\_vd，SE\_ResNet18\_vd，Xception71。
  - 百度超大规模数据集预训练模型：YOLOv3-DarkNet，MobileNetV3\_large\_x1\_0，ResNet50\_vd，ResNet101\_vd。
- EasyEdge：EasyEdge支持的模型较多，详见[查看模型网络适配硬件](#)。若模型不在此列表，可以尝试使用自定义网络生成端计算组件。

**软件版本支持** 使用EasyDL的Jetson系列SDK需要安装指定版本的JetPack和相关组件。所支持的JetPack版本会随着SDK版本的升级和新版本JetPack的推出而不断的更新。在使用SDK前请务必保证软件版本满足此处声明版本。目前所支持的JetPack版本包括：

- JetPack5.0.2

- JetPack5.0.1
- JetPack4.6
- JetPack4.5
- JetPack4.4 (deprecated, 该版本SDK会在未来某个版本移除, 请切换至新版本JetPack)
- JetPack4.2.2 (已移除, 请切换至新版本JetPack)

安装JetPack时请务必安装对应的组件：

- 使用SDK Manager安装JetPack需要勾选TensorRT、OpenCV、CUDA、cuDNN等选项。
- 使用SD Card Image方式（仅对Jetson Nano和Jetson Xavier NX有效）则无需关心组件问题，默认会全部安装。

**Release Notes** | 时间 | 版本 | 说明 | | --- | --- | --- | | 2022.12.29 | 1.7.2 | 新增支持JetPack5.0.2；缓存机制优化；模型性能优化 | | 2022.07.28 | 1.6.0 | 新增支持JetPack5.0.1，新增目标追踪接入实时流的demo | | 2022.05.18 | 1.5.0 | 部分模型切换格式，max\_batch\_size含义变更，由输入图片数不大于该值变更为等于该值；移除适用于JetPack4.2.2的SDK；示例代码demo\_stream\_inference重构；示例代码移除frame\_buffer，新增更安全高效的safe\_queue | | 2021.12.22 | 1.3.5 | 新增支持JetPack4.6；支持在EasyEdge平台语义分割模型生成开发套件；修复缓存问题；支持自定义缓存路径 | | 2021.10.20 | 1.3.4 | 新增支持JetPack4.5；大幅提升EasyDL有损压缩加速模型的推理速度 | | 2021.06.29 | 1.3.1 | 视频流支持分辨率调整；支持将预测后的视频推流，新增推流demo | | 2021.05.13 | 1.3.0 | 新增视频流接入支持；EasyDL模型发布新增多种加速方案选择；目标追踪支持x86平台的CPU、GPU加速版；展示已发布模型性能评估报告 | | 2021.03.09 | 1.2.1 | EasyEdge新增一系列模型的支持；性能优化 | | 2021.01.27 | 1.1.0 | EasyDL经典版高性能分类模型升级；

EasyDL经典版检测模型新增均衡选项；

EasyEdge平台新增Jetson系列端计算组件的生成；

问题修复 | | 2020.12.18 | 1.0.0 | 接口升级和一些性能优化 | | 2020.08.11 | 0.5.5 | 部分模型预测速度提升 | | 2020.06.23 | 0.5.4 | 支持JetPack4.4DP，支持EasyDL专业版更多模型 | | 2020.05.15 | 0.5.3 | 专项硬件适配SDK支持Jetson系列 |

2022-5-18: 【接口变更】 PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE含义变更。变更前：预测输入图片数不大于该值均可。变更后：预测输入图片数需等于该值。SDK内部对该接口变更做了兼容处理，在输入图片数小于该值时依然可正常运行，但预测性能会和等于该值时一致。推荐根据实际输入图片数量需求修改该值，尽可能保持最小。【版本移除】适用于JetPack4.4版本的SDK被标记为deprecated，SDK会在未来某个版本移除，建议切换至最新版本JetPack。适用于JetPack4.2.2版本的SDK被移除。

2020-12-18: 【接口升级】参数配置接口从1.0.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

2021-10-20: 【版本移除】适用于JetPack4.2.2版本的SDK被标记为deprecated，该版本代码已停止更新，SDK会在未来某个版本移除，请切换至新版本JetPack

**快速开始 安装依赖** 本SDK适用于JetPack4.5、JetPack4.6、JetPack5.0系列版本，请务必安装其中之一版本，并使用对应版本的SDK。注意在安装JetPack时，需同时安装CUDA、cuDNN、OpenCV、TensorRT等组件。

如已安装JetPack需要查询相关版本信息，请参考下文中的开发板信息查询与设置。

### 使用序列号激活

首先在官网获取序列号。



图片加载失败

将获取到的序列号填写到demo文件中或以参数形式传入。



图片加载失败

**编译并运行Demo** 模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

编译运行：

```
cd src
mkdir build && cd build
cmake ..
make -j$(nproc)
**make install 为可选，也可将lib所在路径添加为环境变量**
sudo make install
sudo ldconfig
./demo_batch_inference/easyedge_batch_inference {模型RES文件夹} {测试图片路径或仅包含图片的文件夹路径} {序列号}
```

demo运行示例：

```
baidu@nano:~/ljay/easydl/sdk/demo/build$ ./demo_batch_inference/easyedge_batch_inference ../../../../RES/
/ljay/images/mix008.jpeg
2020-08-06 20:56:30,665 INFO [EasyEdge] 548125646864 Compiling model for fast inference, this may take a while (Acceleration)
2020-08-06 20:57:58,427 INFO [EasyEdge] 548125646864 Optimized model saved to:
/home/baidu/.baidu/easyedge/jetson/mcache/24110044320/m_cache, Don't remove it
Results of image /ljay/images/mix008.jpeg:
2, kiwi, p:0.997594 loc: 0.352087, 0.56119, 0.625748, 0.868399
2, kiwi, p:0.993221 loc: 0.45789, 0.0730294, 0.73641, 0.399429
2, kiwi, p:0.992884 loc: 0.156876, 0.0598725, 0.3802, 0.394706
1, tomato, p:0.992125 loc: 0.523592, 0.389156, 0.657738, 0.548069
1, tomato, p:0.991821 loc: 0.665461, 0.419503, 0.805282, 0.573558
1, tomato, p:0.989883 loc: 0.297427, 0.439999, 0.432197, 0.59325
1, tomato, p:0.981654 loc: 0.383444, 0.248203, 0.506606, 0.400926
1, tomato, p:0.971682 loc: 0.183775, 0.556587, 0.286996, 0.711361
1, tomato, p:0.968722 loc: 0.379391, 0.0386965, 0.51672, 0.209681
Done
```

检测结果展示：



测试Demo HTTP 服务

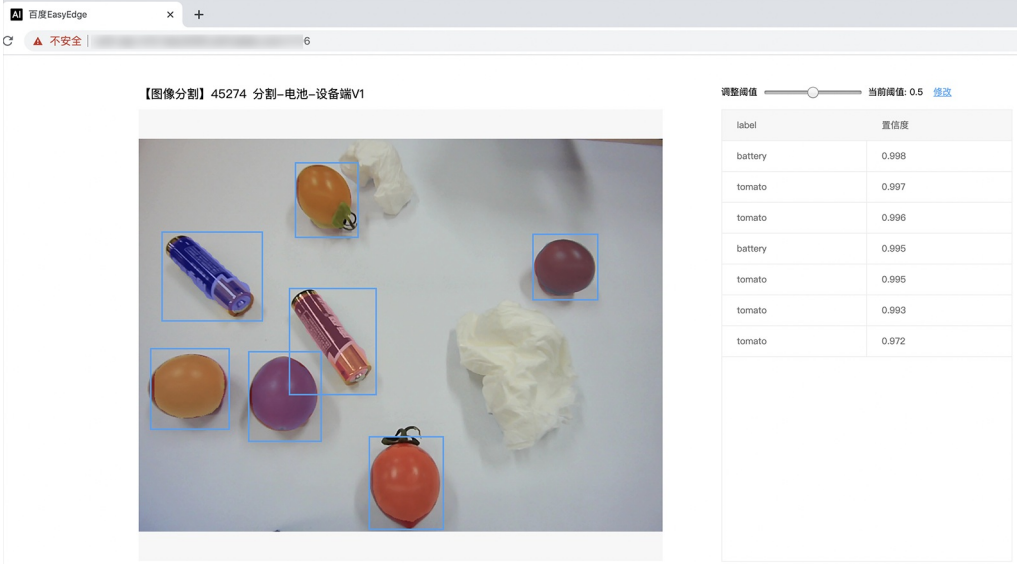
编译demo完成之后，会同时生成一个http服务，运行

```
**./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}**
./easyedge_serving ../../../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，http://{设备ip}:24401，选择图片来进行测试。



【图像分割】45274 分割-电池-设备端V1

调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972

同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

### 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型运行参数
EdgePredictorConfig config;
config.model_dir = model_dir;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, serial_num);
config.set_config(params::PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE, 1); // 优化的模型可以支持的batch_size
config.set_config(params::PREDICTOR_KEY_GTURBO_FP16, false); // 置true开启fp16模式推理会更快, 精度会略微降低, 但取决于硬件是否支持fp16, 不是所有模型都支持fp16, 参阅文档
config.set_config(params::PREDICTOR_KEY_GTURBO_COMPILE_LEVEL, 1); // 编译模型的策略, 如果当前设置的max_batch_size与历史编译存储的不同, 则重新编译模型

// step 2: 创建并初始化Predictor
auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

### 初始化接口

```

auto predictor = global_controller()->CreateEdgePredictor<EdgePredictorConfig>(config);
if (predictor->init() != EDGE_OK) {
    exit(-1);
}

```

若返回非0, 请查看输出日志排查错误原因。

### 预测接口



```
/**
 * @brief
 * 单图预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片预测接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image,
    std::vector<std::vector<EdgeResultData>>& results
) = 0;

/**
 * @brief
 * 批量图片预测接口，带阈值
 * @related infer(cv::Mat & image, EdgeColorFormat origin_color_format, std::vector<EdgeResultData> &result, float threshold)
 */
virtual int infer(
    std::vector<cv::Mat> &images,
    EdgeColorFormat origin_color_format,
    std::vector<std::vector<EdgeResultData>> &results,
    float threshold
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

批量图片的预测接口的使用要求在调用 `init` 接口的时候设置一个有效的 `max_batch_size`，其含义见下方参数配置接口的介绍。

**参数配置接口** 参数配置通过结构体EdgePredictorConfig完成。

```

struct EdgePredictorConfig {
    /**
     * @brief 模型资源文件夹路径
     */
    std::string model_dir;

    std::map<std::string, std::string> conf;

    EdgePredictorConfig();

    template<typename T>
    T get_config(const std::string &key, const T &default_value);

    template<typename T = std::string>
    T get_config(const std::string &key);

    template<typename T>
    const T *get_config(const std::string &key, const T *default_value);

    template<typename T>
    void set_config(const std::string &key, const T &value);

    template<typename T>
    void set_config(const std::string &key, const T *value);

    static EdgePredictorConfig default_config();
};

```

运行参数选项的配置以key、value的方式存储在类型为std::map的conf中，并且键值对的设置和获取可以通过EdgePredictorConfig的set\_config和get\_config函数完成。同时也支持以环境变量的方式设置键值对。EdgePredictorConfig的具体使用方法可以参考开发工具包中的demo工程。

针对Jetson开发工具包，目前EdgePredictorConfig的运行参数所支持的Key包括如下项：

```

/**
 * @brief 当有同类型的多个设备的时候，使用哪一个设备，如：
 * GPU: 使用哪张GPU卡
 * EdgeBoard(VMX), Movidius NCS : 使用哪一张加速卡
 * 值类型: int
 * 默认值: 0
 */
static constexpr auto PREDICTOR_KEY_DEVICE_ID = "PREDICTOR_KEY_DEVICE_ID";

/**
 * @brief 生成 batch_size 为 max_batch_size 的优化模型，单次预测图片数量可以小于或等于此值（推荐等于此值，见release notes）
 * 值类型: int
 * 默认值: 4
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE = "PREDICTOR_KEY_GTURBO_MAX_BATCH_SIZE";

/**
 * @brief 设置device对应的GPU卡可以支持的最大并发量
 * 实际预测的时候对应GPU卡的最大并发量不超过这里设置的范围，否则预测请求会排队等待预测执行
 * 值类型: int
 * 默认值: 1
 */
static constexpr auto PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY = "PREDICTOR_KEY_GTURBO_MAX_CONCURRENCY";

/**
 * @brief 是否开启fp16模式预测，开启后预测速度会更快，但精度会略有降低。并且需要硬件支持fp16
 * 值类型: bool
 * 默认值: false
 */
static constexpr auto PREDICTOR_KEY_GTURBO_FP16 = "PREDICTOR_KEY_GTURBO_FP16";

/**
 * @brief 模型编译等级
 * 1: 如果当前max_batch_size与历史编译产生的max_batch_size不相等时，则重新编译模型（推荐）

```

```

* 2：无论历史编译产生的max_batch_size为多少，均根据当前max_batch_size重新编译模型
* 值类型: int
* 默认值: 1
*/
static constexpr auto PREDICTOR_KEY_GTURBO_COMPILE_LEVEL = "PREDICTOR_KEY_GTURBO_COMPILE_LEVEL";

/**
 * @brief GPU工作空间大小设置
 * workspace_size = workspace_prefix * (1 << workspace_offset)
 * workspace_offset: 10 = KB, 20 = MB, 30 = GB
 * 值类型: int
 * 默认值: WORKSPACE_PREFIX: 100, WORKSPACE_OFFSET: 20, 即100MB
 */
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX = "PREDICTOR_KEY_GTURBO_WORKSPACE_PREFIX";
static constexpr auto PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET = "PREDICTOR_KEY_GTURBO_WORKSPACE_OFFSET";

/**
 * @brief 需要使用的dla core
 * 值类型: int
 * 默认值: -1(不使用)
 */
static constexpr auto PREDICTOR_KEY_GTURBO_DLA_CORE = "PREDICTOR_KEY_GTURBO_DLA_CORE";

/**
 * @brief 自定义缓存文件存储路径
 * 值类型: string
 * 默认值: ~/.baidu/easyedge/mcache/{model_id * 1000000 + release_id}
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_DIR = "PREDICTOR_KEY_GTURBO_CACHE_DIR";

/**
 * @brief 自定义缓存文件命名，默认即可
 * 值类型: string
 * 默认值: 根据配置自动生成
 */
static constexpr auto PREDICTOR_KEY_GTURBO_CACHE_NAME = "PREDICTOR_KEY_GTURBO_CACHE_NAME";

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型: string
 * 默认值: 空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

**PREDICTOR\_KEY\_GTURBO\_CACHE\_NAME**：首次加载模型会先对模型进行编译优化，通过此值可以设置优化后的产出文件名。

**PREDICTOR\_KEY\_GTURBO\_CACHE\_DIR**：首次加载模型经过编译优化后，产出的优化文件会存储在这个位置，可以按需修改。

**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_PREFIX**、**PREDICTOR\_KEY\_GTURBO\_WORKSPACE\_OFFSET**：设置运行时可以被用来使用的最大临时显存。

**PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE**：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数需等于此值。

**PREDICTOR\_KEY\_DEVICE\_ID**：设置需要使用的 GPU 卡号，对于 Jetson，此值无需更改。

**PREDICTOR\_KEY\_GTURBO\_COMPILE\_LEVEL**：模型编译等级。通常模型的编译会比较慢，但编译产出是可以复用的。可以在第一次加载模型的时候设置合理的 `max_batch_size` 并在之后加载模型的时候直接使用历史编译产出。是否使用历史编译产出可以通过此值 `compile_level` 来控制，当此值为 0 时，表示忽略当前设置的 `max_batch_size` 而仅使用历史产出（无历史产出时则编译模型）；当此值为 1 时，会比较历史产出和当前设置的 `max_batch_size` 是否相等，如不等，则重新编译；当此值为 2 时，无论如何都会重新编译模型。

**PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY**：通过此值设置单张 GPU 卡上可以支持的最大 infer 并发量，其上限取决于硬件限制。init 接口会根据此值预分配 GPU 资源，建议结合实际使用控制此值，使用多少则设置多少。注意：此值的增加会降低单次 infer 的速度，建议优先考虑 batch inference。

**PREDICTOR\_KEY\_GTURBO\_FP16**：默认是 fp32 模式，置 true 可以开启 fp16 模式预测，预测速度会有所提升，但精度也会略微下降，权衡使用。注意：不是所有模型都支持 fp16 模式，也不是所有硬件都支持 fp16 模式。已知不支持 fp16 的模式包括：EasyDL 图像分类高精度模型。

## 预测视频接口

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

class `VideoDecoding` :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct `VideoConfig`

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;           // 输入源类型
    std::string source_value;         // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};               // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};          // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};                 // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};          // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;            // frame存储为视频文件的路径
    bool save_all{false};             // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

`source_type`：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。`source_value`：若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。`skip_frames`：设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。`retrieve_all`：若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。`input_fps`：用于抽帧前设置fps。`resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。`conf`：高级选项。部分配置会通过该map来设置。

#### 注意：

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过resolution设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。

具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

#### 返回格式

预测成功后，从 `EdgeResultData`中可以获得对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测或图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};

```

### 关于矩形坐标

x1 图片宽度 = 检测框的左上角的横坐标 y1 图片高度 = 检测框的左上角的纵坐标 x2 图片宽度 = 检测框的右下角的横坐标 y2 图片高度 = 检测框的右下角的纵坐标

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### http服务

1. 开启http服务 http服务的启动参考demo\_serving.cpp文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里, 图片的解码运行在cpu之上, 可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量, 根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

### 2. 请求http服务

开发者可以打开浏览器, `http://{设备ip}:24401`, 选择图片来进行测试。

URL中的get参数:

参数	说明	默认值
threshold	阈值过滤, 0~1	如不提供, 则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()
```

Java请求示例参考[这里](#)

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考接口使用-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

**多线程预测** Jetson 系列 SDK 支持多线程预测，创建一个 predictor，并通过 PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY 控制所支持的最大并发量，只需要 init 一次，多线程调用 infer 接口。需要注意的是多线程的启用会随着线程数的增加而降低单次 infer 的推理速度，建议优先使用 batch inference 或权衡考虑使用。

#### 已知问题 1. 多线程时图片按线程分配不均 或 不同batch size的图片交叉调用infer接口时，部分结果错误

A：EasyDL图像分类高精度模型在有些显卡上可能存在此问题，可以考虑填充假图片数据到图片比较少的线程或batch以使得infer间的图片绝对平均。

#### 2. 显存持续增长或遇到 terminate called after throwing an instance of 'std::runtime\_error' what(): Failed to create object

A：如果遇到此问题，请确认没有频繁调用 init 接口，通常调用 infer 接口即可满足需求。

#### 3. 开启 fp16 后，预测结果错误

A：不是所有模型都支持 fp16 模式。目前已知的不支持fp16的模型包括：EasyDL图像分类高精度模型。目前不支持的将会在后面的版本陆续支持。

#### 4. 部分模型不支持序列化

A：针对JetPack4.4、4.5版本，部分模型无法使用序列化，如已知的BML的MobileNetV1-SSD和物体检测高性能模型。需要每次加载模型的时候

编译模型，过程会比较慢。此问题将在后续JetPack版本中修复。目前JetPack4.6版本SDK已修复该问题。

**开发板信息查询与设置 查询L4T或JetPack版本** 查询JetPack版本信息，可以通过下面这条命令先查询L4T的版本。

```
**在终端输入如下命令并回车**
$ head -n 1 /etc/nv_tegra_release
**就会输出类似如下结果**
$ # # R32 (release), REVISION: 4.3, GCID: 21589087, BOARD: t210ref, EABI: aarch64, DATE: Fri Jun 26 04:38:25 UTC 2020
```

从输出的结果来看，板子当前的L4T版本为R32.4.3，对应JetPack4.4。注意，L4T的版本不是JetPack的版本，一般可以从L4T的版本唯一对应到JetPack的版本，下面列出了最近几个版本的对应关系：

```
L4T R32.6.1 --> JetPack4.6
L4T R32.5.1 --> JetPack4.5.1
L4T R32.5 --> JetPack4.5
L4T R32.4.3 --> JetPack4.4
L4T R32.4.2 --> JetPack4.4DP
L4T R32.2.1 --> JetPack4.2.2
L4T R32.2.0 --> JetPack4.2.1
```

**功率模式设置与查询** 不同的功率模式下，执行AI推理的速度是不一样的，如果对速度需求很高，可以把功率开到最大，但记得加上小风扇散热~

```
**1. 运行下面这条命令可以查询开发板当前的运行功率模式**
$ sudo nvpmode -q verbose
**$ NV Power Mode: MAXN**
**$ 0**
**如果输出为MAXN代表是最大功率模式**

**2. 若需要把功率调到最大，运行下面这条命令**
$ sudo nvpmode -m 0

**如果你进入了桌面系统，也可以在桌面右上角有个按钮可以切换模式**

**3. 查询资源利用率**
$ sudo tegrastats
```

#### FAQ 1. EasyDL SDK与云服务效果不一致，如何处理？

后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

#### 2. 运行SDK报错 Authorization failed 日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

#### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

#### 4. 运行demo时报找不到libeasyedge\_extension.so

需要export libeasyedge\_extension.so所在的路径，如路径为/home/work/baidu/cpp/lib，则需执行：

```
export LD_LIBRARY_PATH=/home/work/baidu/cpp/lib:${LD_LIBRARY_PATH}
```

或者在编译完后执行如下命令将lib文件安装到系统路径：

```
sudo make install
```



如不能安装，也可手动复制lib下的文件到/usr/local/lib下。

## 5. 运行demo时报如下之一错误

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Compiling model for fast inference, this may take a while (Acceleration) Killed
```

\*\*或\*\*

```
2020-12-17 16:15:07,924 INFO [EasyEdge] 547633188880 Build graph failed
```

请适当降低PREDICTOR\_KEY\_GTURBO\_MAX\_BATCH\_SIZE和PREDICTOR\_KEY\_GTURBO\_MAX\_CONCURRENCY的值后尝试。

**6. 运行有损压缩加速的模型，运算精度较标准模型偏低** 首先请保证数据集不会太小。其次可以通过将模型目录RES中的calibrationtable移除，并通过将PREDICTOR\_KEY\_GTURBO\_FP16设置为true，使用FP16的运算精度重新评估模型效果。若依然不理想，可将calibrationtable移除并将PREDICTOR\_KEY\_GTURBO\_FP16设置为false,从而使用更高精度的FP32的运算精度。

# EasyDL 结构化数据使用说明

## EasyDL结构化数据介绍

### 简介

Hi，您好，欢迎使用百度EasyDL结构化数据

目前EasyDL结构化数据支持训练以下模型：

- 表格数据预测

通过机器学习技术从表格化数据中发现潜在规律，从而创建机器学习模型，并基于机器学习模型处理新的数据，为业务应用生成预测结果

- 时序预测

通过机器学习技术从历史数据中发现潜在规律，从而对未来的变化趋势进行预测。

### 可视化操作

无需机器学习专业知识，模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型

### 操作步骤

#### Step 1 创建模型

确定模型名称，记录希望模型实现的功能

#### Step 2 上传并标注数据

#### Step 3 训练模型并校验效果

选择部署方式与算法，用上传的数据一键训练模型

模型训练完成后，可在线校验模型效果

#### Step 4 发布模型

根据训练时选择的部署方式，将模型以云端API的方式发布使用

更详细的操作指导，请参考各类模型的技术文档

## 表格数据预测

### 表格数据预测介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

定制表格数据预测模型，旨在帮助用户通过机器学习技术从表格化数据中发现潜在规律，从而创建机器学习模型，并基于机器学习模型处理新的数据，为业务应用生成预测结果。本文介绍表格数据预测模型，根据预测数据的不同，可以分为如下几种类型：

- 回归：目标列是连续的实数范围，或者属于某一段连续的实数区间。如在销量预测场景中，销量值可能是某个取值范围内的任意值，解决该问题的模型属于回归模型。
- 二分类：目标列是离散值，且只有两种可能的取值。如在精准营销场景中预测一个用户是否为潜在购买用户，其目标列仅存在“True”和“False”两种取值，解决该问题的模型属于二分类模型。
- 多分类：目标列是离散值，并具有有限的可能取值。如在用户分类场景中，根据用户的历史消费数据，将用户划分到不同消费偏好的类别中，解决该问题的模型属于多分类模型。

以下是关于表格数据预测模型的技术文档。

## 应用场景

- 精准营销：从客户消费记录中挖掘客户群的共有特征，分析出客户的购物偏好，从而实现广告的精准投放
- 信用评分：金融公司分析客户的历史行为数据，建立用户信用模型，从而确定贷款额度等
- 价格预测：从历史数据中发现商品的变化规律以及影响价格的因素，从而为未来的商业行为提供支持
- 客户流失预测：根据客户历史数据获得数据挖掘模型，从而生成客户流失预测列表，为市场营销策略提供有价值的业务洞察力。
- 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作。在数据已经准备好的情况下，最快几分钟即可获得定制模型。

下面将详细介绍每一步的操作方式和注意事项。如果文档没有解决您的问题，请在百度云控制台内[提交工单](#)反馈。



## 数据准备

### 表格数据集介绍

#### 表格数据介绍

训练数据的质量决定了训练所得模型效果可达到的上限。数据上传后无法修改其内容。如果在导入训练数据后需要对其进行更改，必须重新导入。

#### 数据要求

##### 数据文件格式要求：

- 目前仅支持CSV格式的数据文件
- 一次仅能上传一个文件，可以是一个CSV文件或由多个CSV文件压缩成的zip包
- 单个上传文件大小不能超过5GB
- 一个数据集包含的总文件大小不能超过20GB

##### 数据文件内容要求：

- 当数据文件包含列名时，列名称可以包含字母、数字和下划线（\_），但不能以下划线开头。
- 文件内容以换行符（即字符“\n”，或称为LF）分隔各行，行内容以英文逗号（即字符“,”）分隔各列
- 必须包含要预测的值即目标列，且目标列的数据类型会决定模型的类型。
- 文件中文本列取值长度不能超过4096个字符。
- 必须至少包含两列，且不得超过1000列。
- 数据集的总行数不能超过1000万行。

- zip包中的多个CSV文件必须使用相同的编码格式，都包含列名或都不包含列名；且列的顺序必须保持一致
- 在扩充数据集时，新导入数据文件的首行与数据集的列名相同时，将被视为列名，否则将被视作数据

## 🔗 创建表格数据集

在EasyDL经典版中，您可以在“数据总览”页面，完成数据集创建、数据导入等操作，为模型构建准备好数据。

### 创建数据集

数据集需要先定义，然后再导入数据。

1. 单击“数据总览”，进入数据集列表页面。
2. 单击“创建数据集”，进入数据集创建页面。
3. 输入数据集名称，单击“完成”结束创建。

完成创建后，可以在数据集列表中查看新建的数据集。系统默认生成V1版本，当前数据内容为空，可以通过导入的方式向其中添加数据。

### 导入数据

通过导入的方式可以向数据集中添加或追加数据。

1. 单击“数据总览”，进入数据集列表页面。
2. 单击待导入数据集所在行的“导入”按钮，进入数据集导入页面。
3. 导入数据文件。

导入的数据文件可以是CSV文件或由CSV文件组成的压缩包文件。

如果导入的是CSV文件，支持数据预览，如果是压缩包格式，则不支持预览。

4. 根据数据文件的实际情况进行列名设置。
  - 设置首行为列名：将导入的数据文件中的首行作为列名。
  - 设置首行非列名：此时系统会自动生成列名，而将首行作为数据。
5. 单击“确认并返回”完成导入操作。

## 模型训练

### 🔗 表格预测模型介绍

#### 表格数据预测模型介绍

表格数据预测模型是基于结构化数据进行建模，系统会基于用户上传的数据使用预置算法进行模型构建与训练。表格数据预测模型目前支持回归和分类两种类型的模型，其中分类模型包括二分类和多分类模型。

#### 回归

回归模型通常用来预测一个数值，其反映的是变量或属性间的依赖关系，建模过程即求解将一个或多个变量映射到一个实数值的函数。它可以应用到市场营销的各个方面，如销量预测、价格预测等场景中。

#### 分类

分类是找出一组对象的共同或差异点以将其划分为相同或不同的类，其目的是通过分类模型，将数据项映射到某个给定的类别。它可以应用到客户分群、客户行为预测、客户满意度分析等场景中。

其中二分类模型是指预测值包括两种类别，多分类模型是指预测值包括多种类别。

### 🔗 创建模型

在EasyDL经典版中，您可以在“模型中心”进行模型的创建。在EasyDL中模型可以包括多个版本，每次训练会生成一个版本。各个版本的模型之间相互独立，可以分别进行版本发布等操作。

#### 创建模型

模型需要先创建，然后才能进行训练。

1. 单击“创建模型”，进入模型创建页面。

## 2. 填写模型创建信息，如下图所示。

The screenshot shows the 'Create Model' interface in the EasyDL console. The page title is '经典版-表格数据预测模型' (Classic Edition - Table Data Prediction Model). Below the title is a brief description: '定制基于表格数据的模型，可实现表格中某列类别或数值的预测，该类模型可应用在销量预测、授信评估等场景。此类模型训练速度较快，具有X个特征列的Y条样本的表格数据，一般可在30分钟内训练完毕。' (Customize a model based on table data, which can realize the prediction of categories or values in a certain column of the table. This type of model can be applied in scenarios such as sales prediction and credit assessment. The training speed of this type of model is relatively fast. With X feature columns and Y samples of table data, it can generally be trained within 30 minutes.)

The interface includes a sidebar on the left with navigation options: '我的模型' (My Models), '创建模型' (Create Model), '训练模型' (Train Model), '校验模型' (Validate Model), '发布模型' (Publish Model), 'EasyData数据服务' (EasyData Data Service), and '数据总览' (Data Overview). The main content area is titled '模型列表 > 创建模型' (Model List > Create Model). It contains a form with the following fields:

- 模型类别: 表格数据预测 (Model Category: Table Data Prediction)
- \* 模型名称: [Text Input Field]
- 模型归属: [Radio Buttons for '公司' (Company) and '个人' (Personal)]
- 请输入公司名称: [Text Input Field]
- \* 所属行业: [Dropdown Menu]
- \* 应用场景: [Dropdown Menu]
- \* 邮箱地址: [Text Input Field with placeholder '\*\*\*\*@34.com']
- \* 联系方式: [Text Input Field with placeholder '111\*\*\*\*111']
- \* 功能描述: [Text Area with a 0/500 character count]

A '下一步' (Next Step) button is located at the bottom of the form.

- 模型名称：指定模型的名称
- 模型归属：公司或个人，并输入相关名称
- 所属行业：请根据实际情况进行选择
- 应用场景：请根据实际情况进行选择
- 邮箱地址：请根据实际情况进行设置
- 联系方式：请根据实际情况进行设置
- 功能描述：用于记录模型创建的背景、用途等方面的信息。

## 3. 单击“下一步”完成模型创建。

完成创建后，可以在模型列表中查看新建的模型。新建的模型不包含任何版本的模型，在训练后会生成新的模型版本。

## 🔗 训练模型

EasyDL经典版提供的的机器学习算法不仅性能高、可扩展，还针对速度、规模和准确性进行了优化，可以在大规模数据集上进行训练。

### 训练模型

在准备好数据集并创建模型后，可以创建训练任务。

1. 单击“训练模型”，进入模型训练页面。
2. 填写模型训练信息，如下图所示。



- 选择模型：选择要训练的模型
- 选择数据集：选择训练模型使用的数据集
- 选择目标列：从数据集中选择一列作为预测列
- 算法类型：包括二分类、多分类和回归，也可以选择自动，此时系统会根据数据集以及选择的目标列进行判断。
- 部署方式：当前仅支持公有云API方式。

### 3. 单击“开始训练”启动训练任务。

启动训练任务后，系统会在模型下的列表中创建一个新的模型版本，新建的版本处于“训练中”的状态，当处于“训练完成”状态时表示模型已完成训练。

### 4. 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有表格预测操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 查看模型

训练任务结束后，可以查看模型的各项指标，以确定模型是否满足要求。不同类型的模型包含了不同的指标，用户可以根据实际的业务场景查看关键指标是否满足要求。

#### 查看模型

对于“训练完成”的模型，可以查看其评估结果。

1. 单击“我的模型”，进入模型列表页面。
2. 单击待查看模型的“历史版本”，进入模型版本列表页面。
3. 点击待查看模型版本所在行的“完整评估结果”，系统展示评估结果页面。
  - 回归模型的评估结果包括各项常用评估指标以及特征重要性，如下所示：

模型管理 > diamonds价格预测

diamonds价格预测评估报告

部署方式: 公有云API (目前仅支持) 版本: V1

整体评估

特征 10列 | 目标列 price | 算法类型 回归

diamonds价格预测 V1整体效果欠佳, 建议针对识别错误的样本示例继续优化模型效果。 [如何优化效果?](#)

名称	数值
MAE (Mean Absolute Error) 平均绝对误差	6.192
MSE (Mean Squared Error) 均方误差	1519.512
MAPE (Mean Absolute Percentage Error) 平均绝对百分比误差	0.73%
R2 Score (决定系数) 回归得分函数	1.000

详细评估

特征重要性 最多展示重要性Top15的特征

- 二分类模型的评估结果包括混淆矩阵、F1-Score阈值曲线、KS曲线、ROC曲线、P-R曲线、Lift曲线、Gain曲线和特征重要性，如下所示：

部署方式: 公有云API (目前仅支持) 版本: V1

训练时长: 7分钟

整体评估

特征 20列 | 目标列 TARGET\_5Yrs | 算法类型 二分类

以 1 为正值

整体评估: 阈值 0.5 恢复推荐阈值

准确率 70.5%

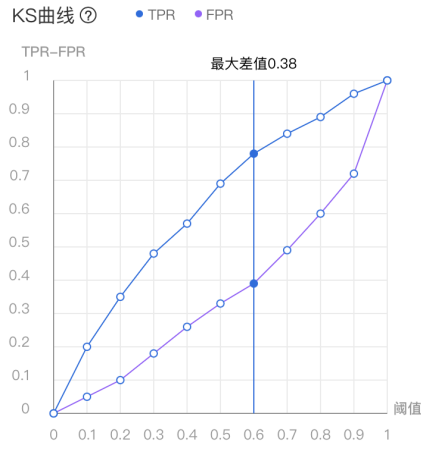
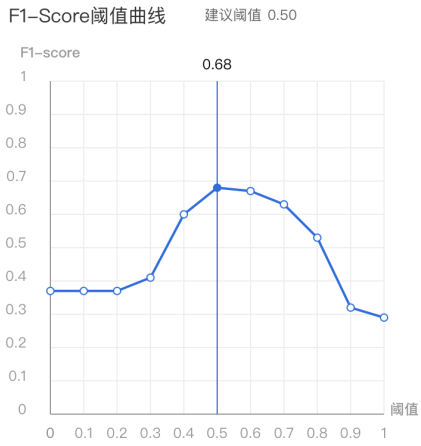
F1-score 67.8%

精确率 71.7%

召回率 83.3%

混淆矩阵

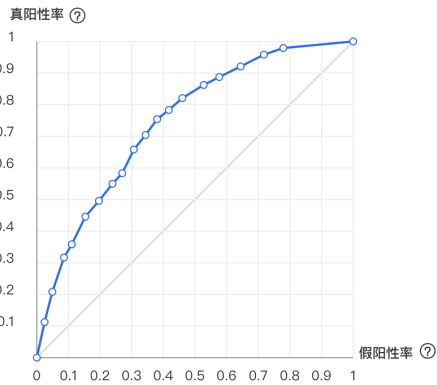
真实值 \ 预测值	1	0
	1	83.33%
0	48.47%	51.53%



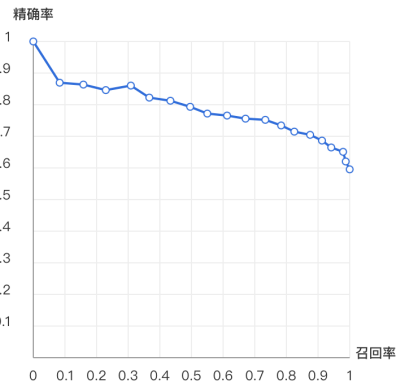
详细评估：

## 详细评估

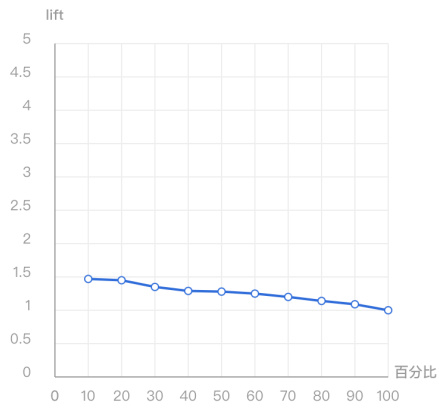
ROC曲线 (AUC 值为 0.746)



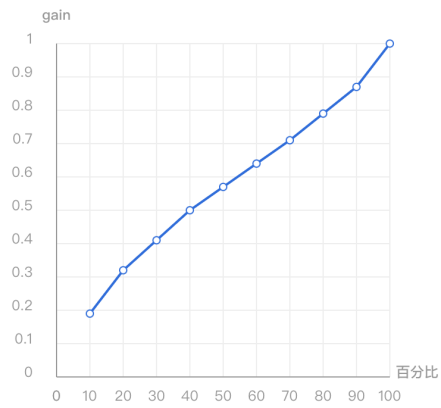
P-R曲线



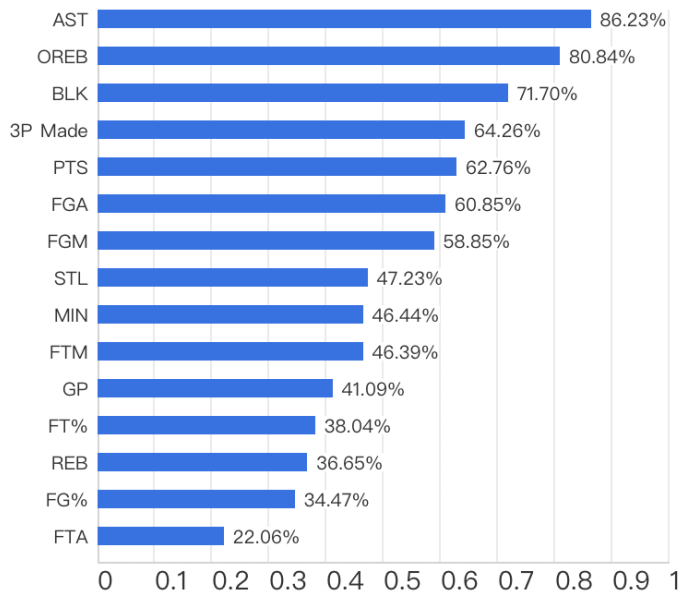
Lift曲线



Gain曲线



### 特征重要性 <sup>?</sup> 最多展示重要性Top15的特征



- 多分类模型的评估结果包括混淆矩阵、ROC曲线、P-R曲线和特征重要性，如下所示：

#### 整体评估

特征 36列 | 目标列 Class | 算法类型 多分类



混淆矩阵 <sup>?</sup>

整体评估：

真实值 \ 预测值	1	5	2	3	7	4
1	98.30%	0.21%	0.00%	1.49%	0.00%	0.00%
5	4.17%	91.15%	1.04%	1.04%	2.60%	0.00%
2	0.00%	0.43%	98.70%	0.43%	0.00%	0.43%
3	0.24%	0.00%	0.00%	96.12%	1.21%	2.43%
7	0.00%	2.10%	0.00%	2.10%	93.46%	2.34%
4	0.51%	1.52%	0.00%	20.71%	12.63%	64.65%

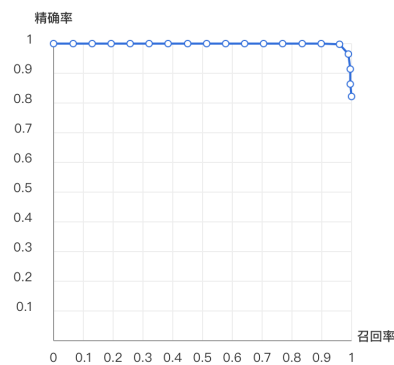
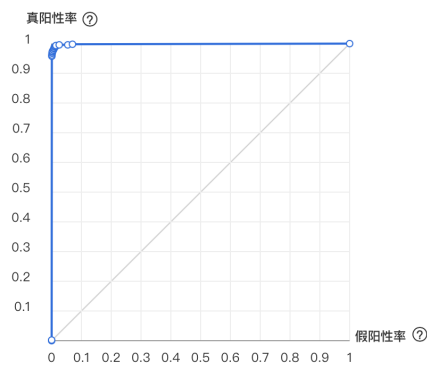
详细

Kappa值 <sup>?</sup> 0.90900776713592 几乎完全一致 (参考区间0.81-1.00 几乎完全一致) 如何优化效果 <sup>?</sup>

ROC曲线 <sup>?</sup> (AUC <sup>?</sup> 值为 0.998)

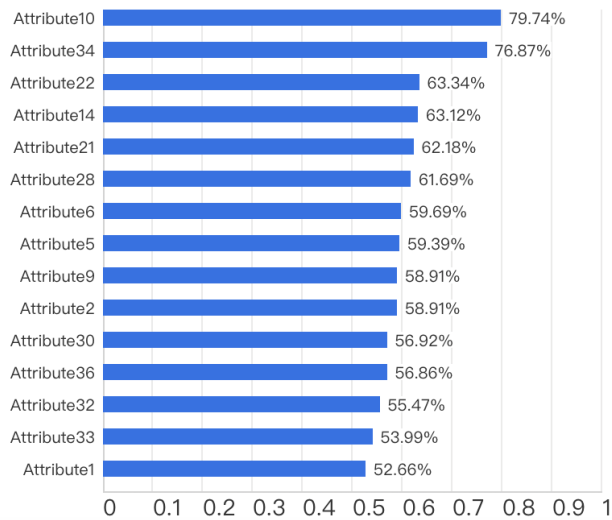
P-R曲线 <sup>?</sup>

评估：





## 特征重要性 ② 最多展示重要性Top15的特征



## 校验模型

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。

## 校验模型

在训练任务成功完成后，即可使用实际数据进行校验。

1. 单击“校验模型”，进入模型校验页面。
2. 选择要校验的模型及其版本。
3. 单击“启动模型校验服务”。
  - 校验数据支持两种输入方式，表单方式或json格式，并支持切换
  - 系统自动填充了校验数据，用户可以直接使用预置的数据进行预测，也可以修改后再进行预测。
4. 单击“预测”，可以在右侧结果面板中查看预测结果。

校验示例如下所示：

**校验模型**

选择模型: iris分类 部署方式: 公有云API (目前仅支持) 选择版本: V1

当前模型准确率 97.83% [评估报告](#) [识别结果](#) [如何优化效果?](#)

预测数据		
字段名	类型	取值
sepal_length	数值	<input type="text" value="5.2"/>
sepal_width	数值	<input type="text" value="3.5"/>
petal_length	数值	<input type="text" value="1.4"/>
petal_width	数值	<input type="text" value="0.2"/>

```

1  {
2  "spec...:  "setosa"
3  }

```

预测
申请上线

## 模型发布

## 模型发布整体说明

训练完成后，可将模型部署在公有云服务器上，通过API进行调用。

### 公有云API

- 训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求
- 一次API请求中最多可以包含100条预测数据

### 相关费用

将模型发布为API后，将获得1000次免费调用次数，如需更多调用量，请在百度云控制台内[提交工单](#)反馈。

#### 公有云部署

#### 如何发布表格数据预测API

训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可发布公有云API：

- 选择模型
- 选择部署方式「公有云部署」
- 选择版本
- 自定义服务名称、接口地址后缀
- 申请发布

申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工单](#)反馈。

发布模型界面示意：

#### 发布模型

选择模型：

部署方式：

选择版本：

\* 服务名称：

\* 接口地址：

其他要求：

0/500

[提交申请](#)

#### 标准接口规范参考

标准接口请求参考说明：

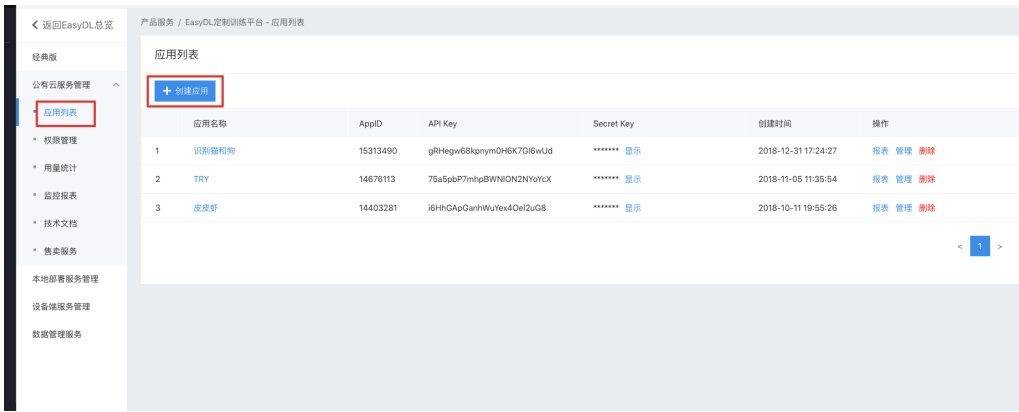
字段名称	必须	类型	说明
include_req	否	boolean	返回结果是否包含特征数据；false，不包含；true，包含，默认为false
data	是	array(object)	待预测数据，每条待预测数据是由各个特征及其取值构成的键值对的集合

标准接口响应字段说明：

字段名称	必须	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
results	否	array(object)	预测结果数组

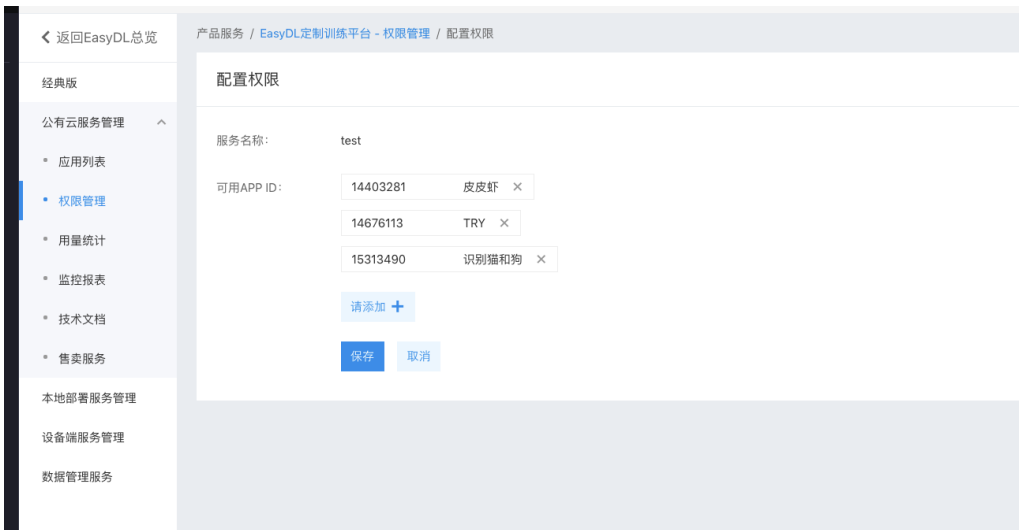
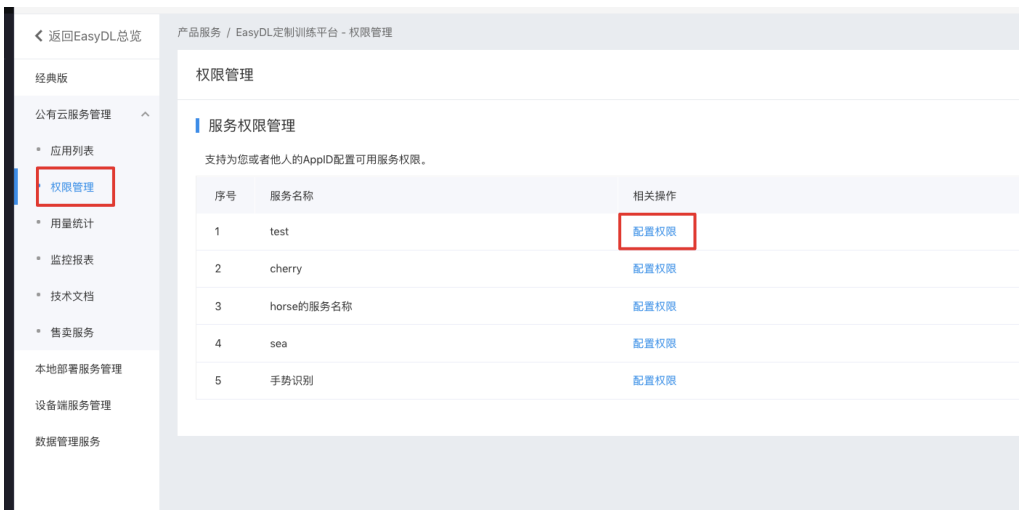
### 接口赋权

在正式使用之前，还需要做的一项工作为接口赋权，需要登录[EasyDL经典版控制台](#)中创建一个应用，获得由一串数字组成的appid，然后就可以参考接口文档正式使用了



同时支持在「公有云服务管理」-「权限管理」中为第三方用户配置权限

示意图如下：



### API调用文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

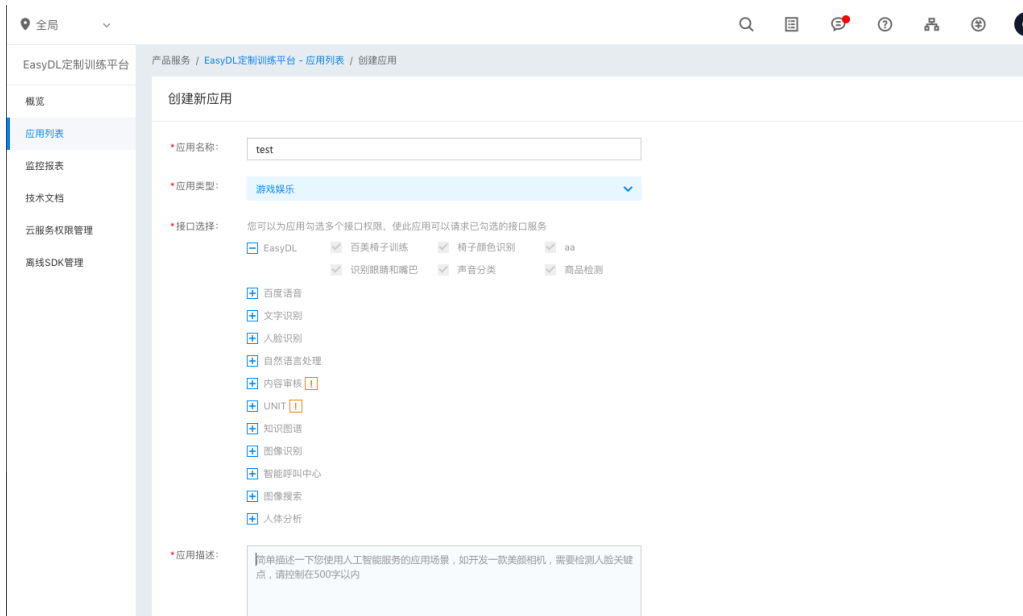
- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### 接口描述

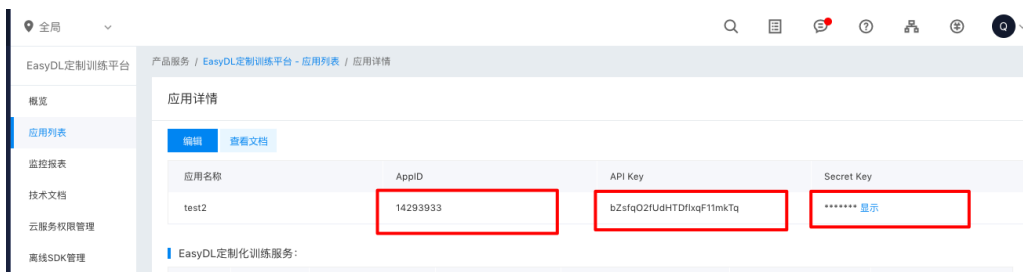
基于自定义训练出的表格数据预测模型，实现表格数据预测。模型训练完毕后发布可获得定制化表格数据预测API

### 接口鉴权

- 1、在[EasyDL经典版控制台](#)创建应用



## 2、应用详情页获取AK SK



### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求接口：

```
{
  "include_req": false,
  "data": <待预测数组>
}
```

考虑到表格字段内容和长度的不固定性，我们建议您参考“校验服务”页面提供的详细信息。您可以访问该页

面：<https://ai.baidu.com/easydl/app/validate/ml/models/verify>，启动校验服务后，点击“复制”按钮从中复制数据请求的Body部分作为参考模板。这将帮助您理解如何灵活处理各种不同的字段。

以如下数据特征列为例，请求body格式为：

```
{
  "include_req": false,
  "data": [{
    "sepal_length": 5.1,
    "sepal_width": 3.5,
    "petal_length": 1.4,
    "petal_width": 0.2
  },
  {
    "sepal_length": 5.3,
    "sepal_width": 3.4,
    "petal_length": 1.2,
    "petal_width": 0.3
  }
  ]
}
```

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
include_req	否	boolean	-	返回结果是否包含特征数据：false，不包含；true，包含，默认为false
include_proba	否	boolean	-	返回结果是否包含分类概率：false，不包含；true，包含，默认为false
data	是	array	-	待预测数据，每条待预测数据是由各个特征及其取值构成的键值对的集合

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
batch_result	否	array(object)	预测结果数组

#### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、data格式错误等等，可检查下请求数据格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## ☞ 服务器部署

## ☞ Linux集成文档-Python

### 简介

本文档介绍 EasyDL 的 Linux Python SDK 的使用方法，适用于 EasyDL 和 BML。

EasyDL 通用版：

- 网络类型支持：图像分类，物体检测，图像分割，声音分类，表格预测
- 硬件支持：
  - Linux x86\_64 CPU (基础版，加速版)
  - Linux x86\_64 Nvidia GPU (基础版，加速版)
- 语言支持：Python 3.5, 3.6, 3.7, 3.8, 3.9

BML：

- 网络类型支持：图像分类，物体检测，声音分类
- 硬件支持：
  - Linux x86\_64 CPU (基础版)

- Linux x86\_64 Nvidia GPU (基础版)
- 语言支持：Python 3.5, 3.6, 3.7, 3.8, 3.9

## Release Notes

时间	版本	说明
2022.10.27	1.3.5	新增华为Atlas300、飞腾Atlas300 Python SDK，支持图像分类、物体检测、人脸检测、实例分割
2022.09.15	1.3.3	EasyDL CPU普通版新增支持表格预测
2022.05.27	1.3.1	CPU、GPU普通版新增支持BML Cloud小目标检测模型
2021.12.22	1.2.7	声音分类模型升级
2021.10.20	1.2.6	CPU基础版、CPU加速版、GPU基础版推理引擎优化升级
2021.08.19	1.2.5	CPU基础版、CPU无损加速版、GPU基础版新增支持EasyDL小目标检测
2021.06.29	1.2.4	CPU、GPU新增EasyDL目标跟踪支持；新增http server服务启动demo
2021.03.09	1.2.2	EasyDL CPU加速版新增支持分类、高性能检测和均衡检测的量化压缩模型
2021.01.27	1.2.1	EasyDL经典版分类高性能模型升级；支持更多模型
2020.12.18	1.2.0	推理引擎升级；接口升级；性能优化
2020.09.17	1.1.19	支持更多模型
2020.08.11	1.1.18	性能优化
2020.06.23	1.1.17	支持更多EasyDL专业版模型
2020.04.16	1.1.15	技术优化；升级 OpenVINO 版本
2020.03.12	1.1.14	新增声音识别python sdk
2020.02.12	1.1.13	新增口罩模型支持
2020.01.16	1.1.12	预测函数默认使用推荐阈值
2019.12.26	1.1.11	EasyDL 专业版支持 SDK 加速版
2019.12.04	1.1.10	支持图像分割
2019.10.21	1.1.9	支持 EasyDL 专业版
2019.08.29	1.1.8	CPU 加速版支持
2019.07.19	1.1.7	提供模型更新工具
2019.05.16	1.1.3	NVIDIA GPU 支持
2019.03.15	1.1.0	架构与功能完善
2019.02.28	1.0.6	引擎功能完善
2019.02.13	1.0.5	paddlepaddle 支持
2018.11.30	1.0.0	第一版！

2020-12-18: 【接口升级】序列号的配置接口从1.2.0版本开始已升级为新接口，以前的方式被置为deprecated，并将在未来的版本中移除。请尽快考虑升级为新的接口方式，具体使用方式可以参考下文介绍以及demo工程示例，谢谢。

## 快速开始

### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。
- 使用声音分类SDK需要安装额外依赖
  - \* pip 安装 `resampy pydub six librosa` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装 `ffmpeg` (windows系统的ffmpeg已基于sdk中无需额外安装，linux系统需要手动安装)
- 使用表格预测SDK需要安装额外依赖 `pip安装 brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3`

```
catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2
```

```
psutil==5.7.2 pymml==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0
```

另外ml算法安装（目前只支持python3.7）

```
pip install BaiduAI_TabularInfer-0.0.0-cp37-cp37m-linux_x86_64.whl 安装 paddlepaddle
```

- 使用x86\_64 CPU 基础版 预测时必须安装（目标跟踪、表格预测除外）：

```
python -m pip install paddlepaddle==2.2.2 -i https://mirror.baidu.com/pypi/simple
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU 基础版 预测时必须安装（目标跟踪、表格预测除外）：

```
python -m pip install paddlepaddle-gpu==2.2.2.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA10.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2 -i https://mirror.baidu.com/pypi/simple #CUDA10.2的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post110 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.0的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post111 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.1的PaddlePaddle
python -m pip install paddlepaddle-gpu==2.2.2.post112 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html #CUDA11.2的PaddlePaddle
```

不同cuda版本的环境，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持

安装 OpenVINO 使用x86\_64 CPU 加速版 SDK 预测时必须安装。

1) 请参考 [OpenVINO toolkit](#) 文档安装 2021.4版本, 安装时可忽略Configure the Model Optimizer及后续部分

2) 运行之前，务必设置环境变量

```
source /opt/intel/opencvino_2021/bin/setupvars.sh
```

安装 cuda、cudnn

- 使用Nvidia GPU 加速版 预测时必须安装。依赖的版本为 cuda9.0、cudnn7。版本号必须正确。

安装 pytorch (torch >= 1.7.0)

- 目标跟踪模型的预测必须安装pytorch版本1.7.0及以上（包含：Nvidia GPU 基础版、x86\_64 CPU 基础版）。
- 目标跟踪模型Nvidia GPU 基础版还需安装依赖cuda、cudnn。

关于不同版本的pytorch和CUDA版本的对应关系：[pytorch官网](#) 目标跟踪模型还有一些列举在requirements.txt里的依赖（包括torch >= 1.7.0），均可使用pip下载安装。

```
pip3 install -r requirements.txt
```

2. 安装 easyedge python wheel 包 安装说明

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。安装说明：[华为 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装

```
pip3 install -U EasyEdge_Devkit_Atlas300-{版本号}-cp36-cp36m-linux_x86_64.whl
```

安装说明：[飞腾 Atlas300](#) 除了需要安装BaiduAI\_EasyEdge\_SDK包，还需安装



```
pip3 install -U EasyEdge_Devkit_Phytium.Atlas-[版本号]-cp36-cp36m-linux_aarch64.whl
```

### 3. 使用序列号激活

#### 获取序列号

**将离线服务说明**  
发布和离线服务。将训练完成的模型部署在本地，离线调用模型。可以选择将模型部署在本地的服务器、小型设备、软硬一体方案专项适配硬件上。通过API、SDK进一步集成。灵活适应不同业务场景。

[发布新设备](#) [控制台](#)

**服务器** 通用小型设备 专项适配硬件

**SDK** **API**

此处发布、下载的SDK为未授权SDK，需要前往控制台[获取序列号](#)激活后才能正式使用。SDK内附有对应版本的Demo及开发文档，开发者可参考源代码完成开发。

模型名称	发布版本	应用平台	模型加速	发布状态	发布时间	
sun_小目标检测	134319-V1 <a href="#">查看任务报告</a>	通用X86 CPU-Linux	基础版	已发布	2021-08-19 20:24	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:24	<a href="#">下载加速版SDK</a>
		高伟达GPU-Linux	基础版	已发布	2021-08-19 20:35	<a href="#">下载SDK</a>
			精度无损压缩加速	已发布	2021-08-19 20:34	<a href="#">下载加速版SDK</a>
			基础版	已发布	2021-08-19 18:17	<a href="#">下载SDK</a>

#### 修改demo.py 填写序列号

```
pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
```

### 4. GPU 加速版 使用 GPU 加速版，在安装完 whl 之后，必须：

1. 从[这里](#)下载 TensorRT7.0.0.11 for cuda9.0，并把解压后的 lib 放到 C++ SDK 的 lib 目录或系统 lib 目录
2. 运行时，必须在系统库路径中包含 C++ SDK 下的 lib 目录。如设置 LD\_LIBRARY\_PATH

```
cd ${SDK_ROOT}
```

**\*\*1. 安装 python wheel 包\*\***

```
tar -xzf python/*.tar.gz
pip install -U {对应 Python 版本的 wheel 包}
```

**\*\*2. 设置 LD\_LIBRARY\_PATH\*\***

```
tar -xzf cpp/*.tar.gz
export EDGE_ROOT=$(readlink -f $(ls -h | grep "baidu_easyedge_linux_cpp"))
export LD_LIBRARY_PATH=$EDGE_ROOT/lib
```

**\*\*3. 运行 demo\*\***

```
python3 demo.py {RES文件夹路径} {测试图片路径}
```

如果是使用 C++ SDK 自带的编译安装的 OpenCV，LD\_LIBRARY\_PATH 还需要包括 C++ SDK 的 build 目录下的 thirdparty/lib 目录

如果没有正确设置 LD\_LIBRARY\_PATH，运行时可能报错：

```
ImportError: libeasyedge.so.0.4.3: cannot open shared object file: No such file or directory
ImportError: libopencv_core.so.3.4: cannot open shared object file: No such file or directory
```

### 5. 测试 Demo

#### 5.1 表格预测 输入对应模型文件夹（默认为RES）和测试数据地址（csv文件地址），运行：

```
python3 demo.py {model_dir} {/xxx/xxx.csv}
```

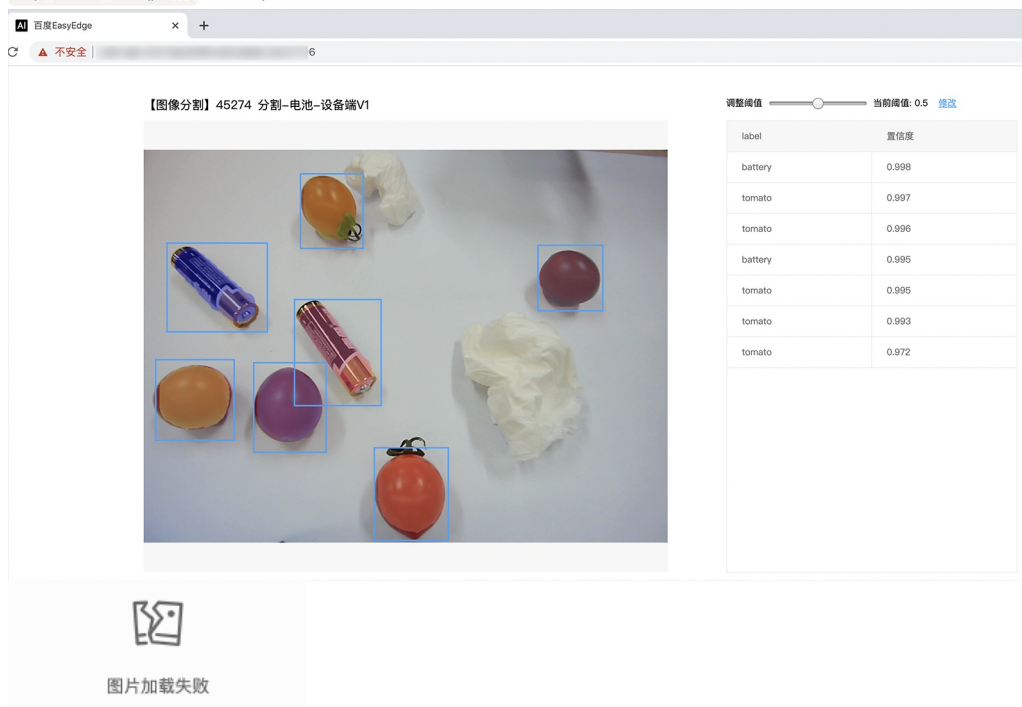
#### 6. 测试Demo HTTP 服务 输入对应的模型文件夹（默认为RES）、序列号、设备ip和指定端口号，运行：

```
python3 demo_serving.py {model_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
```

后，会显示：

Running on http://0.0.0.0:24401/

字样，此时，开发者可以打开浏览器，[http://\(设备ip\):24401](http://(设备ip):24401)，选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里`http\_client\_test()`函数请求http服务进行推理。



## 使用说明

### 使用流程 demo.py

```
import BaiduAI.EasyEdge as edge

pred = edge.Program()
pred.set_auth_license_key("这里填写序列号")
pred.init(model_dir=(RES文件夹路径), device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image((numpy.ndarray的图片))
pred.close()
```

### demo\_serving.py

```
import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir=(RES文件夹路径), license=serial_key)
**请参考同级目录下demo.py里:**
**pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)**
**对以下参数device\device_id和engine进行修改**
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
```

## 初始化

- 接口

```
def init(
    self,
    model_dir,
    device=Device.CPU,
    engine=Engine.PADDLE_FLUID,
    config_file="conf.json",
    preprocess_file="preprocess_args.json",
    model_file="model",
    params_file="params",
    label_file="label_list.txt",
    infer_cfg_file="infer_cfg.json",
    device_id=0,
    thread_num=1,
):
    """
    Args:
        model_dir: str
        device: BaiduAI.EasyEdge.Device , 比如 : Device.CPU
        engine: BaiduAI.EasyEdge.Engine , 比如 : Engine.PADDLE_FLUID
        config_file: str
        preprocess_file: str
        model_file: str
        params_file: str
        label_file: str 标签文件
        infer_cfg_file: 包含预处理、后处理信息的文件
            device_id: int 设备ID
            thread_num: int CPU的线程数
    Raises:
        RuntimeError, IOError
    Returns:
        bool: True if success
    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

## 预测图像

- 接口

```
def infer_image(self, img, threshold=0.3, channel_order="HWC", color_format="BGR", data_type="numpy"):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type
    Returns:
        list
    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测，矩形的左上角坐标 (相对长宽的比例值)
x2, y2	float	0~1	物体检测，矩形的右下角坐标 (相对长宽的比例值)
mask	string/numpy.ndarray	图像分割的mask	

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

- iii) 图像分割

```
{
  "name": "cat",
  "score": 1.0,
  "location": {
    "left": ...,
    "top": ...,
    "width": ...,
    "height": ...,
  },
  "mask": ...
}
```

mask字段中，data\_type为numpy时，返回图像掩码的二维数组

```
{
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
  {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
```

其中1代表为目标区域，0代表非目标区域

data\_type为string时，mask的游程编码，解析方式可参考 [demo](#)

### 预测视频（目前仅限目标跟踪模型调用）

- 接口

```
def infer_frame(self, frame, threshold=None):
    """
    视频推理(抽帧之后)
    :param frame:
    :param threshold:
    :return:
    """
```

- 返回格式dict

字段	类型	说明
pos	dict1	当前帧每一个类别的追踪目标的像素坐标(tlwh)
id	dict2	当前帧每一个类别的追踪目标的id
score	dict3	当前帧每一个类别的追踪目标的识别置信度
label	dict4	class_idx(int)与label(string)的对应关系
class_num	int	追踪类别数

### 预测声音

- 使用声音分类SDK需要安装额外依赖 `pip 安装 resampy pydub` 音频默认格式支持wav文件预测，如果需要预测mp3等其他音频格式的数据需要系统额外安装ffmpeg（windows系统的ffmpeg已集成在sdk中无需额外安装，linux系统需要手动安装）

- 接口

```
def infer_sound(self, sound_binary, threshold=0.3):
    """
    Args:
        sound_binary: sound_binary
        threshold: confidence
    Returns:
        list
    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类的置信度
label	string		分类的类别
index	number		分类的类别

### 表格预测

- 考虑到表格字段内容和长度的不固定性，我们建议您参考“校验服务”页面提供的详细信息。您可以访问该页面：<https://ai.baidu.com/easydl/app/validate/ml/models/verify>，并从中复制数据请求的 Body 部分作为参考模板。这将帮助您理解如何灵活处理各种不同的字段。

- 接口

```
def infer_csv(self, data):
    """
    结构化数据推理
    Args:
        data: pd.DataFrame or list or dict
    Returns:
    """
```

- 返回格式: list 接口直接反馈预测结果数组

**升级模型** 适用于经典版升级模型，执行 `bash update_model.sh`，根据提示，输入模型路径、激活码、模型ID、模型版本，等待模型更新完毕即可。

## FAQ

**Q: EasyDL 离线 SDK 与云服务效果不一致，如何处理？** A: 后续我们会消除这部分差异，如果开发者发现差异较大，可联系我们协助处理。

**Q: 运行时报错 "非法指令" 或 "illegal instruction"** A: 可能是 CPU 缺少 avx 指令集支持，请在 [paddle官网](#) 下载 noavx 版本覆盖安装

**Q: NVIDIA GPU预测时，报错显存不足：** A: 如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过 `export FLAGS_fraction_of_gpu_memory_to_use=0.3` 来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 allocate 更多的显存。

**Q: 我想使用多线程预测，怎么做？** 如果需要多线程预测，可以每个线程启动一个Progam实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的thread\_num参数。

**Q: 运行SDK报错 Authorization failed**

**情况一：日志显示 Http perform failed: null respond** 在新的硬件上首次运行，必须联网激活。

SDK 能够接受 HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

**情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx)** 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/baidu/easyedge` 目录，再重新激活。

**情况三：Atlas Python SDK日志提示ImportError: libavformat.so.58: cannot open shared object file: No such file or directory 或者其他类似so找不到** 可以在LD\_LIBRARY\_PATH环境变量加上libs和thirdpartylibs路径，例如

```
export LD_LIBRARY_PATH=/xxx/libs:/xxx/thirdpartylibs:$LD_LIBRARY_PATH # tips: 这里/xxx需要替换为真实路径，/xxx路径查找方法如下
```

查找安装包内libs和thirdpartylibs路径的方法如下(以华为Atlas300 SDK为例，其他SDK查找方法类似)：

```
pip3 show EasyEdge-Devkit-Atlas300 # 结果中会显示 Location 路径，也就是包的安装路径
**libs和thirdpartylibs两个路径在 Location 所指示的路径 easyedge_CANN 子文件夹下**
```

## 通用小型设备部署

## Windows集成文档

### 简介

本文档介绍表格预测通用小型设备Windows SDK的使用方法。

- 硬件支持：
  - Intel CPU 普通版 \* x86\_64
  - CPU 加速版 - Intel Xeon with AVX2 and AVX512 - *Intel Core Processors with AVX2* - Intel Atom Processors with SSE \* - AMD Core Processors with AVX2
- 操作系统支持
  - 普通版：64位 Windows 7 及以上，64位Windows Server2012及以上
  - 加速版：64位 Windows 10，64位Windows Server 2019及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015-2019
- 协议
  - HTTP
- 更详细的环境说明可参考SDK内的README.md

\*intel 官方合作，拥有更好的适配与性能表现

**Release Notes** | 时间 | 版本 | 说明 | | ----- | ----- | ----- | | 2023-06-29 | 1.8.2 | 优化模型算法 | | 2023-05-17 | 1.8.1 | 预测引擎升级，修复部分网络内存泄露问题 | | 2023-03-16 | 1.8.0 | 预测引擎升级 | | 2022-12-29 | 1.7.2 | 预测引擎升级 | | 2022-10-27 | 1.7.1 | GPU底层引擎升级，下线基础版CUDA10.0及以下版本支持 | | 2022-09-15 | 1.7.0 | 新增支持表格预测 | | 2022-07-28 | 1.6.0 | 优化模型算法 | | 2022-05-27 | 1.5.1 | 新增支持BML Cloud小目标检测模型 | | 2022-05-18 | 1.5.0 | 修复各别机器下程序崩溃的问题 | | 2022-04-25 | 1.4.1 | EasyDL, BML升级支持paddle2模型 | | 2022-03-25 | 1.4.0 | 优化模型算法 | | 2021-12-22 | 1.3.5 | CPU基础版推理引擎优化升级；demo程序优化环境依赖检测 | | 2021-10-20 | 1.3.4 | CPU加速版推理引擎优化升级 | | 2021-08-19 | 1.3.2 | 新增DEMO二进制文件 | | 2021-06-29 | 1.3.1 | 预测引擎升级 | | 2021-05-13 | 1.3.0 | 模型发布新增多种加速方案选择；目标追踪支持x86平台的GPU及加速版；展示已发布模型性能评估报告 | | 2021-04-08 | 1.2.3 | 支持BML平台模型仓库本地上传模型 | | 2021-03-09 | 1.2.2 | CPU加速版支持int8量化模型 | | 2021-01-27 | 1.2.1 | 新增模型支持；性能优化；问题修复 | | 2020.12.18 | 1.2.0 | 推理引擎升级 | | 2020-11-26 | 1.1.20 | 新增一些模型的加速版支持 | | 2020.10.29 | 1.1.20 | 修复已知问题 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020-09-17 | 1.1.19 | 支持更多模型 | | 2020.08.11 | 1.1.18 | 支持专业版更多模型 | | 2020.06.23 | 1.1.17 | 支持专业版更多模型 | | 2020.05.15 | 1.1.16 | 优化性能，修复已知问题 | | 2020.04.16 | 1.1.15 | 升级引擎版本 | | 2020.03.13 | 1.1.14 | 支持EdgeBoardVMX | | 2020.02.23 | 1.1.13 | 支持多阶段模型 | | 2020.01.16 | 1.1.12 | 预测默认使用推荐阈值 | | 2019.12.26 | 1.1.11 | CPU加速版支持物体检测高精度 | | 2019.12.04 | 1.1.10 | 支持图像分割 | | 2019.10.21 | 1.1.9 | 支持 EasyDL 专业版 | | 2019.08.29 | 1.1.8 | CPU 加速版支持 | | 2019.07.19 | 1.1.7 | 提供模型更新工具 | | 2019.05.16 | 1.1.3 | NVIDIA GPU 支持 | | 2019.03.15 | 1.1.0 | 架构与功能完善 | | 2019.02.28 | 1.0.6 | 引擎功能完善 | | 2019.02.13 | 1.0.5 | paddlepaddle 支持 | | 2018.11.30 | 1.0.0 | 第一版！ |

## 快速开始

### 1. 安装依赖

必须安装：

安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

### Visual C++ Redistributable Packages for Visual Studio 2015-2019

<https://docs.microsoft.com/en-us/cpp/windows/latest-supported-vc-redist?view=msvc-160>

### 表格预测依赖

brotlipy==0.7.0 certifi==2020.6.20 joblib==1.0.1 kaggle==1.5.12 Pillow py4j pycosat python-dateutil python-slugify ruamel\_yaml text-unidecode threadpoolctl flask pandas==1.0.5 scikit-learn==0.23.2 lightgbm==2.2.3 catboost==0.24.1 xgboost==1.2.0 numpy==1.19.5 scipy==1.5.2 psutil==5.7.2 pymml==0.9.7 torch==1.8.0 jieba==0.42.1 pyod==0.8.5 pyarrow==6.0.0 scikit-optimize==0.9.0 pyspark==3.3.0

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe，输入Serial Num



点击“启动服务”，等待数秒即可启动成功，本地服务

默认运行在

<http://127.0.0.1:24401/>

其他任何语言只需通过HTTP调用即可。



### 使用说明

### 调用说明

Python 使用示例代码如下

### json结构说明

考虑到表格字段内容和长度的不固定性，我们建议您参考SDK内置demo页面展示的json示例，这将帮助您理解如何灵活处理各种不同的字段。



```
import requests

with open('./1.json', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./1.json", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

**include <sys/stat.h>**
**include <curl/curl.h>**
**include <iostream>**
**include <string>**
**define S_ISREG(m) (((m) & 0170000) == (0100000))**
**define S_ISDIR(m) (((m) & 0170000) == (0040000))**

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./1.json";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

结果 获取的结果存储在response字符串中。 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数

预测结果数组

集成指南

基于HTTP集成

通过EasyEdge.exe启动服务后，参照上面的调用说明，通过HTTP请求集成到自己的服务中

## 基于c++ dll集成

### 集成前提

解压开的SDK包中包含src、lib、dll、include四个目录才支持基于c++ dll集成

### 集成方法

参考src目录中的CMakeLists.txt进行集成

## 基于c# dll集成

### 集成前提

解压开的SDK包中包含src\demo\_serving\_csharp、dll两个目录才支持基于c# dll集成

### 集成方法

参考src\demo\_serving\_csharp目录中的CMakeLists.txt进行集成

## FAQ

### 1. 服务启动失败，怎么处理？

根据SDK内的README.md检查依赖是否都已正确安装

请确保相关依赖都安装正确，版本必须如下：  
.NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

如使用的是CPU加速版，需额外确保Opencv安装正确，版本为2020.3.1LTS版 如使用Windows Server，需确保开启桌面体验

2. 服务调用时返回为空，怎么处理？ 调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？ SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，缺失DLL？ 打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

6. 启动失败，报错NotDecrypted？ Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

7. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

8. 勾选“开机自动启动”后，程序闪退

一般是写注册表失败。

可以确认下HKEY\_CURRENT\_USER下Software\Microsoft\Windows\CurrentVersion\Run能否写入（如果不能写入，可能被杀毒软件等工具管制）。也可以尝试基于bin目录下的easyedge\_serving.exe命令行形式的二进制，自行配置开机自启动。

其他问题 如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## 故障处理

## 训练相关问题

### 数据处理失败或者状态异常怎么办？

- 如果是首次导入数据失败，请检查数据文件编码格式是否为UTF-8或GBK
- 如果是非首次导入数据失败，请检查新导入文件的首行是否与首次导入时一致，即都为列名或都为数据
- 如果自查没有发现问题，请在百度云控制台内[提交工单](#)反馈

### 模型训练失败怎么办？

- 如果遇到模型训练失败的情况，请在百度云控制台内[提交工单](#)反馈

## 模型效果相关问题

### 实际调用服务时模型效果变差？

- 在实际业务场景中，数据的分布可能会发生变化，即使用历史数据训练的模型不能正确对新的数据进行预测，此时需要收集新的数据并重新进行模型训练
- 如果训练模型使用的数据量太小，导致训练数据不能正确反映全部的数据特征，也会导致上线模型的效果变差。

\*\*如果线上请求数据与训练数据未发生显著的分布变化，请在百度云控制台内[提交工单](#)反馈

## 模型上线相关问题

### 希望加急上线怎么处理？

- 请在百度云控制台内[提交工单](#)反馈

### 每个账号可以上线几个模型？是否可以删除已上线的模型？

- 每个账号最多申请发布十个模型，已上线模型无法删除

### 申请发布模型审核不通过都是什么原因？

- 可能原因有，1、经过电话沟通当前模型存在一些问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝。2、电话未接通且模型效果较差，会直接拒绝。如果需要申诉，请在百度云控制台内[提交工单](#)反馈

## 🔗 训练任务失败错误排查

表格预测一般是由于任务类型选择错误引起的。当训练任务失败时，请您检查选择的任务类型与目标列是否匹配，以下为各个任务类型对应的目标：

算法类型	目标列
二分类	目标列是离散值，且只有两种可能的取值。如在精准营销场景中预测一个用户是否为潜在购买用户，其目标列仅存在“True”和“False”两种取值，解决该问题的模型属于二分类模型。
多分类	目标列是离散值，并具有有限的可能取值。如在用户分类场景中，根据用户的历史消费数据，将用户划分到不同消费偏好的类别中，解决该问题的模型属于多分类模型。 <b>不建议将重复率很小的值或时间列作为目标列。</b>
回归	目标列是连续的实数范围，或者属于某一段连续的实数区间。如在销量预测场景中，销量值可能是某个取值范围内的任意值，解决该问题的模型属于回归模型。 <b>目标列不能包含大量无法转成数值的异常值</b>

若您还是无法判断算法类型，请选择默认的自动。

## 时序预测

### 时序预测介绍

#### 简介

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

定制时序预测模型，旨在帮助用户通过机器学习技术从历史数据中发现潜在规律，从而对未来的变化趋势进行预测。本文介绍**时序预测模型**：相较于表格数据预测使用的分类或回归模型，时序预测模型使用的训练数据中必须包含有效时序的特征，一般时序具有固定的频率，且在连续时间范围内的每个时间点上都有一个值。

以下是关于时序预测模型的技术文档。

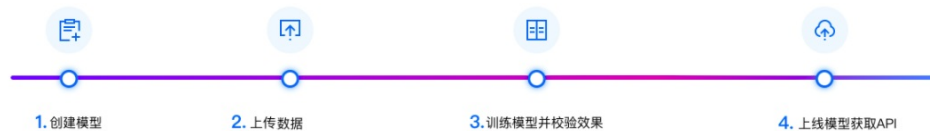
## 应用场景

- 销量预测：基于历史销量数据预测当期的销售量，进而帮助厂商制定更合理的生产或备货计划，从而提高利润
- 交通流量预测：基于给定路段的历史交通量数据推测未来的交通量，为交通运输规划与研究提供决策依据
- 价格预测：从历史数据中发现商品的变化规律以及影响价格的因素，从而为未来的商业行为提供支持

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作。在数据已经准备好的情况下，最快几分钟即可获得定制模型。

下面将详细介绍每一步的操作方式和注意事项。如果文档没有解决您的问题，请在百度云控制台内[提交工单](#)反馈。



## 数据准备

### 时序数据集介绍

#### 时序数据介绍

时序数据包含有序特征，常规时序数据是具有一定频率的并且在连续时间范围内的每个采样点上都有一个值。

一个时序数据集可以包含一个或多个时间序列，如下数据集包含一个品牌在A、B两个地区的每日销售数据：

```

datetime,area,sales_quantity
9/3/2018,A,2000
9/3/2018,B,600
9/4/2018,A,2300
9/4/2018,B,550
9/5/2018,A,2100
9/5/2018,B,650
9/6/2018,A,2400
9/6/2018,B,700
9/7/2018,A,2450
9/7/2018,B,650
  
```

上述数据内容可以分为A地区销量时序：

```

datetime,area,sales_quantity
9/3/2018,A,2000
9/4/2018,A,2300
9/5/2018,A,2100
9/6/2018,A,2400
9/7/2018,A,2450
  
```

B地区销量时序：

```

datetime,area,sales_quantity
9/3/2018,B,600
9/4/2018,B,550
9/5/2018,B,650
9/6/2018,B,700
9/7/2018,B,650
  
```

一个时序数据集除具有时间特征以及一个标量特征外，还可以具有其它影响标量取值的特征，如在销量数据场景下，当天的气温、是否节假日等因素也会影响销售数据：

```
datetime,is_holiday,sales_quantity
9/3/2018,Y,600
9/4/2018,N,550
9/5/2018,N,650
9/6/2018,Y,700
9/7/2018,N,650
```

### 数据要求

- 目前仅支持CSV格式的数据文件
- 一次仅能上传一个文件，可以是一个CSV文件或由多个CSV文件压缩成的zip包
- 单个上传文件大小不能超过5GB
- 一个数据集包含的总文件大小不能超过20GB

### 数据文件内容要求

- 当数据文件包含列名时，列名称可以包含字母、数字和下划线（\_），但不能以下划线开头。
- 文件内容以换行符（即字符“\n”，或称为LF）分隔各行，行内容以英文逗号（即字符“,”）分隔各列
- 必须包含要预测的值即目标列，且目标列的数据类型会决定模型的类型。
- 文件中文本列取值长度不能超过4096个字符。
- 文件必须至少包含两列，并至少包含一个日期列，总列数不得超过1000列。
- 数据集的总行数不能超过1000万行。
- zip包中的多个CSV文件必须使用相同的编码格式，都包含列名或都不包含列名；且列的顺序必须保持一致
- 在扩充数据集时，新导入数据文件的首行与数据集的列名相同时，将被视为列名，否则将被视作数据。

## 创建时序数据集

在EasyDL经典版中，您可以在“数据总览”页面，完成数据集创建、数据导入等操作，为模型构建准备好数据。

### 创建数据集

数据集需要先定义，然后再导入数据。

1. 单击“数据总览”，进入数据集列表页面。
2. 单击“创建数据集”，进入数据集创建页面。
3. 输入数据集名称，单击“完成”结束创建。

完成创建后，可以在数据集列表中查看新建的数据集。系统默认生成V1版本，当前数据内容为空，可以通过导入的方式向其中添加数据。

### 导入数据

通过导入的方式可以向数据集中添加或追加数据。

1. 单击“数据总览”，进入数据集列表页面。
2. 单击待导入数据集所在行的“导入”按钮，进入数据集导入页面。
3. 导入数据文件。

导入的数据文件可以是CSV文件或由CSV文件组成的压缩包文件。

如果导入的是CSV文件，支持数据预览，如果是压缩包格式，则不支持预览。

4. 根据数据文件的实际情况进行列名设置。
  - 设置首行为列名：将导入的数据文件中的首行作为列名。
  - 设置首行非列名：此时系统会自动生成列名，而将首行作为数据。
5. 单击“确认并返回”完成导入操作。

## 模型训练

### 时序预测模型介绍

#### 时序预测模型介绍

时序预测模型是基于包含时间特征的结构化数据进行建模，系统会基于用户上传的数据使用预置算法进行模型构建与训练。当完成模型训练后，系统不仅提供了常见的评估指标而且会生成可视化的预测序列效果图，帮助用户检查模型的好坏。对于达到业务要求的时序预测模型，可以部署为在线服务，通过远程调用的方式对新的时间数据进行预测。

#### 应用场景

- 销量预测：基于历史销量数据预测当期的销售量，进而帮助厂商制定更合理的生产或备货计划，从而提高利润
- 交通流量预测：基于给定路段的历史交通量数据推测未来的交通量，为交通运输规划与研究提供决策依据
- 价格预测：从历史数据中发现商品的变化规律以及影响价格的因素，从而为未来的商业行为提供支持

### 创建模型

在EasyDL中，您可以在“模型中心”进行模型的创建。在EasyDL中模型可以包括多个版本，每次训练会生成一个版本。各个版本的模型之间相互独立，可以分别进行版本发布等操作。

#### 创建模型

模型需要先创建，然后才能进行训练。

1. 单击“创建模型”，进入模型创建页面。
2. 填写模型创建信息，如下图所示。

时序预测模型 [提交工单](#) 收起 ^

定制基于时序数据的模型，根据历史数据对未来的变化趋势进行预测，该类模型可应用在销量预测、交通流量预测等场景。此类模型训练速度较快，具有TODO个特征列的TODO条样本的表格数据，一般可在TODO分钟内训练完毕。

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

模型列表 > 创建模型

模型类别 时序预测

模型名称 \*

模型归属 公司 个人

请输入公司名称

所属行业 \* 请选择行业

应用场景 \* 请选择应用场景

邮箱地址 \* z\*\*\*\*\*@baidu.com

联系方式 \* 135\*\*\*\*919

功能描述 \*

0/500

下一步

- 模型名称：指定模型的名称
- 模型归属：公司或个人，并输入相关名称
- 所属行业：请根据实际情况进行选择
- 应用场景：请根据实际情况进行选择
- 邮箱地址：请根据实际情况进行设置
- 联系方式：请根据实际情况进行设置

- 功能描述：用于记录模型创建的背景、用途等方面的信息。

### 3. 单击“下一步”完成模型创建。

完成创建后，可以在模型列表中查看新建的模型。新建的模型不包含任何版本的模型，在训练后会生成新的模型版本。

## 🔗 训练模型

EasyDL提供的的时序预测算法不仅性能高、可扩展，还针对速度、规模和准确性进行了优化，可以在大规模数据集上进行训练。

### 训练模型

在准备好数据集并创建模型后，可以创建训练任务。

1. 单击“训练模型”，进入模型训练页面。
2. 填写模型训练信息，如下图所示。

如果选择的数据集中仅包含时间和数值两列，则训练参数配置如下所示：

训练模型

---

选择模型  ▼

选择数据集  ▼

选择时间列  ▼

选择时间间隔   ▼

选择目标列  ▼ [预览序列](#)

滑动窗口大小  ▼

预测长度  ▼

部署方式  公有云API

[开始训练](#)

- 选择模型：选择要训练的模型
- 选择数据集：选择训练模型使用的数据集
- 选择目标列：从数据集中选择表示时序的日期时间列
- 选择时间间隔：请根据序列中相邻两个样本点的时间间隔进行设置
- 选择目标列：需要被预测的随时间而变化的列
- 滑动窗口大小：表示使用多年的历史数据生成预测数据
- 预测长度：要预测的序列的长度，该长度因小于滑动窗口大小

如果选择的数据集中除时间和数值外还有其它列，则训练参数配置如下所示：



## 训练模型

选择模型	销量预测	▼
选择数据集	dominicks_OJ V1	▼
选择时间列 ②	WeekStarting	▼
选择分组字段 ②	Store x Brand x	
选择时间间隔 ②	1	日 ▼
选择目标列 ②	Quantity	▼ <a href="#">预览序列</a>
滑动窗口大小 ②	30	▲ ▼
预测长度 ②	10	▲ ▼
部署方式	<input checked="" type="checkbox"/> 公有云API	

[开始训练](#)

相比于仅包含一个序列的训练任务，其需要配置分组字段：

- 分组字段用于将数据集划分为具有相同时间戳的多个时序
- 最迟支持设置两个分组字段
- 分组字段只能为类别列

### 3. 单击“开始训练”启动训练任务。

启动训练任务后，系统会在模型下的列表中创建一个新的模型版本，新建的版本处于“训练中”的状态，当处于“训练完成”状态时表示模型已完成训练。

### 4. 平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有时序预测操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

### 🔗 查看模型

训练任务结束后，可以查看模型的各项指标，以确定模型是否满足要求。不同类型的模型包含了不同的指标，用户可以根据实际的业务场景查看关键指标是否满足要求。

#### 查看模型

对于“训练完成”的模型，可以查看其评估结果。

1. 单击“我的模型”，进入模型列表页面。
2. 单击待查看模型的“历史版本”，进入模型版本列表页面。
3. 点击待查看模型版本所在行的“完整评估结果”，系统展示评估结果页面。

预测结果包括整体评估以及详细评估：

- 整体评估：提供了包括MAE、MSE、MAPE、R2 Score四项数值指标
- 详细评估：则给出了预测结果的可视化表示，其中预测序列包含了预测部分的对比以及差值情况；完整序列则不仅仅包含了生成的预测序列部分，也包含了用于生成预测序列的已知序列部分。

我的模型 > mymodel01模型评估报告

部署方式 公有云API (目前仅支持) 版本 V1

训练时长 8分钟

整体评估

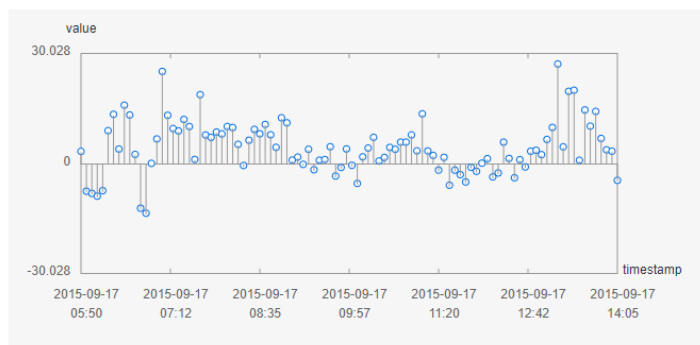
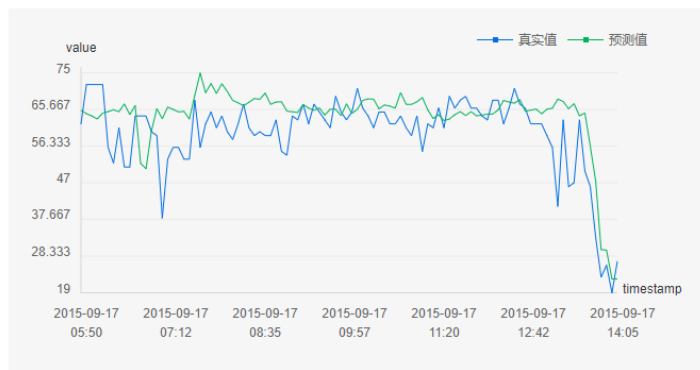
特征 1列 目标列 value 算法类型 时序预测

mymodel01 V1整体效果欠佳。 如何优化效果?

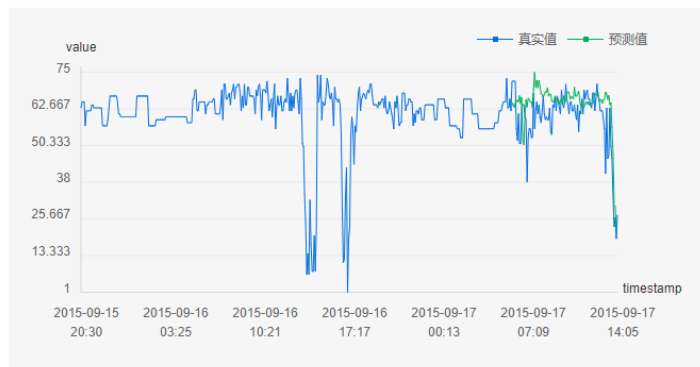
名称	数值
MAE (Mean Absolute Error) 平均绝对误差	6.530
MSE (Mean Squared Error) 均方误差	72.380
MAPE (Mean Absolute Percentage Error) 平均绝对百分比误差	12.36%
R2 Score (决定系数) 回归得分函数	0.476

详细评估

预测序列



完整序列



校验模型

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。

校验模型

在训练任务成功完成后，即可使用实际数据进行校验。

1. 单击“校验模型”，进入模型校验页面。
2. 选择要校验的模型及其版本。
3. 单击“启动模型校验服务”。
  - 对于单序列模型，系统会自动生成校验数据
  - 对于多序列模型，可以通过上传CSV文件来填充测试数据，但每次测试时只能包含一个序列的数据
4. 单击“预测”，可以在右侧结果面板中查看预测结果。

校验示例如下所示：

模型列表

选择模型: mymodel01 | 部署方式: 公有云API (目前仅支持) | 选择版本: V1

当前模型MAE(平均绝对误差) 6.530 [评估报告](#) | 识别结果 [如何优化效果?](#)

预测数据		
	timestamp	value
	日期	数值
1	2015-09-08 11:39:00	73
2	2015-09-08 11:44:00	62
3	2015-09-08 11:59:00	66
4	2015-09-08 12:19:00	69
5	2015-09-08 12:24:00	65
6	2015-09-08 12:27:00	76

```

1  {
2    "tim...: [
3      "2015-09-09 15:33:00",
4      "2015-09-09 15:38:00",
5      "2015-09-09 15:43:00",
6      "2015-09-09 15:48:00",
7      "2015-09-09 15:53:00",
8      "2015-09-09 15:58:00",
9      "2015-09-09 16:03:00",
10     "2015-09-09 16:08:00",
11     "2015-09-09 16:13:00",
12     "2015-09-09 16:18:00",
13   ]
14   "val...: [
15     "65.057045099412"

```

预测 | 申请上线

## 模型发布

### ☞ 模型发布整体说明

训练完成后，可将模型部署在公有云服务器上，通过API进行调用。

### 公有云API

- 训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统或硬件设备整合
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

### 相关费用

将模型发布为API后，将获得1000次免费调用次数，如需更多调用量，请在百度云控制台内[提交工单](#)反馈。

### ☞ 公有云部署

### ☞ 如何发布时序预测API

训练完毕后可以在左侧导航栏中找到【发布模型】，依次进行以下操作即可发布公有云API：

- 选择模型
- 选择部署方式「公有云部署」
- 选择版本
- 自定义服务名称、接口地址后缀
- 申请发布

申请发布后，通常的审核周期为T+1，即当天申请第二天可以审核完成。如果需要加急、或者遇到莫名被拒的情况，请在百度云控制台内[提交工](#)

单反馈。

发布模型界面示意：

发布模型

选择模型: mymodel01

部署方式: 公有云部署

选择版本: V1

服务名称:

接口地址: https://aip.baidubce.com/rpc/2.0/ai\_custom/v1/time\_series/

其他要求: 若接口无法满足您的需求, 请描述希望解决的问题, 500汉字以内

标准接口规范参考

标准接口请求参考说明:

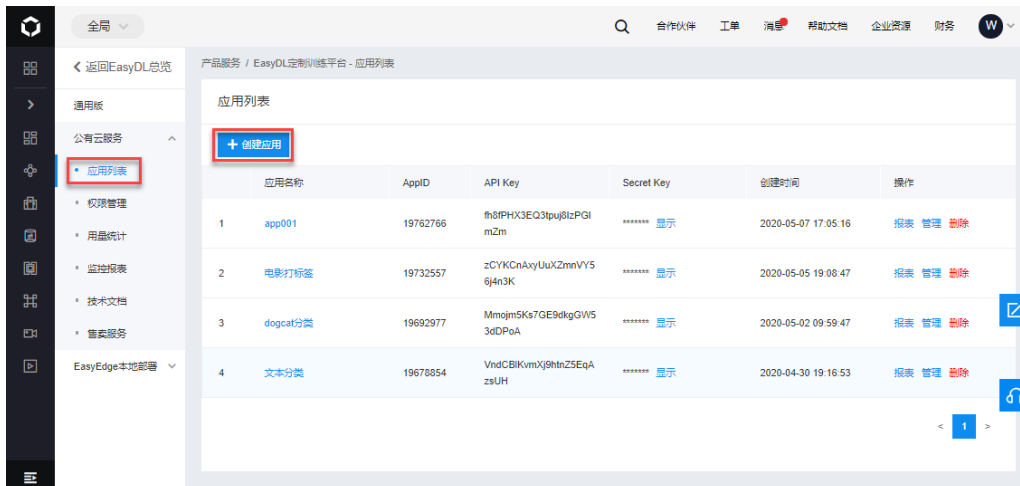
字段名称	必须	类型	说明
data	是	object	待预测数据, 待预测数据是由各个特征序列组成的对象。若训练时设置了分组字段, 一次API请求中的数据必须同一个分组的序列。

标准接口响应字段说明:

字段名称	必须	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回
error_msg	否	string	错误描述信息, 当请求错误时返回
result	否	object	预测结果

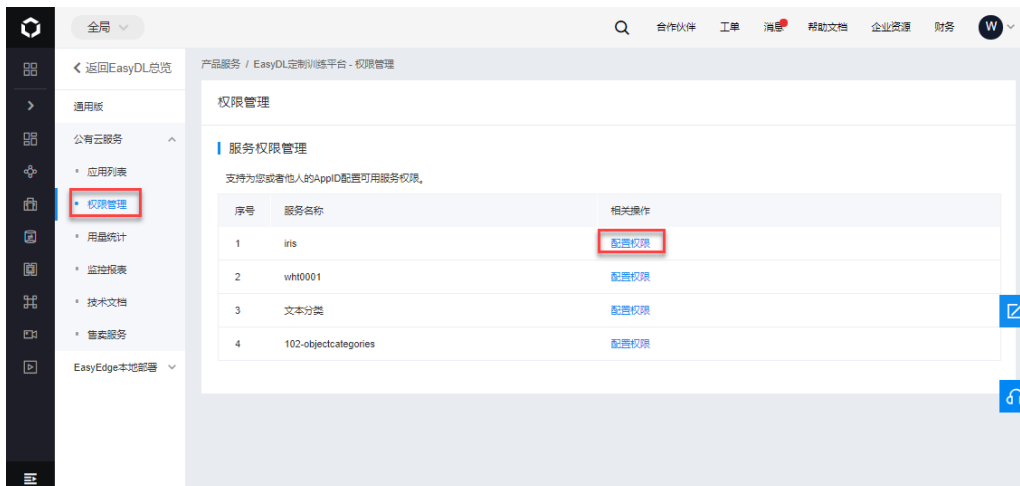
接口赋权

在正式使用之前, 还需要做的一项工作为接口赋权, 需要登录EasyDL控制台中创建一个应用, 获得由一串数字组成的appid, 然后就可以参考接口文档正式使用了



同时支持在「公有云服务管理」-「权限管理」中为第三方用户配置权限

示意图如下：





API调用文档

本文档主要说明定制化模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

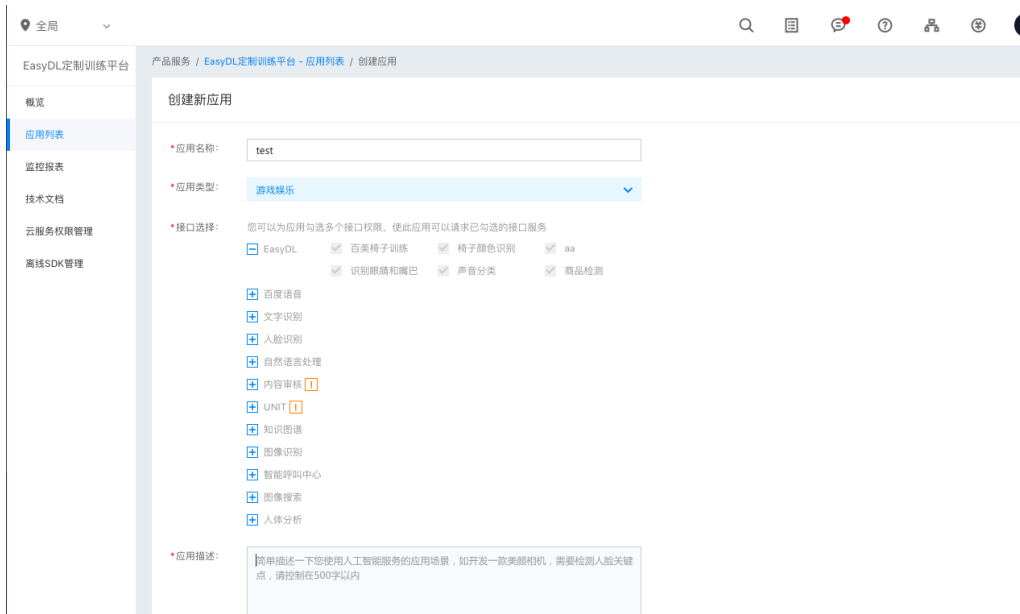
- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#) ,与其他开发者进行互动

接口描述

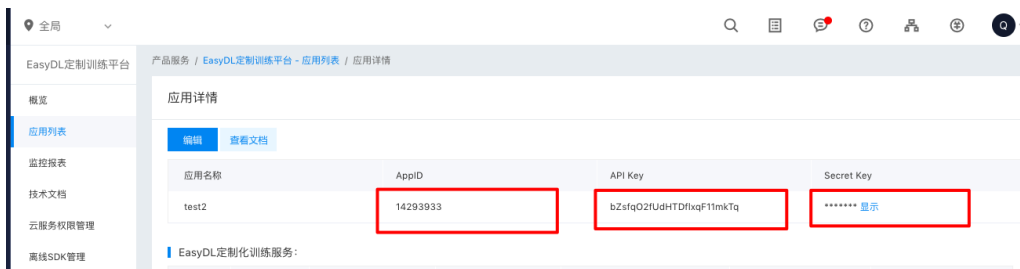
基于自定义训练出的表格数据预测模型，实现表格数据预测。模型训练完毕后发布可获得定制化表格数据预测API

接口鉴权

1、在EasyDL控制台创建应用



2、应用详情页获取AK SK



请求说明

请求示例

HTTP 方法：POST

请求URL：请首先进行自定义模型训练，完成训练后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求接口：

```
{
  "include_req": false,
  "data": <待预测数组>
}
```

具体示例如下：

```
{
  "data": {
    "datetime":
      [ "2015-09-09 15:33:00", "2015-09-09 15:38:00", "2015-09-09 15:43:00"],
    "sales_quantity":
      [ "10", "15", "20" ]
  }
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
data	是	object	-	待预测数据，待预测数据是由各个特征序列组成的对象。若训练时设置了分组字段，一次API请求中的数据必须同一个分组的序列。

返回说明

返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码，当请求错误时返回
error_msg	否	string	错误描述信息，当请求错误时返回
results	否	object	预测结果数组

在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#)，用于帮助开发者在线调试接口，查看在线调用的请求内容和返回结果、复制和下载示例代码等功能，简单易用。

错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请在百度云控制台内 <a href="#">提交工单</a> 反馈
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请在百度云控制台内 <a href="#">提交工单</a> 反馈
336001	Invalid Argument	入参格式有误，比如缺少必要参数、data格式错误等等，可检查下请求数据格式是否有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请在百度云控制台内 <a href="#">提交工单</a> 反馈

## 故障处理

### 🔗 训练任务失败错误排查

时序预测任务失败，一般主要是由于时间列格式错误或配置错误导致的，您可以参照如下内容检测您的数据或配置是否正确：

#### 时间列要求

在创建时序预测任务时，所使用的数据集的时间列应为Date类型并以yyyy-MM-dd开头，否则会造成训练任务失败。如 "yyyy-MM-dd'T'HH:mm:ssX" "yyyy-MM-dd HH:mm:ss" "yyyy-MM-dd" 等等

#### 时间间隔

选择时间列与目标列后，系统会根据数据的前100行去推断时间间隔，但是难免会推断错误。

预处理阶段会根据选择的时间间隔对数据进行重新采样，若选择了过大的时间间隔，降采样会导致数据过短。反之，过小的时间间隔会导致数据



分布异常，合理的选择时间间隔是保证预测质量的重要环节之一。

### 时间序列长度

时序预测通常会采用历史的一段时间的数据作为特征，因此**数据长度必须满足一定要求才能保证有足够的训练与评估样本**。

当时序数据为分组数据时，每个分组会作为一个单独的时间序列，因此需要保证**每个分组的时间序列足够长**。

另外选取了不适当的时间间隔，降采样后也会导致数据量不够的情况。

### 分组字段

为了确保每个时间列有充分的数据，限制了平均**每个分组至少包含20条数据**，若数据中存在大量超短时序数据，请您提前做好筛选。

### 窗口长度与预测长度

时序模型的输入为窗口长度中的各个特征，输出为预测长度的目标列。因此当数据中的时间序列较短的时候，请您**合理地降低窗口长度和预测长度**，有助于模型有充足的样本进行预测。

## EasyDL OCR使用说明

### EasyDL OCR介绍

#### 🔗 功能介绍

EasyDL文字识别，可定制识别图片中的文字信息，结构化输出关键字段内容，极大提升OCR模型训练效率，满足个性化卡证票据识别需求

#### • 数据标注

创建数据集并上传真实图片，定义数据识别字段作为标注标签，在图片中框选对应的 Key/Value 内容区域，自动识别框选区域内容完成转写，标注人员对识别结果进行查验纠正即可完成标注

#### • 数据生成

基于已标注数据，将图中已框选 Value 区内容进行抹除，选择对应的字体、字号、颜色，并根据该字段的内容选择相匹配的语料库，即可完成虚拟数据生成底板的创建，并基于此底板生成任意张版式相同内容不同的虚拟数据，快速扩充数据集规模，结合真实数据一同用作模型训练集

#### • 模型训练与管理

支持根据使用场景需求创建多个的识别模型，选择包含已标注数据及虚拟数据的数据集进行训练，即可自动排队完成训练，同时输出预测准确率供参考；也可扩充数据集对现有模型进行迭代训练，产出新版本

#### • 服务部署

对训练完成的模型可上传真实数据进行模型校验，效果满意后即可一键发布上线，自动分配机器资源完成部署，并生成标准API接口供业务调用

#### 🔗 特色优势

#### 🔗 零门槛操作

提供一站式流程化训练，并预置最佳预训练模型及训练参数，无需算法基础、无需关注算法细节即可完成模型训练

#### 🔗 高精度效果

基于百度丰富的商用模型实训经验，预置最佳实践产出的预训练模型，并基于百度自研的 EnDet 实体检测模型进行训练，模型平均准确率可达 90% 以上

#### 🔗 低成本数据

提供可视化数据管理平台，对上传图片进行智能预标注，仅需核对修改即可完成标注，并可基于一张标注图片批量生成虚拟数据，快速扩充训练集，启动模型训练

#### 🔗 超灵活部署

支持多种部署方式，公有云服务可一键部署，即刻生成 Restful API，毫秒级调用响应，高并发承载；同时，完整平台支持私有化部署，可用于搭建企业内部 AI 中台；也可支持产出模型容器化打包进行本地部署，快速完成项目交付

#### 🔗 应用场景

- **证照电子化审批**：对政府、金融、企业等审批流程中涉及到的各种证照，如食品/药品经营许可证、特种设备审批证等，进行定制训练，快速

提取关键信息完成线上审批，实现 7\*24 小时无间断服务

- **财税报销电子化**：对不同金融或税务机构发型的各类财务发票、银行单据进行定制训练，快速实现财税凭证的录入，大幅度节约凭证邮寄、录入成本，实现线上电子化报税报销
- **保险智能理赔**：对不同版式的保单或不同地区、不同医疗系统开具的医疗票据进行定制训练，实现保险理赔相关材料的快速录入，降低人力成本，提升保险理赔的业务安全性及快捷性

## API文档

### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/ai\\_custom/v1/ocr](https://aip.baidubce.com/rest/2.0/ai_custom/v1/ocr)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">“Access Token获取”</a>

Header如下：

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

#### 请求参数

参数	是否必选	类型	可选值范围	说明
image	和 url 二选一	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式
url	和 image 二选一	string	-	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/jpeg/png/bmp格式，当image字段存在时url字段失效，不支持https的图片链接。当image字段存在时 url 字段失效
modelld	是	string	-	模型 ID，自训练产出模型的唯一标示，可用于调用指定的已发布模型进行结构化识别，可在 <a href="#">「我的模型」</a> 页查看并复制使用
detect_direction	是	string	true/false	是否开启图像方向矫正功能，可选值有： - <b>true</b> ：开启图像方向矫正功能，可自动矫正不同旋转角度的图片进行识别； - <b>false</b> ：关闭图像矫正功能，如要识别的内容均为正向图片，建议可关闭此功能避免误矫正。

### 返回说明

#### 返回参数

字段	类型	说明
log_id	int	调用请求的唯一日志id，如需技术支持进行问题排查请反馈此id以快速进行问题定位
error_code	int	0代表成功，如果有错误码返回可以参考错误码列表排查问题
error_msg	string	如果error_code具体的失败信息，可以参考下方错误码列表排查问题
result	Object	识别返回的结果，每一个key代表识别字段名称，对应的value为该字段的识别结果
+ "key"	Array	识别内容，“key”为数据标注时创建的字段名称，将不同版式的内容进行归一化输出
++ probability	Object	字段的置信度，包括平均和最小置信度
+++ average	int	字的平均置信度
+++ min	int	字的最小置信度
++ location	Object	字段在原图上对应的矩形框位置
+++ top	int	字段文本框左上角点的上边距
+++ left	int	字段文本框左上角点的左边距
+++ width	int	字段文本框的宽度
+++ height	int	字段文本框的高度
++ word	string	字段识别结果

### 返回示例

```
{
  "log_id": "161648117706127",
  "result": [
    "收款卡号": [
      {
        "probability": {
          "average": 0.9972677230834961,
          "min": 0.6964160077398716
        },
        "location": {
          "top": 404,
          "left": 451,
          "width": 303,
          "height": 61
        },
        "word": "119086830765501"
      }
    ],
    "日期": [
      {
        "probability": {
          "average": 0,
          "min": 0
        },
        "location": {
          "top": 1,
          "left": 1,
          "width": 0,
          "height": 0
        },
        "word": "20180330"
      }
    ]
  ]
}
```

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。

- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天请求量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216102	service not support	请求了不支持的服务，请检查调用的url
216110	appid not exist	appid不存在，请确保调用接口所使用的应用归属于创建模型的百度云账号，且已开通该接口权限
216200	empty image	图片为空，请检查后重新尝试
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
282000	internal error	服务器内部错误，可能是图片尺寸过大文字太多识别超时，如果持续出现此类错误，请在控制台提交工单联系技术支持团队
336100	model temporarily unavailable	模型长期未调用需激活，遇到该错误码请等待5分钟后再次请求，即可恢复正常，若反复重试依然报错或有疑问请在百度智能云控制台内提交工单反馈

## EasyDL 跨模态使用说明

### EasyDL跨模态整体介绍

#### 概述

Hi，您好，欢迎使用百度EasyDL定制化训练和服务平台。

EasyDL平台的跨模态模型定制能力，基于文心·跨模态大模型的领先语义理解技术，为企业/开发者提供一整套跨模态定制与应用能力。

当前EasyDL平台提供了1种模型定制能力：

- 图文匹配：定制图文匹配模型，对文本及图片信息进行深度理解，计算两者的匹配度

EasyDL平台后续将提供更多类型的跨模态模型定制能力。

#### 🔗 产品优势

#### 🔗 可视化操作

无需深度学习专业知识，通过模型创建-数据上传-模型训练-模型发布全流程可视化便捷操作，最快15分钟即可获得一个高精度模型。

#### 操作步骤

**Step 1 创建模型** 确定模型名称，记录希望模型实现的功能。

**Step 2 上传并标注数据** 不同类型的任务对应的数据格式不一致，您可以上传未标注数据并使用平台提供的标注工具进行标注。或直接上传各任务的标注数据。

**Step 3 训练模型并校验效果** 选择部署方式与算法，用上传的数据一键训练模型。

模型训练完成后，可在线校验模型效果。

**Step 4 发布模型** 根据训练时选择的部署方式，将模型以云端API的方式发布使用

更详细的操作指导，请参考各类模型的技术文档

#### 🔗 高精度效果

EasyDL跨模态任务内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

文心大模型是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

#### 🔗 灵活的部署方案

训练完成后，可将模型部署在公有云服务器上。

- 训练完成的模型存储在云端，可通过独立Rest API调用模型，实现AI能力与业务系统整合
- 具有完善的鉴权、流控等安全机制，GPU集群稳定承载高并发请求

## 图文匹配

### 整体介绍

#### 🔗 任务简介

定制图文匹配模型，可实现视觉、文本跨模态理解能力，计算图文匹配度。您只需提供图片及文本的训练数据，即可训练获得图文匹配模型。

更多详情访问：[EasyDL跨模态方向](#)

#### 🔗 应用场景

- 内容质量评论：计算互联网内容中图片与文案的匹配程度，进而量化评价内容质量，可应用于电商、广告营销、互联网社区等领域
- 图文素材推荐：基于对已有图文素材的匹配度分析，在用户进行内容创作时，推荐相匹配的图片或文案素材
- 其他：尽情脑洞大开，训练你希望实现的图文匹配模型

#### 🔗 技术特色

图文匹配模型内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本、图像数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化。

文心大模型是百度发布的产业级知识增强大模型，是千行百业AI开发的首选基座大模型。文心大模型既包含基础通用的大模型，也包含了面向重点领域和重点任务的大模型，还提供丰富的工具与平台，支撑企业与开发者进行高效便捷的应用开发。“知识增强”是文心的核心特色，文心能够

同时从大规模知识和海量多元数据中持续学习，如同站在巨人的肩膀上，训练效率和理解准确率都得到大幅提升，并具备了更好的可解释性。

## 使用流程

训练模型的基本流程如下图所示，全程可视化简易操作，在数据已经准备好的情况下，最快15分钟即可获得定制模型。



## 数据准备

### 创建数据集并导入

**创建数据集** 在训练模型之前，需要先在数据总览【创建数据集】。输入数据集名称（限制50汉字），默认生成数据集版本V1，标注类型为图文匹配，配置后点击“完成”，成功创建一条空的图文匹配数据集。

版本	数据集ID	数据量	最近导入状态	标注类型	标注状态	清洗
V2	12211	3	● 已完成	图文匹配	0% (0/3)	-
V1	12210	9	● 已完成	图文匹配	0% (0/9)	-

**导入数据** 创建数据集后，在「数据总览」页面中，找到该数据集，点击右侧操作列下的「导入」，即可进入导入数据页面，可以通过以下方式导入数据：

- 导入未标注的数据，在线进行数据标注
- 直接导入标注好的数据

不论您上传无标注信息的数据或有标注信息的数据，都需要以下述格式要求进行上传。同时目前 有标注信息 上传格式仅支持 json（平台通用）

### 导入未标注的数据 本地导入

支持上传图片、压缩包

- 目前支持图片类型为jpg, png, bmp, jpeg，图片大小限制在14M以内。
- 图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px。
- 上传已标注文件要求格式为zip格式压缩包

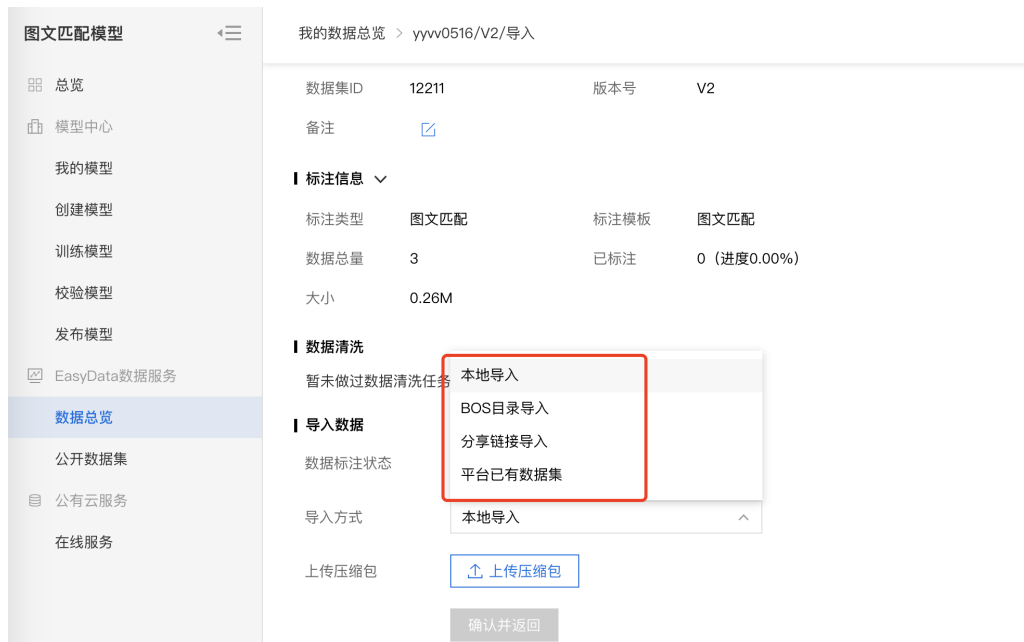
**已有数据集** 支持选择百度云 BOS 导入、分享链接导入、平台已有数据集导入；支持选择线上已有的数据集，包括其他图像类模型的数据集

- BOS目录导入格式要求：请确保将全部图片已保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入
- 分享链接导入请确保将全部图片已保存至同一压缩包，压缩包仅支持zip格式，压缩前源文件大小限制5G以内；仅支持来自百度BOS、阿里OSS、腾讯COS、华为OBS的共享链接
- 图片类型支持jpg/png/bmp/jpeg，单图需小于14M，长宽比小于 3:1，其中最长边需要小于4096px，最短边需要大于30px
- 您的账户下图片数据集大小限制为10万张图片，如果需要提升数据额度，可在平台提交工单



导入已标注的数据 本地导入 上传压缩包，标注格式仅支持 json（平台通用）

- 上传已标注文件要求格式为zip格式压缩包，同时压缩前源文件大小在5GB以内
- 压缩包内需要包括图片源文件（jpg/png/bmp/jpeg）及同名的json格式标注文件，详细请见示例压缩包 已有数据集 支持选择百度云BOS导入、分享链接导入、平台已有数据集导入，标注格式仅支持 json（平台通用）
- BOS目录导入格式要求：请确保将全部图片已保存至同一层文件目录，该层目录下子文件目录及非相关内容（包括压缩包格式等）不导入
- 分享链接导入请确保将全部图片已保存至同一压缩包，压缩包仅支持zip格式，压缩前源文件大小限制5G以内；仅支持来自百度云BOS、阿里OSS、腾讯COS、华为OBS的共享链接
- 图片类型支持jpg/png/bmp/jpeg，单图需小于14M，长宽比小于 3:1，其中最长边需要小于4096px，最短边需要大于30px
- 您的账户下图片数据集大小限制为10万张图片，如果需要提升数据额度，可在平台提交工单



## 在线标注

**在线标注 Step 1 进入标注页面** 上传未标注的数据后，可以通过以下方式进入标注页面：

- 在「数据总览」页面，该数据集对应的操作列下，点击「查看与标注」，即可进入标注页面



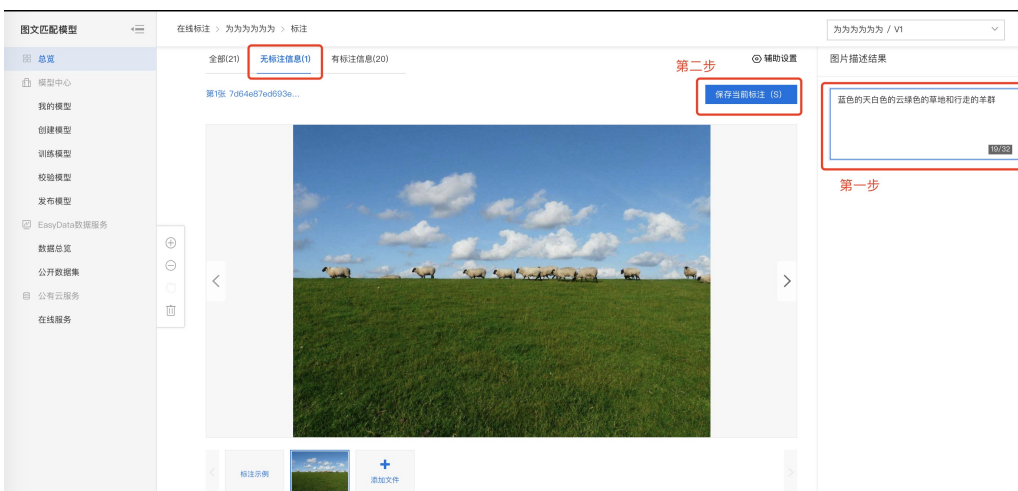
**Step 2 进行文本标注**

针对尚未进行标注的数据，通过 ([ 方式进行标注：

- 选中右侧文本框输入描述文本，文本长度限制在32个汉字。
- 点击保存文字描述，点击下一张图片，您将开始对下一张图片进行文本描述。

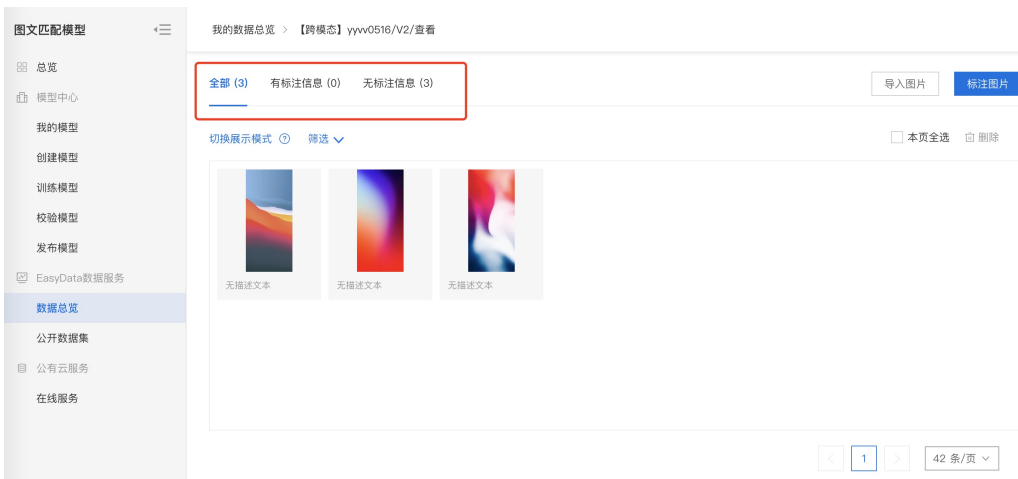
针对 ([ 进行标注的数据，通过 ([ 方式进行标注修改：

- 进入需修改标签的文本标注页面，选中 ([ 右边文本框重新编辑描述文本，文本长度限制在32个汉字。
- 点击保存文字描述，点击下一张图片，您将开始对下一张图片进行文本描述。



**Step 3 查看标注信息** 通过 ([ 方式查看 ([ 标注的文本信息：

- 在「数据总览」页面，该数据集对应的操作列下，点击「查看」，进入查看标注页面后，点击「有标注信息」，即可查看标注图文的相应情况



- 点击图文视角切换，转换为文字列表信息，点击查看，可 ([ 查看标注图文的相应情况



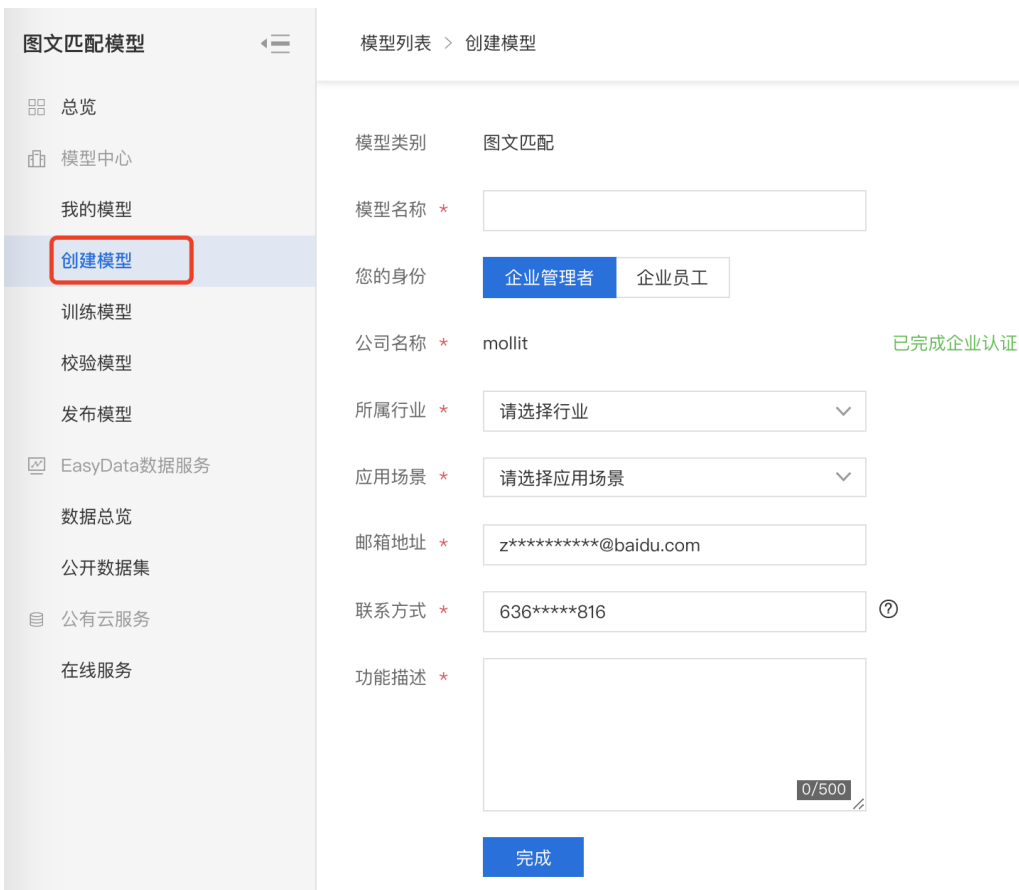


### 模型训练

#### 创建模型

**创建模型** 在模型中心目录中选择「创建模型」，填写模型名称、模型归属、所属行业、应用场景、邮箱地址、联系方式、功能描述等信息，即可创建模型。

目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练，若需要创建超过10个以上的模型，请在百度云控制台内[提交工单](#)反馈。



模型创建成功后，即可在「我的模型」中看到刚刚创建的模型。

注：

- 1.创建模型后可持续新增模型版本，因此不必每次训练模型都创建模型。
- 2.目前单个用户在每种类型的模型下最多可创建10个模型，每个模型均支持多次训练。
- 3.如果您是 enterprise 用户，建议您按照真实企业信息进行填写，便于EasyDL团队后续更好的为您服务。

#### 发起训练

**\*\*训练模型\*\***

完成数据的标注，或提交已标注的数据后，即可在「模型中心」目录中点击「训练模型」，开始模型的训练。

按以下步骤操作，启动模型训练：



**Step 1 选择模型** 选择此次训练的模型 **Step 2 训练配置**

**部署方式**

可选择「公有云部署」。

**选择算法**

您可以根据训练的需求，选择「高精度」或「高性能」算法。不同的算法将影响训练时间、预测速度与模型准确率。

- 如果您选择了高精度的模型，模型预测准确率更高，少于1000条样本同样有很好的效果。使用高精度的算法训练模型将会耗时更久，实验环境下1000个样本，预计在20-60分钟左右完成训练
- 高性能算法即将对外提供。相同训练数据量的情况下，训练耗时更短，模型预测速度更快。使用10000条训练样本，将在10min内完成训练。同样的数据量情况下，效果比高精度的模型4-5%

「高精度」算法内置**文心大模型**，将大数据预训练与多源丰富知识相结合，通过持续学习技术，不断吸收海量文本数据中词汇、结构、语义等方面的新知识，实现模型效果不断进化

**Step 3 添加数据****添加训练数据**

- 可选择多个数据集
- 训练时间与数据量大小和您选择的模型类型有关，如果您选择的是高性能的模型，使用10000条训练样本将在10min内完成训练；如果您选择的是高精度的模型，使用10000条训练样本，将在20-60min完成训练

**添加自定义测试集**

上传不包含在训练集的测试数据，可获得更客观的模型效果评估结果。

**添加自定义测试集的目的：**

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果

期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可

**Step 4 训练模型**

点击「开始训练」，即可开始训练模型。

- 训练时间与数据量大小、选择的算法、训练环境有关
- 训练环境可选择GPU P40或GPU V100。其中GPU P40可以免费使用；GPU V100训练速度更快，需要付费使用，可参考[价格说明](#)
- 模型训练过程中，可以设置训练完成的短信提醒并离开页面

平台提供付费算力，付费算力可用于模型训练，可根据实际需求购买算力使用时长。

各类算力价格如下：

算力环境	规格	算力	速度比例	价格
CPU	CPU_16核_64G	/	/	单卡 ¥ 4.02/小时
GPU P4	TeslaGPU_P4_8G显存单卡_12核CPU_40G内存	5.5 TeraFLOPS	1	单卡 ¥ 4.02/小时
GPU P40	TeslaGPU_P40_24G显存单卡_12核CPU_40G内存	12 TeraFLOPS	1.47	单卡 ¥ 21.60/小时
GPU V100	TeslaGPU_V100_16G显存单卡_12核CPU_56G内存	14 TeraFLOPS	3.66	单卡 ¥ 27.00/小时

优惠政策：

为回馈开发者长期以来对EasyDL平台的大力支持，训练算力将针对单账户 x 单操作台粒度提供5小时免费训练时长（例如，每账户享有跨模态图文匹配操作台5小时免费训练时长）。

同时，用户此前购买的算力小时包仍生效使用，支持算力小时包和储值两种付费方式。算力按分钟计费，账单金额精确至小数点后2位。训练失败、训练状态为排队中时长均不纳入收费时长。

🔗 评估效果

## 模型评估

可通过模型评估报告或模型校验了解模型效果：

- 模型评估报告：训练完成后，可以在【我的模型】列表中看到模型效果，以及详细的模型评估报告。

- 模型在线校验：可以在左侧导航中找到【校验模型】，在线校验模型效果。

「完整评估结果」页面中将记录整体评估与详细评估的报告，包括该模型整体的Mean Recall、Recall@1、Recall@5、Recall@10等指标。

整体评估中，各指标的释义如下：

- Mean Recall：通过模型计算得到top1、top5、top10匹配结果的召回率平均值，该数值越大表明模型效果越好
- Recall@1：通过模型计算得到top1匹配结果的召回率平均值，该数值越大表明模型效果越好
- Recall@5：通过模型计算得到top5匹配结果的召回率平均值，该数值越大表明模型效果越好
- Recall@10：通过模型计算得到top10匹配结果的召回率平均值，该数值越大表明模型效果越好

如果在训练阶段，使用的数据集中，数据集总量在100条以内，训练出来的模型的效果评估报告的参考价值较小，建议您训练时数据量准备充足

### \*\*模型校验\*\*

在完成训练后，发布模型前，可以先进行模型校验，以确保模型在实际环境中能获得预期的性能。操作方法如下：

1. 在左侧「模型中心」目录中点击「校验模型」，进入校验模型页面
2. 选择需要校验能力的模型、部署方式、版本，点击「启动模型校验服务」
3. 校验服务启动后，在左侧上传图片，右侧文本框内输入文本，点击「校验」后，识别结果栏将输出匹配度结果，您可参考[匹配度说明](#)了解匹配度的分析方法。

#### 🔗 匹配度说明

#### 🔗 如何运用匹配度分析？

图文匹配旨在为您选出“图片—文本”匹配度最优组合。因此，您可通过“一张图片 vs 多段文本”或“多张图片 vs 一段文本”进行多次校验模型，选出最满意的图文匹配组合。

例：






- 通过上传一张图片，计算一张图片与多段文本的匹配度。经分析，上传的图片和5段文本匹配度大小不一，您可以根据模型给出的匹配度结合自身感受选出最满意的组合；

如下表所示，前三段文字描述包含了图片主体“猫”和“蝴蝶”，以及主体关系“追”，因此图文匹配度较高；第四段文字仅包含主体，未说明关系，第五段文字仅抽象的描述了主体属性，因此导致图文匹配度较低。

上传图片	匹配文本	匹配度
	草地上一只小猫在追一只蝴蝶	66.66%
	一只小猫为了追蝴蝶跳起来了	61.07%
	一只跳起来的小猫在追空中的蝴蝶	60.76%
	草地·猫·蝴蝶	47.58%
	可爱跳脱萌宠	24.73%

- 也可以针对同一段文本，计算文本与多张图片的匹配度，通过排序得到最相关的图片。

如下图所示，前三张图片符合“水墨画”特点，同时图片主体包含“熊猫”和“竹子”，以及主体关系“吃”，因此图文匹配度较高；后两张不符合“水墨画”特点，因此导致图文匹配度较低。

匹配文本	大熊猫吃竹子的水墨画				
上传图片					
匹配度	40.36%	46.50%	45.58%	34.46%	27.27%

## 效果优化

通过模型迭代、检查并优化训练数据，能够提升模型效果。 **\*\*模型迭代\*\***

一个模型很难一次性就训练到最佳的效果，通常会需要结合模型评估报告和校验结果不断扩充数据和调优。

为此平台提供了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，以获得适合业务需求的模型效果。

如果模型已经是上线状态，依然支持模型迭代，只是需要在训练完毕后更新线上服务接口，在接口地址不变的情况下可以持续优化效果。

### **\*\*检查并优化训练数据\*\***

- 检查是否存在训练数据过少的情况，建议图文标注匹配数量不少于1000个，如果低于这个量级建议扩充
- 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

## 模型部署

### 发布公有云API

## 发布模型，生成在线API

训练完毕后可以在左侧导航栏中找到「发布模型」，依次进行以下操作即可发布公有云API：

1. 选择模型
2. 选择部署方式「公有云部署」
3. 选择版本
4. 自定义服务名称、接口地址后缀
5. 申请发布

发布模型界面示意：

**图文匹配模型**

总览

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

公开数据集

公有云服务

在线服务

### 发布模型

选择模型: pm测试模型

部署方式: 公有云部署

选择版本: V1

服务名称 \*

接口地址 \* [https://aip.baidubce.com/rpc/2.0/ai\\_custom/v1/VQA/](https://aip.baidubce.com/rpc/2.0/ai_custom/v1/VQA/)

其他要求: 若接口无法满足您的需求, 请描述希望解决的问题, 500汉字以内

0/500

提交申请

**发布完成** 申请发布后, 通常的审核周期为T+1, 即当天申请第二天可以审核完成。

如果需要加急、或者遇到莫名被拒的情况, 请在百度云控制台内[提交工单](#)反馈。

#### 调用公有云API

本文档主要说明定制化模型发布后获得的API如何使用, 如有疑问可以通过以下方式联系我们:

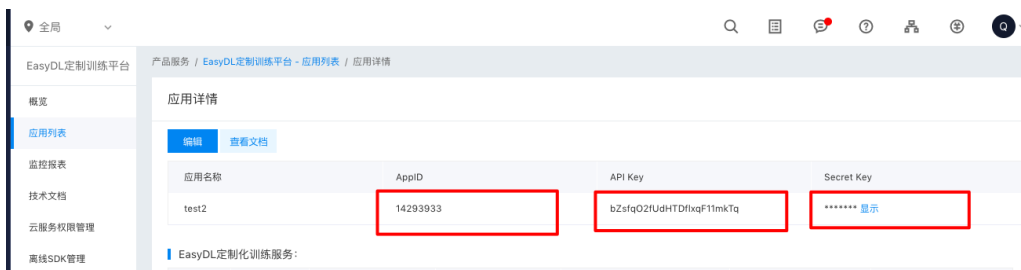
- 在百度云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#), 与其他开发者进行互动
- 加入EasyDL官方QQ群 (群号:868826008) 联系群管

#### 接口描述

基于自定义训练出的图文匹配模型, 实现个性化图文匹配相似度计算。模型训练完毕后发布可获得定制API

#### 接口鉴权

- 在[EasyDL——控制台](#)创建应用
- 应用列表页获取AK SK



#### 请求说明

HTTP 方法: POST

请求URL: 请首先进行自定义模型训练, 完成训练后可在服务列表中查看并获取url。URL参数:

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考 <a href="#">"Access Token获取"</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001的错误码很可能是因为请求方式错误，需以json方式请求。

Body请求示例：

```
{
  "text": "<UTF-8编码文本>",
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

字段名称	必须	类型	说明
text	是	string	文本，utf-8编码，支持txt格式，文本长度限制为32个字
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部

请求示例代码

```
"""
EasyDL 图文匹配 调用模型公有云API Python3实现
"""

import json
import base64
import requests
"""
使用 requests 库发送请求
使用 pip (或者 pip3) 检查我的 python3 环境是否安装了该库，执行命令
pip freeze | grep requests
若返回值为空，则安装该库
pip install requests
"""

##### 目标文本的 本地文件路径，UTF-8编码，最大长度4096汉字
TEXT_FILEPATH = "【您的测试文本地址，例如：./example.txt】"

##### 目标图片的 本地文件路径，支持jpg/png/bmp格式
IMAGE_FILEPATH = "【您的测试图片地址，例如：./example.jpg】"

##### 服务详情 中的 接口地址
MODEL_API_URL = "【您的API地址】"

##### 调用 API 需要 ACCESS_TOKEN。若已有 ACCESS_TOKEN 则于下方填入该字符串
##### 否则，留空 ACCESS_TOKEN，于下方填入 该模型部署的 API_KEY 以及 SECRET_KEY，会自动申请并显示新 ACCESS_TOKEN
ACCESS_TOKEN = "【您的ACCESS_TOKEN】"
API_KEY = "【您的API_KEY】"
SECRET_KEY = "【您的SECRET_KEY】"

PARAMS = {}
print("1. 读取目标文本 '{}'.format(TEXT_FILEPATH)")
with open(TEXT_FILEPATH, 'r') as f:
    text_str = f.read()
print("将读取的文本填入 PARAMS 的 'text' 字段")
PARAMS["text"] = text_str

print("2. 读取目标图片 '{}'.format(IMAGE_FILEPATH)")
with open(IMAGE_FILEPATH, 'rb') as f:
    base64_data = base64.b64encode(f.read())
    base64_str = base64_data.decode('UTF8')
print("将 BASE64 编码后图片的字符串填入 PARAMS 的 'image' 字段")
PARAMS["image"] = base64_str

if not ACCESS_TOKEN:
    print("3. ACCESS_TOKEN 为空，调用鉴权接口获取TOKEN")
    auth_url = "https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials\"
    "&client_id={}&client_secret={}".format(API_KEY, SECRET_KEY)
    auth_resp = requests.get(auth_url)
    auth_resp_json = auth_resp.json()
    ACCESS_TOKEN = auth_resp_json["access_token"]
    print("新 ACCESS_TOKEN: {}".format(ACCESS_TOKEN))
else:
    print("3. 使用已有 ACCESS_TOKEN")

print("4. 向模型接口 'MODEL_API_URL' 发送请求")
request_url = "{}?access_token={}".format(MODEL_API_URL, ACCESS_TOKEN)
response = requests.post(url=request_url, json=PARAMS)
response_json = response.json()
response_str = json.dumps(response_json, indent=4, ensure_ascii=False)
print("结果:\n{}".format(response_str))
```

返回说明

返回参数



字段名称	必须	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回
error_msg	否	string	错误描述信息, 当请求错误时返回
score	否	number	图像与文本信息的匹配度, 数值为0-1之间

### 在线调试

EasyDL零基础开发平台提供了 [示例代码中心\(API调试平台\)-示例代码](#) , 用于帮助开发者在线调试接口, 查看在线调用的请求内容和返回结果、复制和下载示例代码等功能, 简单易用。

**错误示例** 需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (649285136) 或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (649285136) 或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在, 请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请通过QQ群 (649285136) 联系群管手动提额
18	Open api qps request limit reached	QPS超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请通过QQ群 (649285136) 联系群管手动提额
19	Open api total request limit reached	请求总量超限额, 已上线计费的接口, 请直接在控制台开通计费, 调用量不受限制, 按调用量阶梯计费; 未上线计费的接口, 请通过QQ群 (649285136) 联系群管手动提额
100	Invalid parameter	无效的access_token参数, 请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误, 请再次请求, 如果持续出现此类错误, 请通过QQ群 (868826008) 或工单联系技术支持团队

336001	Invalid Argument	入参格式有误，比如缺少必要参数、文本编码错误等等，可检查下文本编码、代码格式是否有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队；入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请在百度智能云控制台内提交工单反馈
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或者代码格式有误。有疑问请通过QQ群（868826008）或工单联系技术支持团队
336003	Base64解码失败	图片/音频/文本格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请在百度智能云控制台内提交工单反馈
336004	输入文本或图片大小不合法	文本超出大小限制，每个文本限制512个字符（包括汉字、字符、数字或字母），有疑问请通过QQ群（868826008）或工单联系技术支持团队；图片超出大小限制，图片限4M以内，请根据接口文档检查入参格式，有疑问请在百度智能云控制台内提交工单反馈
336005	文本或图片解码失败	文本编码错误，请检查并修改文本格式；图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	text字段缺失（未上传文本）；image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（868826008）或工单联系技术支持团队

## EasyDL 零售行业版使用说明

### 零售版服务介绍

#### 简介

EasyDL是百度大脑中的一个定制化训练和服务平台，EasyDL零售版是EasyDL针对零售场景推出的行业版。EasyDL零售版提供两种服务，分别为定制商品检测服务和货架拼接服务。

**定制模型服务**是EasyDL零售版的一项服务，专门用于训练货架合规性检查、自助结算台、无人零售货柜等场景下的定制化AI模型，训练出的模型将以API的形式为客户提供服务。该服务包含以下2种定制模型：

#### 1. 商品检测模型

- 适用场景：适用于适用于货架、端架、挂架等场景的商品陈列规范核查，支持识别商品基本信息，陈列顺序、层数、场景，统计排面数量和占比
- 服务功能：
  - 商品基本信息识别：商品的名称、品牌、规格、编号；商品在图片中的坐标位置；商品识别的置信度
  - 商品陈列层数识别：商品陈列所在货架层数和货架总层数；商品陈列顺序；货架是否拍摄完整判断
  - 商品陈列场景识别：场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台
  - 商品排面占比统计：商品的排面数及排面占比；每层货架可识别商品数量及未知商品数量；货架的总空位数、每层货架空位数及货架利用率
  - 商品陈列翻拍识别：识别商品陈列照片是对手机屏幕翻拍的可能性

#### 2. 地堆检测模型

- + 适用场景：适用于堆箱、堆头、地龙等场景的商品陈列规范核查，支持识别商品基本信息，可视商品计数，纵深商品计数和占地面积
- + 服务功能：
  - + 商品基本信息识别：商品的名称、品牌、规格、编号；商品在图片中的坐标位置；商品识别的置信度；陈列顺序；可视商品计数，纵深商品计数和占地面积
  - + 商品陈列场景识别：场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台
  - + 商品陈列翻拍识别：识别商品陈列照片是对手机屏幕翻拍的可能性

**货架拼接服务**基于百度EasyDL深度学习算法，支持将多个货架局部图片或视频，组合为完整货架图片。同时支持输出在完整货架图中的商品检测结果，包含SKU的名称和数量，适用于需要在长货架进行商品检测的业务场景。

**翻拍识别服务**能够识别出通过手机翻拍出的商品陈列照片，比如商品货架陈列图片和地堆商品陈列图片，可降低人工审核人力，高效审核零售业

务中通过翻拍原有图片来造假的图片。

**价签识别服务**能够识别货架和促销活动中的价签信息，可识别各个价签在图片中的像素位置，以及价签内商品名称和价格，可用于洞察商品在线下渠道分销的价格区间。

## 🔗 功能介绍

### 🔗 定制商品检测服务

- **AI模型训练平台**

专门用于定制货架合规性检查、自助结算台、无人零售货柜等零售场景下识别商品的高精度AI模型。

- **全可视化操作**

所有模型训练相关的操作都可以在网页上进行，无需编程，仅需五步即可部署定制化AI模型。

- **预置的商品库**

预置近千种商品单品图可供客户在创建SKU时选择，用于合成训练数据，极大降低了训练数据采集和标注成本。

- **可自定义商品**

客户可根据业务需求创建属于自己的商品，商品信息支持完全自定义，充分满足客户定制化需求。

- **全面的商品信息**

商品基本信息识别

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

商品陈列层数识别

接口支持识别商品陈列所在货架层数，货架类型支持：货架、端架和立式冰柜内货架

商品陈列场景识别

接口支持识别商品陈列的场景，场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、地堆、割箱、地龙、小端架、吧台

商品排面占比统计

接口支持统计商品排面数/占比、未识别商品数、总空位数、每货架层的空位数及货架利用率

商品陈列翻拍识别

识别商品陈列照片是对手机屏幕翻拍的可能性

### 🔗 货架拼接服务

- **拼接和商品检测相结合**

支持将多个货架局部图片或视频，组合为完整货架图片，并支持输出在完整货架图中的商品检测结果，包含SKU的名称和数量。

- **丰富的服务方式**

支持三种服务方式：云服务API、完全开源的SDK以及可以直接体验的手机APP。

### 🔗 适用场景

- **货架合规性检查**

精准识别出货架、冰柜和端架上陈列商品的数量和种类，为品牌商分析陈列排面占比，重点SKU分销率、缺货率、合格率提供数据支撑。

- **互动营销**

训练定制的商品识别，实现对C端用户提交的商品图片进行识别，配合游戏规则完成闯关/抽奖式的互动营销。

## 🔗 技术优势

### • 免费训练与测试

平台提供大量免费的GPU训练资源，及每天500次免费调用量，用于模型迭代和效果验证，有效降低项目开发和测试成本。

### • 高可用模型效果

针对零售场景专项算法调优，结合图像合成与增强技术提升模型泛化能力，模型准确率可达97%+，保证模型在生产环境中具有高可用性。

### • 预置模型和数据

平台提供直接可用的商品检测API，覆盖常见商品品类；提供大量预置单品图数据，可用于训练定制模型，有效提升项目落地效率。

### • 企业级安全保障

数据加密与隔离，完善的服务调用鉴权，为客户的数据和模型提供企业级安全保障。

### • 功能完善且丰富

全面覆盖各类零售场景的商品识别需求，应对不同场景的业务需求提供多种可选服务类型。

## 🔗 与EasyDL物体检测的相同点和不同点

EasyDL零售版是EasyDL专门针对零售场景下识别商品推出的版本，相比于物体检测模型，零售版更贴合快消零售场景下的业务需求，专门用于训练货架合规性检查、自助结算台、无人零售货柜等场景下的定制化商品检测AI模型，训练出的模型可发布成云服务API，服务支持四种功能：商品基本信息识别、商品陈列层数识别、商品陈列场景识别和商品排面占比统计，适用于识别货架中的商品信息，商品计数和陈列顺序等，辅助货架商品陈列合规检查，如辅货率、陈列情况等。

### 相同点

同为检测模型，接口支持返回目标物体的名称和物体在图片上的位置。

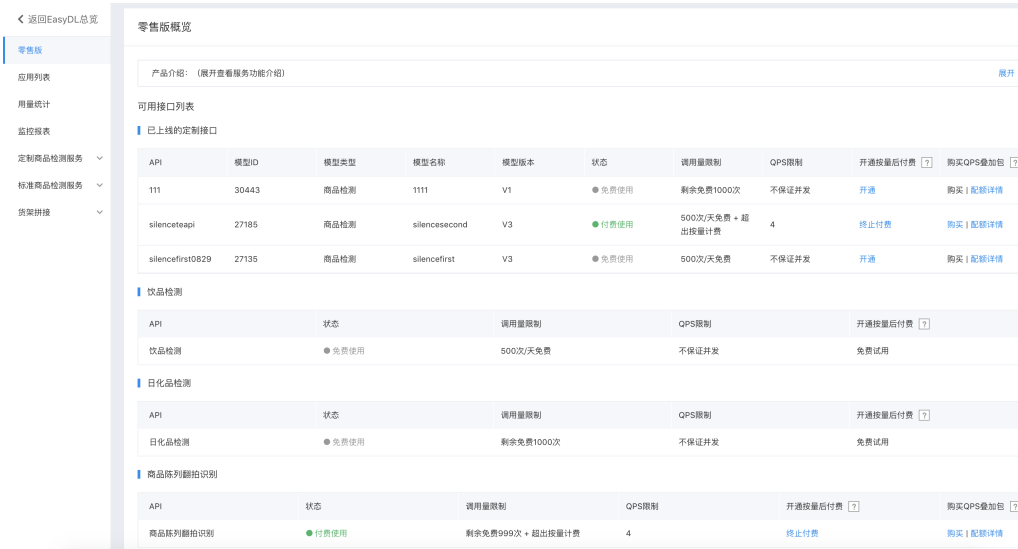
### 不同点

- 模型算法不同：零售版的模型算法专门根据零售行业的场景和业务需求做了专项优化，基于百度大脑大规模零售数据预训练，并利用商品增强合成技术将SKU单品图合成实景货架图，有针对性的提高了训练商品检测模型的精确度。
- 训练数据不同：零售版的数据除了需要标注的实景业务图片外，支持为每个SKU标签上传单品图。SKU单品图用来降低实景图即训练数据采集和标注成本的。为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分的成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且**单品图无需进行任何标注**，即可让模型学习到这个SKU各个角度的样子。**合成图片过程在训练阶段自动完成，无需操作操作和进行标注**。EasyDL零售版平台预置了近千种商品，每个商品预置了50张左右的单品图，绝大多数情况下无需再自行上传单品图。
- 云服务API功能不同：零售版云服务API支持四种功能：商品基本信息识别、商品陈列层数识别、商品陈列场景识别和商品排面占比统计；物体检测云服务API仅支持商品基本信息识别。

## 购买指南

### 🔗 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。



定制商品检测服务

价目表 - 按调用量后付费

定制商品检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

2. 商品陈列层数识别（可选）

接口支持识别商品陈列所在货架层数，货架类型支持：货架、端架和立式冰柜内货架

3. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

4. 商品排面占比统计（可选）

接口支持统计商品排面数/占比、未识别商品数、空位数及货架利用率

5. 商品陈列翻拍识别（可选）

识别商品陈列照片是对手机屏幕翻拍的可能性

付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用三项服务，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列层数识别和商品陈列场景识别两项服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

月调用量（万次）	单次调用价格（元）	QPS限制	说明
0<月调用量<=15	0.009	4	服务器支持每秒处理4次查询
15<月调用量<=150	0.008	4	服务器支持每秒处理4次查询
150<月调用量	0.007	4	服务器支持每秒处理4次查询

- 商品陈列层数识别（可选），单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.04	4	服务器支持每秒处理4次查询

- 商品陈列场景识别 (可选) , 单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品排面占比统计 (可选) , 单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.02	4	服务器支持每秒处理4次查询

- 商品陈列翻拍识别 (可选) , 单次调用额外收取费用

单次调用价格 (元)	QPS限制	说明
0.05	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制商品检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制商品检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制商品检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

- 商品基本信息识别的费用为43,650元，明细如下：

前15万次落入0~15w阶梯，单次调用0.009元/次，费用为1,350元；

中间15万~150万次落入15~150w阶梯，单次调用0.008元/次，费用为10,800元；

最后150万~600万次落入大于150w阶梯，单次调用0.007元/次，费用为31,500元；

共计43,650元

- 商品陈列层数识别的费用为360,000元，明细如下：

月调用量为600万次，单次调用0.04元/次，费用为240,000元

- 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计319,650元。

### 🔗 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1050元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制商品检测API的所有服务功能均有效

## 定制地堆检测服务

### 价目表 - 按调用量后付费

定制地堆检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

#### 1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

#### 2. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

#### 3. 商品陈列翻拍识别（可选）

识别商品陈列照片是对手机屏幕翻拍的可能性

### 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用商品陈列场景识别服务功能，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列场景识别服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

单次调用价格（元）	QPS限制	说明
0.016	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品陈列翻拍识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.05	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制地堆检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制地堆检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制地堆检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

1. 商品基本信息识别的费用为96,000元，明细如下：

月调用量为600万次，单次调用0.016元/次，费用为96,000元

2. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计132,000元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	60元/天
按月购买	1200元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制地堆检测API的所有服务功能均有效

### 翻拍识别服务

#### 价目表 - 按调用量后付费

##### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

##### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 价目表 - 调用量次数包

如果对调用次数有预估，可以选择购买单次调用价格更低的次数包，价格如下：



规格	价格	QPS限制	有效期
1万次	490 元	4	1年
10万次	4,800 元	4	1年
100万次	45,000 元	4	1年
500万次	212,500 元	4	1年
1000万次	420,000 元	4	1年
2000万次	800,000 元	4	1年

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

#### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

#### 价签识别服务

#### 价目表 - 按调用量后付费

##### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	2	服务器支持每秒处理2次查询

注：调用失败不计费

##### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
价签识别	累计1000次	1	服务器支持每秒处理1次查询

注：成功调用与失败调用均消耗免费额度

#### 价目表 - 调用量次数包

如果业务上对调用次数有预估，可以选择购买单次调用价格更低的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	475 元	2	1年 (366天)
5万次	2,250 元	2	1年 (366天)
10万次	4,250 元	2	1年 (366天)
20万次	8,000 元	2	1年 (366天)
50万次	18,750 元	2	1年 (366天)
100万次	35,000 元	2	1年 (366天)
500万次	150,000 元	2	1年 (366天)

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

#### 🔗 价目表 - QPS叠加包

开通付费后，免费QPS为2，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	100元/天
按月购买	2000元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

#### 🔗 货架拼接服务

货架拼接服务支持按任务数后付费、任务次数包预付费和并发任务叠加包预付费三种计费方式。

#### 🔗 价目表 - 按任务数后付费

##### 付费调用

每个账户享有累计200次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

任务数	价格 (元)	并发任务数限制	说明
每次拼接任务	0.2	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务启动后失败和运行前终止不计费，任务成功和运行后终止会计费用

##### 免费额度

每个账号享有一定量免费调用额度，如下表：

服务	免费任务额度	并发任务数限制	说明
货架拼接	累计200次	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务成功与失败调用均消耗免费额度

#### 🔗 价目表 - 任务次数包

如果对拼接任务次数有预估，可以选择购买**单次任务价格更低**的次数包，价格如下：

规格	价格	并发任务数限制	有效期
1千次	200 元	1	1年
1万次	1,900 元	1	1年
10万次	18,000 元	1	1年
100万次	150,000 元	1	1年
500万次	600,000 元	1	1年

购买后不可退款，任务次数包使用完后，开始按调用量每个任务0.2元收取费用

#### 🔗 价目表 - 并发任务叠加包

开通付费后，并发任务数限制为1，如果有更多的并发请求需要，可以根据业务需求按天或按月购买并发任务叠加包，价格如下：

购买方式	每并发任务价格
按天购买	2元/天
按月购买	40元/月

购买 并发任务叠加包需保证已开通按量后付费或购买任务次数包

购买的并发任务叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

#### 🔗 余额不足提醒与欠费处理

##### 🔗 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

##### 🔗 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

## 场景范例

### 拓展门店验证

#### 🔗 场景简介

对未覆盖铺货的待拓展门店数据，使用门脸文字识别服务确认门店是否真实存在，确认存在并转化为已铺货的业务门店后，可以在平台上移动至业务门店库，可在业务门店拜访场景时用作到访打卡验证。

#### 🔗 使用步骤流程

- 1.创建门店库
- 2.导入待确认门店列表
- 3.使用门脸文字识别验证
- 4.移入业务门店

#### 🔗 最佳实践

##### 1.创建门店库

进入到EasyDL零售版[门店管理页面](#)，参考文档[门店库创建](#) 创建门店库。

## 2.导入待确认门店列表

创建好门店库后，进入到门店库，参考文档 [门店导入](#) 将本地的待确认的未铺货门店数据导入至门店库待确认门店列表。

## 3.使用门脸文字识别验证

为了验证收集到的未铺货门店数据是真实存在的，可以使用门脸文字识别服务进行验证，服务会在鉴权对应的门店库待确认门店列表中，根据待确认门店ID找到对应门店，通过传入的门脸图片和经纬度坐标进行真实性判断，满足判断条件即真实存在，服务或返回对应标识结果，门店会移动至待拓展门店列表中。

### API URL

门店识别API URL：[https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/facade\\_v2](https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/facade_v2)

详细调用方法和参数解释，请见 [门脸文字识别技术文档](#)

### 请求示例

请求API时传入tbc\_store\_id，调用门店识别API的请求参数示例如下：

```
{
  "image": "<base64数据>",
  "tbc_store_id": "QSDD121242331",
  "location": "116.271422,40.228393",
  "location_type": 3
}
```

### 返回示例

仅words\_result中第一个结果会与待确认门店库中tbc\_store\_id对应门店信息进行真实性判断，words\_result.is\_match是判定标识字段，1表示拍摄的门店与门店库中tbc\_store\_id对应门店信息匹配，0则表示不匹配。

```
{
  "log_id": 3257574830993687599,
  "words_result_num": 2,
  "words_result": [
    {
      "brief": "",
      "score": 0.9812122343,
      "words": "Lawson",
      "is_match": 1,
      "type": "ocr",
      "channel": {
        "store_type": "便利店",
        "chain_type": "连锁便利店"
      }
    },
    {
      "score": 0.9700000286,
      "type": "ocr",
      "brief": "",
      "words": "天天超市",
      "channel": {
        "store_type": "小店",
        "chain_type": "独立超市"
      }
    }
  ]
}
```

## 4.移入业务门店

经过第3点验证通过的门店，会自动移动到待拓展门店列表中，待业务侧确认该待拓展的门店已经成功铺货后，可在门店库详情页面，将这些门店移入至业务门店列表，移入后，可使用门脸文字识别服务进行日常业务门店拜访打卡的验证，该场景范例可参考文档：[业务门店拜访](#)

## 业务门店拜访

对已铺货的业务门店拜访过程中，使用门脸文字识别服务验证拜访打卡的是否为当前在SFA里选择的门店。

#### ☞ 使用步骤流程

- 1.创建门店库
- 2.导入业务门店列表
- 3.使用门脸文字识别验证

#### ☞ 最佳实践

##### 1.创建门店库

进入到EasyDL零售版[门店管理页面](#)，参考文档[门店库创建](#) 创建门店库。

##### 2.导入业务门店列表

创建好门店库后，进入到门店库，参考文档[门店导入](#) 将本地的业务门店数据导入至门店库[业务门店列表](#)。

##### 3.使用门脸文字识别验证

为了验证拜访打卡的门店与在SFA中选择的门店一致，可以使用门脸文字识别服务进行验证，服务会在鉴权对应的门店库业务门店列表中，通过传入的门脸图片和经纬度坐标进行匹配度检索，检索到结果的会返回在门店库中对应门店的ID，如果ID和SFA中选择的门店ID一致，则说明当前拍摄的门店，服务或返回对应标识结果。

#### API URL

门店识别API URL：[https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/facade\\_v2](https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/facade_v2)

详细调用方法和参数解释，请见[门脸文字识别技术文档](#)

#### 请求示例

请求API时传入tbc\_store\_id，调用门店识别API的请求参数示例如下：

```
{
  "image": "<base64数据>",
  "location": "116.271422,40.228393",
  "location_type": 3
}
```

#### 返回示例

当在业务门店列表中检索到匹配的门店时，会通过words\_result中的第一个结果返回，words\_result.type是判定标识字段，如果返回"image\_search"，表示该结果为检索到的结果，"store\_id"和"words"分别为门店库中存的门店ID和门店名称；如果返回"ocr"，表示文字识别结果，即在门店库中没有检索到门店，此时返回的是OCR文字识别的结果。

```
{
  "log_id": "3257574830993687599",
  "words_result_num": 2,
  "words_result": [
    {
      "store_id": "SFA123456",
      "brief": "",
      "score": 1.0,
      "words": "Lawson",
      "is_match": -1,
      "type": "image_search",
      "channel": {
        "store_type": "便利店",
        "chain_type": "连锁便利店"
      }
    },
    {
      "score": 0.9700000286,
      "type": "ocr",
      "brief": "",
      "words": "天天超市",
      "channel": {
        "store_type": "小店",
        "chain_type": "独立超市"
      }
    }
  ]
}
```

## 店内陈列洞察

### 场景简介

洞察已铺货的门店店内商品陈列情况，如识别货架中的商品信息，商品计数和陈列顺序等，辅助货架商品陈列合规检查，如铺货率、陈列情况等。

### 使用步骤流程

- 1.训练AI识别模型
- 2.调用云服务API
- 3.查看商品陈列信息

### 最佳实践

#### 1.训练AI识别模型

参考文档 [快速训练一个模型](#)，训练一个支持识别所需SKU的AI模型。

#### 2.调用云服务API

- 参考文档 [服务功能](#)，按需求开启云服务API的服务功能。
- 参考文档 [商品检测API调用方法](#)，按需要获取的业务指标进行字段取值。

#### 请求示例

请求API时，请求参数image\_store传入业务门店ID，请求参数示例如下：

```
{
  "image": "<base64数据>",
  "image_store": "SFA123456"
}
```

#### 3.查看商品陈列信息

当调用API时，在请求参数"image\_store"中传入业务门店ID，可在平台业务门店列表中的门店详情中查看到对应门店ID的商品陈列信息。

## 异常拍照监测

### 场景简介

对拜访门店过程中拍摄的照片进行异常拍摄监测，平台目前支持识别2类异常行为：翻拍和窜拍。

- 商品陈列翻拍识别能够识别出通过手机翻拍出的商品陈列照片，比如商品货架陈列图片和地堆商品陈列图片，可降低人工审核人力，高效审核零售业务中通过翻拍原有图片来造假的图片。
- 窜拍识别服务可对用户上传的数据进行疑似窜拍图（相似图）分组，用户可按照「人」、「门店」、「时间」定义需要识别的范围，服务按定义的范围返回有相似图存在的图片组。

### 使用步骤流程

- 1.调用商品陈列翻拍识别API
- 2.调用窜拍识别API
- 3.查看翻拍信息
- 4.查看窜拍信息

### 最佳实践

#### 1.调用商品陈列翻拍识别API

参考文档 [翻拍识别API调用方法](#) 调用服务API。

#### 请求示例

请求参数示例如下：

```
{
  "image": "<base64数据>"
}
```

#### 返回示例

设定一个判定为翻拍图片的阈值，即如果results.name为"recapture"的"score"大于这个值，则认为这张图片是翻拍。

```
{
  "log_id": 3142711278302327859,
  "results": [
    {
      "name": "original",
      "score": 0.9980294108390808
    },
    {
      "name": "recapture",
      "score": 0.001970605691894889
    }
  ]
}
```

#### 2.调用窜拍识别API

参考文档 [窜拍识别API调用方法](#)，调用服务API，接口调用流程如下：

- 1) 创建图集，获得图集ID

创建图集API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_dataset](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_dataset)

- 2) 指定图集ID，把要识别窜拍的图片上传到该图集

上传图片API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/upload](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/upload)

- 3) 创建任务，指定图集ID，对该图集做窜拍识别，获取任务ID

创建任务API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_task](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_task)

4) 指定任务ID，轮询任务结果，可以每隔一定时间调用API查询任务结果，比如10s查询一次，任务完成后可以查到结果

查询结果API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/query](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/query)

另外，可通过图集列表API和任务列表API查询所有图集和任务。

### 3.查看翻拍信息

当调用API时，在请求参数"image\_store"中传入业务门店ID，可在平台业务门店列表中的门店详情中查看到对应门店ID的商品陈列信息，其中包含翻拍信息。

### 4.查看窜拍信息

当调用API时，在请求参数"image\_store"中传入业务门店ID，可在平台业务门店列表中的门店详情中查看到对应门店ID的窜拍信息。

## 定制商品检测服务

### 服务介绍

#### 简介

定制商品检测服务是EasyDL零售版的一项服务，专门用于训练货架合规性检查、自助结算台、无人零售货柜等场景下的定制化商品检测AI模型，训练出的模型将以API的形式为客户提供服务，服务支持四种功能：商品基本信息识别、商品陈列层数识别、商品陈列场景识别和商品排面占比统计，适用于识别货架中的商品信息，商品计数和陈列顺序等，辅助货架商品陈列合规检查，如铺货率、陈列情况等。

#### 功能介绍

- AI模型训练平台

专门用于定制货架合规性检查、自助结算台、无人零售货柜等零售场景下识别商品的高精度AI模型。

- 全可视化操作

所有模型训练相关的操作都可以在网页上进行，无需编程，仅需五步即可部署定制化AI模型。

- 预置的商品库

预置近千种商品单品图可供客户在创建SKU时选择，用于合成训练数据，极大降低了训练数据采集和标注成本。

- 可自定义商品

客户可根据业务需求创建属于自己的商品，商品信息支持完全自定义，充分满足客户定制化需求。

- 全面的商品信息

#### 商品基本信息识别

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

#### 商品陈列层数识别

接口支持识别商品陈列所在货架层数，货架总层数以及商品的陈列顺序，货架类型支持：货架、端架、冷风柜和立式冰柜内货架

#### 商品陈列场景识别

接口支持识别商品陈列的场景，场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台

#### 商品排面占比统计

接口支持统计商品排面数/占比、未识别商品数、总空位数、每货架层的空位数及货架利用率

#### 商品陈列翻拍识别

识别商品陈列照片是对手机屏幕翻拍的可能性



## 使用流程

1. 创建模型
2. 创建SKU
3. 实景图片上传和标注
4. 训练并校验模型
5. 发布模型，获得定制的商品检测API

## 特色优势

### 免费训练与测试

平台提供大量免费的GPU训练资源，及每天1000次免费调用量，用于模型迭代和效果验证，有效降低项目开发和测试成本

### 高可用模型效果

针对零售场景专项算法调优，结合图像合成与增强技术提升模型泛化能力，模型准确率可达97%+，保证模型在生产环境中具有高可用性

### 预置模型和数据

平台提供直接可用的商品检测API，覆盖常见商品品类；提供大量预置单品图数据，可用于训练定制模型，有效提升项目落地效率

### 企业级安全保障

数据加密与隔离，完善的服务调用鉴权，为客户的数据和模型提供企业级安全保障

## 适用场景

### 货架合规性检查

精准识别出货架、冰柜和端架上陈列商品的数量和种类，为品牌商分析陈列排面占比，重点SKU分销率、缺货率、合格率提供数据支撑

### 互动营销

训练定制的商品识别，实现对C端用户提交的商品图片进行识别，配合游戏规则完成闯关/抽奖式的互动营销

## 购买指南

## 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。

API	模型ID	模型类型	模型名称	模型版本	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
111	30443	商品检测	1111	V1	● 免费使用	剩余免费1000次	不保证并发	开通	购买   配账详情
silenceapi	27185	商品检测	silencesecond	V3	● 付费使用	500次/天免费 + 超出按量计费	4	终止付费	购买   配账详情
silencefirst0829	27135	商品检测	silencefirst	V3	● 免费使用	500次/天免费	不保证并发	开通	购买   配账详情

API	状态	调用量限制	QPS限制	开通按量后付费
饮品检测	● 免费使用	500次/天免费	不保证并发	免费试用

API	状态	调用量限制	QPS限制	开通按量后付费
日化品检测	● 免费使用	剩余免费1000次	不保证并发	免费试用

API	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
商品陈列翻拍识别	● 付费使用	剩余免费999次 + 超出按量计费	4	终止付费	购买   配账详情

## 定制商品检测服务

### 价目表 - 按调用量后付费

定制商品检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

## 1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

## 2. 商品陈列层数识别（可选）

接口支持识别商品陈列所在货架层数，货架类型支持：货架、端架和立式冰柜内货架

## 3. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

## 4. 商品排面占比统计（可选）

接口支持统计商品排面数/占比、未识别商品数、空位数及货架利用率

## 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用三项服务，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列层数识别和商品陈列场景识别两项服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见[服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

月调用量（万次）	单次调用价格（元）	QPS限制	说明
0<月调用量<=15	0.009	4	服务器支持每秒处理4次查询
15<月调用量<=150	0.008	4	服务器支持每秒处理4次查询
150<月调用量	0.007	4	服务器支持每秒处理4次查询

- 商品陈列层数识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.04	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品排面占比统计（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.02	4	服务器支持每秒处理4次查询

注：调用失败不计费

## 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制商品检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制商品检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制商品检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

1. 商品基本信息识别的费用为43,650元，明细如下：

前15万次落入0~15w阶梯，单次调用0.009元/次，费用为1,350元；

中间15万~150万次落入15~150w阶梯，单次调用0.008元/次，费用为10,800元；

最后150万~600万次落入大于150w阶梯，单次调用0.007元/次，费用为31,500元；

共计43,650元

2. 商品陈列层数识别的费用为360,000元，明细如下：

月调用量为600万次，单次调用0.04元/次，费用为240,000元

3. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计319,650元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1050元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制商品检测API的所有服务功能均有效

### 定制地堆检测服务

#### 价目表 - 按调用量后付费

定制地堆检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

2. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

#### 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用商品陈列场景识别服务功能，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列场景识别服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- **商品基本信息识别（必选）**，按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

单次调用价格（元）	QPS限制	说明
0.016	4	服务器支持每秒处理4次查询

- **商品陈列场景识别（可选）**，单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制地堆检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制地堆检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制地堆检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

1. 商品基本信息识别的费用为96,000元，明细如下：

月调用量为600万次，单次调用0.016元/次，费用为96,000元

2. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计132,000元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	60元/天
按月购买	1200元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制地堆检测API的所有服务功能均有效

## 快速训练一个模型

### 简介

本文档介绍使用EasyDL零售版商品检测快速训练一个识别可口可乐的商品检测模型，基本流程如下：

#### 1. 创建模型

- 2.创建SKU
- 3.上传和标注训练数据
- 4.训练模型
- 5.发布模型
- 6.使用模型API

## 🔗 步骤1.创建模型

这个步骤将会介绍如何创建模型

### 进入创建模型页面

在[EasyDL零售版商品检测产品主页](#)点击【开始训练】按钮进入到[模型训练页](#)，下面会出现两种情况：

- 第一种，如果您没有登录百度智能云，则会跳转到百度智能云登录页面，没有百度账户的客户请先[注册百度账户](#)。登录后，会跳转到[模型概览页](#)，点击【商品检测】卡片上的【立即定制】按钮，会跳转模型训练页面的创建模型页。
- 第二种，如果您已登录，会直接进入到【我的模型】页，该页面能够管理已经创建的模型，点击左侧列表中的【创建模型】进入创建模型页面。

### 创建模型

进入创建模型页面后你会看到如下图中展示的内容

模型中心

模型列表 > 创建模型

模型类别: 商品检测

模型名称:

模型归属:  公司  个人

请输入公司名称

应用场景:

不同应用场景对应不同训练算法, 请根据真实应用场景选择

邮箱地址:

联系方式:

功能描述:

0/500

下一步

需要填写的项目如下：

- 模型名称  
模型的名称
- 模型归属  
模型是属于公司的，还是属于个人的，如果是前者，请填写公司名称
- 应用场景

提示：请根据真实业务应用场景选择，选择的场景将会关联后端数据增强算法，若不确定，请选择“其他”

可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

- 邮箱地址  
用于联系到您的邮箱地址

- 联系方式

有效的联系方式将有助于后续模型上线的人工快速审核，以及更快的百度官方支持，推荐填写个人手机号码

- 功能描述

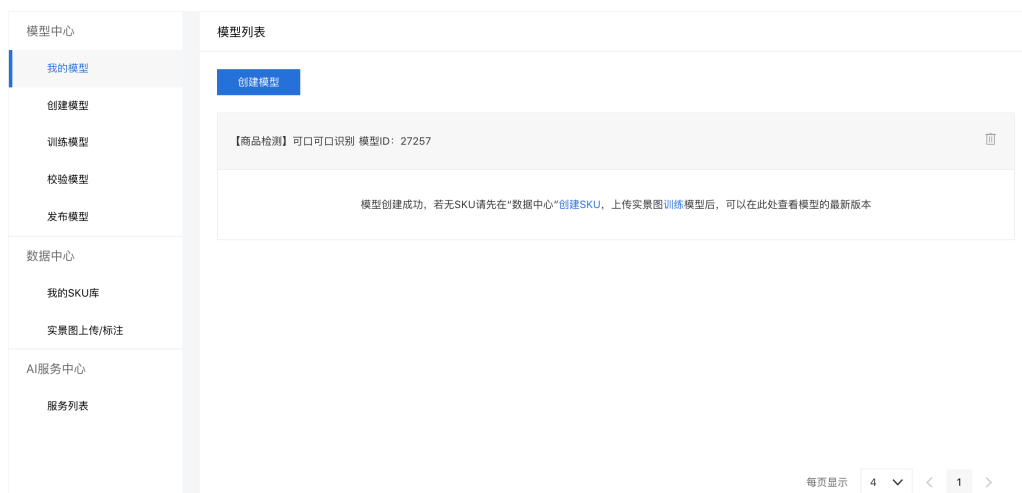
描述改模型将要应到的业务场景，详细的描述，在获取官方支持时，能帮助我们为您提供准确的使用建议

完成所有填写项后点击【下一步】按钮完成模型创建，创建完成后会跳转到【我的模型】页面。

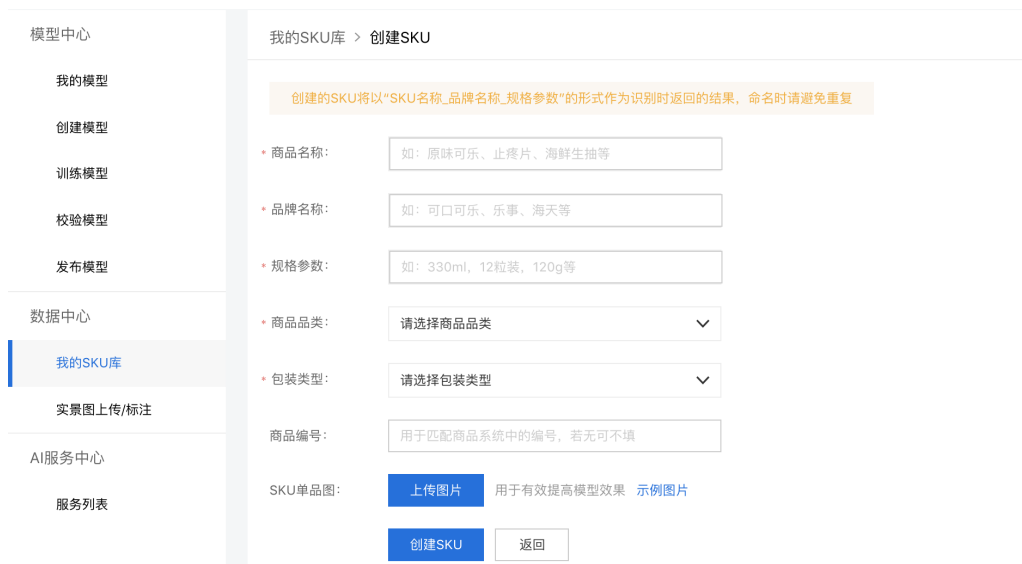
## 🔗 步骤2.创建SKU

这个步骤将会介绍如何创建SKU，SKU是客户需要检测的商品，在训练品台上有两个作用，其一是“SKU名称\_品牌名称\_规格参数”用于标注训练数据的标签，二是SKU的单品图片用于商品增强合成技术，提高模型效果。

完成上一个步骤后，会跳转到【我的模型】页面，这时您会看到如下图展示的内容，由于模型还未训练，所以模型列表中没有显示模型的效果，在训练模型前，需要先完成SKU的创建。



点击左侧列表中的【我的SKU】进入SKU管理页面，点击【创建SKU】按钮进入创建SKU页面，您会看到如下图展示的内容



**提示：**在调用API接口识别SKU时，识别结果中SKU的名字是以“SKU名称\_品牌名称\_规格参数”的形式返回的，所以在填写SKU名称、品牌名称和规格参数时避免这三项内容重复。

需要填写的项目如下：

- SKU名称

SKU的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等

- 品牌名称

SKU的品牌名称，如可口可乐，乐事，伊利等

- 规格参数

SKU的规格，如330ml，500g，20片等

- 商品品类

可选择的有饮品、药品、保健品、零食、香烟、调味品、日用品和其他

- 包装类型

可选择的有瓶装、罐装、袋装、盒装和其他

- 商品编号

如果您自身的业务系统中有现成SKU对应的商品编码，比如商品条形码，可以填在该填写框中，之后模型接口将支持返回该内容，用于您快速匹配SKU

- SKU单品图

**SKU的单品图不是模型训练的必须数据**，其作用为用来合成实景图，连同手工标注的实景图一起用于训练，降低实景图即训练数据采集和标注成本。拍摄角度和上传张数基本原则是覆盖实际检测场景可能出现的角度，具体请参考[SKU单品图数据](#)文档中进行单品图采集。

当每个SKU的实景图大于20张时，可以先不上传SKU单品图进行训练，后续提升模型效果以补充实景图为主，如果无法提供足量的实景图数据，可以通过上传SKU单品图来提升模型效果。

完成填写和上传SKU单品图上传后，页面内容显示如下图所示

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

数据中心

我的SKU库

实景图上传/标注

AI服务中心

服务列表

我的SKU库 > 创建SKU

创建的SKU将以“SKU名称\_品牌名称\_规格参数”的形式作为识别时返回的结果。命名时请避免重复

\* 商品名称: 原味可乐

\* 品牌名称: 可口可乐

\* 规格参数: 500ml

\* 商品品类: 饮品

\* 包装类型: 瓶装

商品编号: 用于匹配商品系统中的编号。若无可不填

SKU单品图: [上传图片](#) 用于有效提高模型效果 [示例图片](#)

您已上传12张SKU单品图

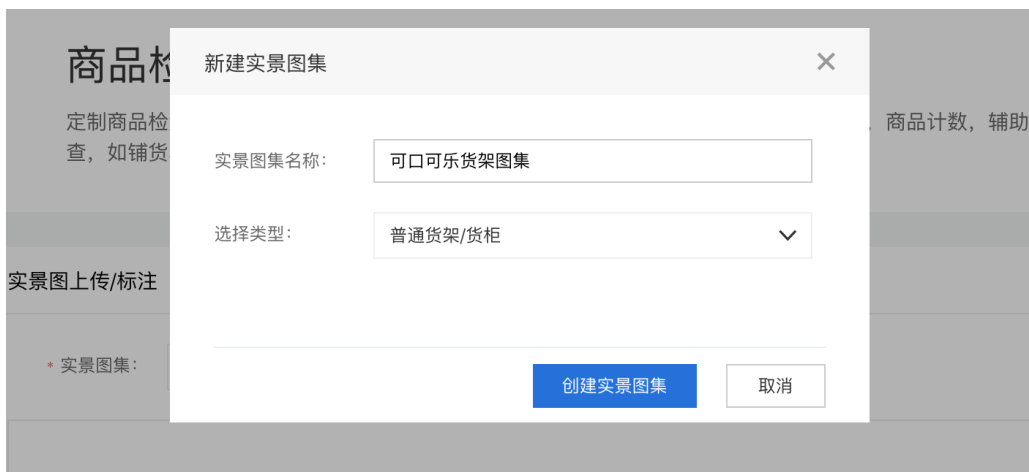
[创建SKU](#) [返回](#)

点击【创建SKU】按钮完成创建，点击后回到【我的SKU库】，SKU列表中的SKU图数需要大约5秒的时间进行计算，刷新页面即可显示SKU单品图片数。

### 步骤3.上传和标注训练数据

这个步骤将会介绍如何上传和标注训练数据，训练数据是SKU在货架上的实景图，需要客户从真实的业务场景中采集，这些图片在被正确标注中，可以用于训练成模型。

完成上一个步骤后，在左侧列表中点击【实景图上传/标注】进入上传和标注页面，在上传前请在实景图集选择栏内创建实景图集，如下图所示



需要填写的项目如下：

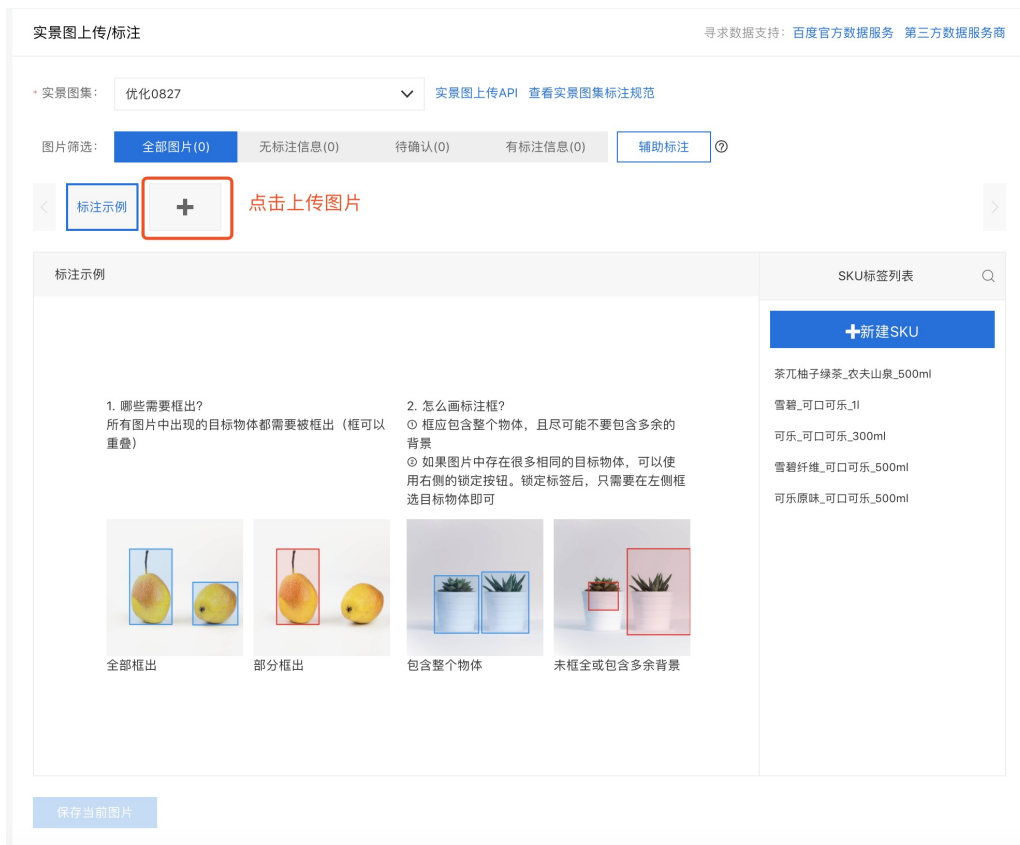
- 实景图集名称

实景图集的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等

- 选择类型

实景图集的类型，请与创建模型时选择的应用场景保持一致，上传时只上传跟选择类型相同的实景图。可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

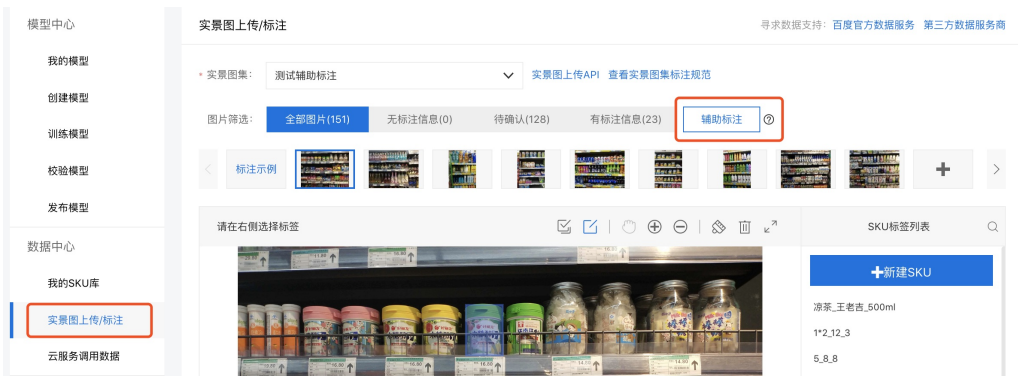
完成创建实景图集后，页面显示为如下图所示的内容



点击页面上【标注】为该实景图集上传作为训练数据的实景图，点击【标注示例】右侧的上传图片上传实景图。

上传完需要标注的图片后，EasyDL零售版的在线标注工具提供了辅助标注功能，该功能可以使用平台预置模型和用户自训练定制模型预先为未标注的图片进行预标注，来降低整体标注工作的成本，如下图所示，使用方式请参考[实景图标注文档](#)。





实景图基本要求如下：

实景图的具体采集要求，请参考[实景图数据要求文档](#)

- 实景图需要是从真实业务场景中采集来的数据
- 支持上传的图片格式为jpg，png，jpeg，bmp，大小限制为4M
- 建议图片尺寸：最长不超过4096px，最小不低于30px，长宽比3：1以内

标注基本要求如下：

实景图的具体标注要求，请参考[实景图标注规范文档](#)

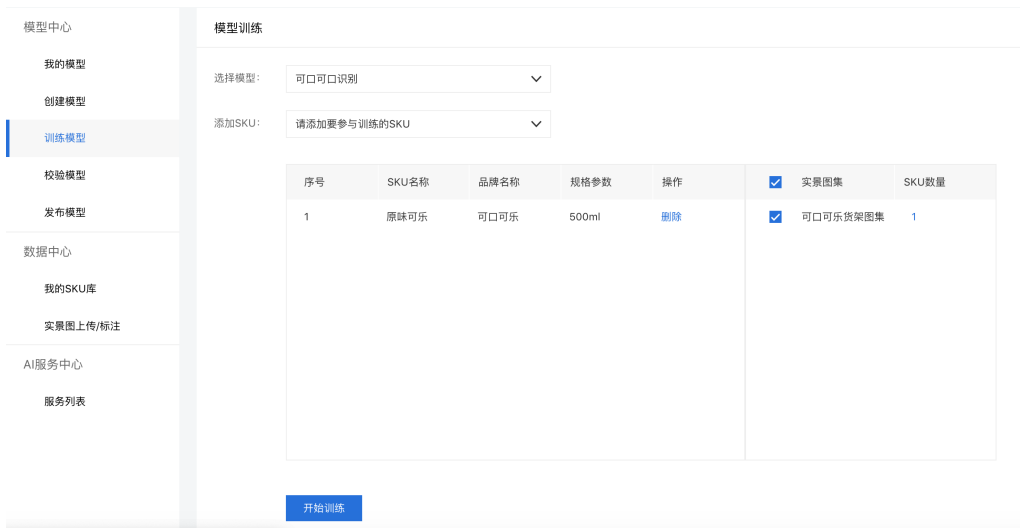
- 完整并仅仅框选要识别的SKU
- 标注框不要框选到其它SKU或是价目标签等非要识别的SKU的干扰信息
- 在实景图中出现的所有要识别的SKU必须全部标注，不能遗漏

完成所有实景图的标注后，返回到【我的SKU库】可以查看到SKU列表中【实景图数】列显示标注了该SKU的实景图的数量，如下图所示



## 步骤4.训练模型

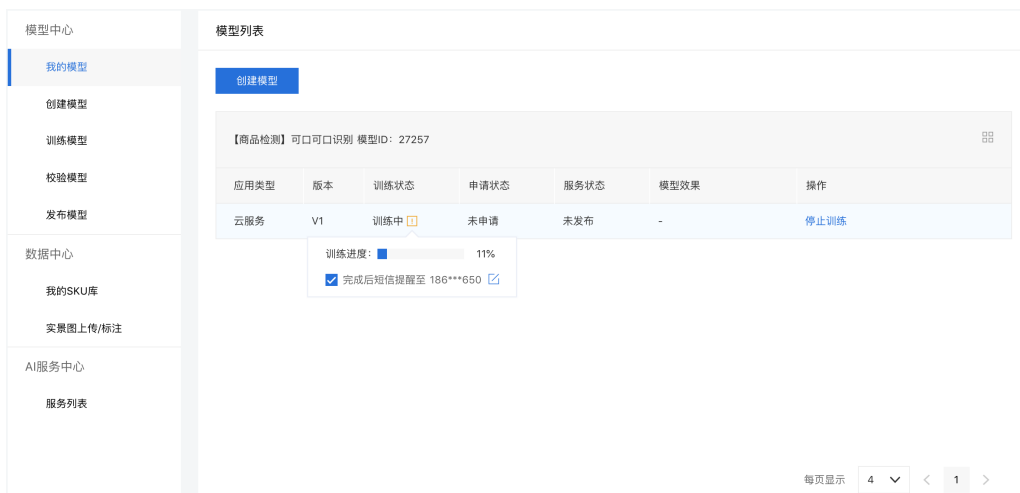
这个步骤将会介绍如何训练模型



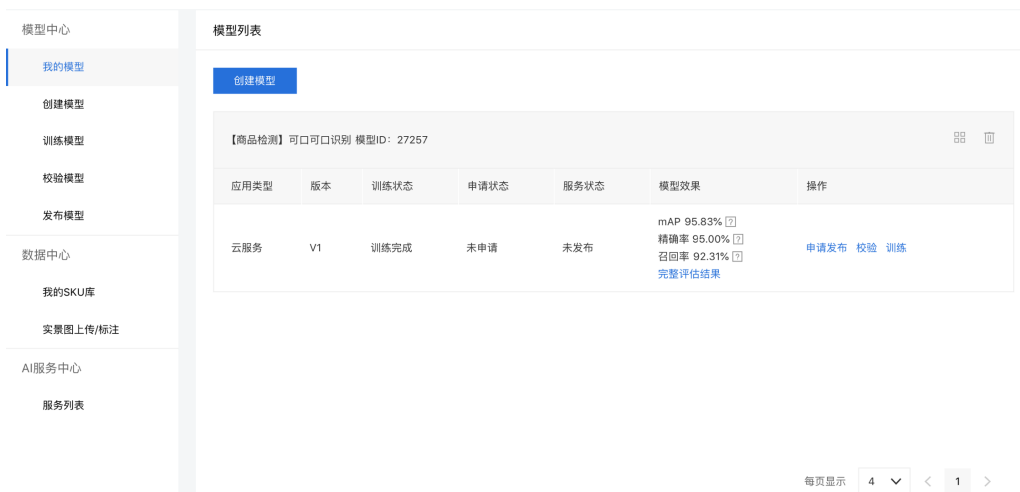
如上面图片所示，点击左侧列表中的【训练模型】，需要先后完成下面三项选择：

1. 选择要训练的模型
2. 选择需要模型支持检测的SKU，选择完成后，下方左侧会显示已添加的SKU，右侧会显示包含已添加SKU的实景图集
3. 选择要参与训练的实景图集

完成选择后，点击【开始训练】按钮页面跳转至【我的模型】页面，如下图所示，可以看到模型已进入训练状态，将鼠标移至状态"训练中"右边的小问号上，可以查看训练进度，训练进度数值只是作为参考，所以推荐打开短信通知功能，这样就第一时间知晓模型训练完成了。



训练完成后，可以点击校验和申请发布。



## 步骤5.发布模型

这个步骤将会介绍如何将训练好的模型发布为服务API

**发布模型**

- 选择模型: 饮品检测
- 选择版本: V1
- 服务名称: kelexuebijiance
- 接口地址: https://aip.baidubce.com/rpc/2.0/ai\_custom\_retail/v1/detection/kelexuebixianwei
- 其他要求: 可乐和薯条纤维检测

[提交申请](#)  
如果有私有化部署需求, 请点击此申请

**标准接口规范参考**

标准接口请求参考说明:

字段名称	必须	类型	说明
image	是	string	图像数据, base64编码, 要求base64编码后大小不超过4M, 最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式
threshold	否	number	阈值, 默认为当前模型推荐阈值(0-1之间), 具体值可以在我的模型列表-模型效果查看

标准接口响应字段说明:

字段名称	必须	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回

在模型训练好后, 点击模型列表内对应模型「操作」列中的「申请发布」, 或是在左侧导航栏点击「发布模型」可以进入发布模型页面, 如上图所示。在对应选项中选择和输入相应内容发起模型发布的申请:

#### 1. 选择模型 (必选)

选择需要发布的模型, 只能选择已经完成训练的模型

#### 2. 选择版本 (必选)

选择需要发布的模型版本, 只能选择完成训练且没有发布过的版本

#### 3. 服务名称 (必填)

为发布的服务命名, 服务名称不得多于20个字符

#### 4. 接口地址 (必填)

自定义服务的API URL, 接口地址需要多于5个字符但不能超过20个字符, 仅限英文

#### 5. 其他要求

如果有其他要求可以输入要求描述

填写完上述信息后, 点击「提交申请」完成发布模型申请。提交申请后, 模型列表内该模型的申请状态和服务状态为有以下几种情况:

申请状态	服务状态	状态描述
审核中	未发布	服务刚申请发布, 模型在审核中
审核成功	发布中	服务通过审核, 进入系统自动发布阶段
审核成功	已发布	服务发布成功
审核失败	未发布	服务未通过审核, 通常为模型训练结果mAP < 0.6, 如需申诉, 可以加入官方QQ群 (群号:1009661589) 咨询群管

提示: 第一次申请发布的模型需要人工审核, 通常4小时内完成, 如果希望加急上线, 请加入官方QQ群 (群号:1009661589) 咨询群管高优审核。非第一次申请发布的模型, 如果模型训练结果mAP>0.6, 则会自动通过审批。审批完成后, 大约需要5分钟左右自动完成发布。

## 步骤6. 使用模型API

发布成功后, 可以点击模型列表内「操作」列中的「配置服务功能」, 如下图:

**模型列表**

【商品检测】 饮品检测 模型ID: 27899 [训练](#) [历史版本](#) [删除](#)

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V1	训练完成	审核成功	已发布	mAP: 96.95% 精确率: 88.46% 召回率: 92.00% <a href="#">完整评估结果</a>	<a href="#">查看版本详情</a> <a href="#">配置服务功能</a> <a href="#">校验</a> <a href="#">体验H5</a>

点击后弹出下图所示窗口, 可以获取模型的云服务API URL, API使用方式请参考[API调用方法文档](#)。

我的模型 > 饮品检测V1的公有云API服务详情

服务名称: kelexuebijiance  
模型版本: V1  
服务状态: 已发布

接口地址: [https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/detection/kelexuebixianwei](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/detection/kelexuebixianwei)

服务功能: 已选功能不同, 单次接口调用价格不同, 详细收费方式请见 [计费文档](#)

基础功能服务

商品基本信息识别  
商品标签 (名称、品牌、规格)、编号、坐标、置信度

可选服务功能

商品陈列层数识别  
商品所在货架层数及货架总层数, 支持区分不同货架  OFF

商品陈列场景识别  
商品陈列的场景类型, 支持货架、端架、立式冰柜  OFF

商品排面占比统计  
商品陈列的排面数及占比  OFF

提示: 开启或关闭功能后, 约5分钟后生效

[立即使用](#) [返回](#)

服务功能描述:  
模型服务接口使用方法请见 [API文档](#)

服务功能	接口返回字段	内容
商品基本信息识别	name	商品标签, 包含商品名称、品牌、规格
	sku_code	商品编号
	score	识别结果的置信度
	location	商品检测框在图片上的像素坐标
商品陈列场景识别	scenes	图片中包含的陈列场景类型, 支持货架、端架和立式冰柜
	scene	每个商品所在的陈列场景
商品陈列层数识别	shelf	商品所在的货架编号, 从左往右依次递增
	layer	商品所在货架层数编号, 从上往下依次递增
	layer_count	统计各货架的总层数
	layer_top	判断货架最顶层是否拍摄完整
		综合统计结果, 包含识别

在该页面可以为模型的云服务API配置服务功能, 支持以下四项功能:

- 商品基本信息识别 (必选)  
接口支持识别商品信息 (商品名称、品牌、规格)、编号和置信度
- 商品陈列层数识别 (可选)  
接口支持识别商品陈列所在货架层数, 货架总层数以及商品的陈列顺序, 货架类型支持: 货架、端架和立式冰柜内货架
- 商品陈列场景识别 (可选)  
接口支持识别商品陈列的场景, 场景类型支持: 普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、地堆、割箱、地龙、小端架、吧台
- 商品排面占比统计 (可选)  
接口支持统计商品排面数/占比、未识别商品数、总空位数、每货架层的空位数及货架利用率

接口单次调用的费用, 根据开启的功能不同而不同, 详情可见[购买指南文档](#)。

可在页面随时开启和关闭可选的功能, 变更功能后约5分钟生效, 生效后单次调用费用按变更后的功能计费, 接口将返回变更后的功能字段, 详情可见[API调用方法文档](#)。

## 模型创建

### 进入模型训练页面

在[EasyDL零售版产品主页](#)点击【开始训练】按钮进入到[模型训练页](#), 下面会出现两种情况:

- 第一种, 如果您没有登录百度智能云, 则会跳转到百度智能云登录页面, 没有百度账户的客户请先[注册百度账户](#)。登录后, 会跳转到[模型概览页](#), 点击【商品检测】卡片上的【立即定制】按钮, 会跳转模型训练页面的创建模型页。
- 第二种, 如果您已登录, 会直接进入【我的模型】页, 该页面能够管理已经创建的模型, 点击左侧列表中的【创建模型】进入创建模型页面。

### 创建模型

进入创建模型页面后你会看到如下图中展示的内容:

模型中心

模型列表 > 创建模型

模型类别: 商品检测

模型名称:

模型归属:  公司  个人

应用场景:  ▼

不同应用场景对应不同训练算法, 请根据真实应用场景选择

邮箱地址:

联系方式:  ②

功能描述:

0/500

需要填写的项目如下：

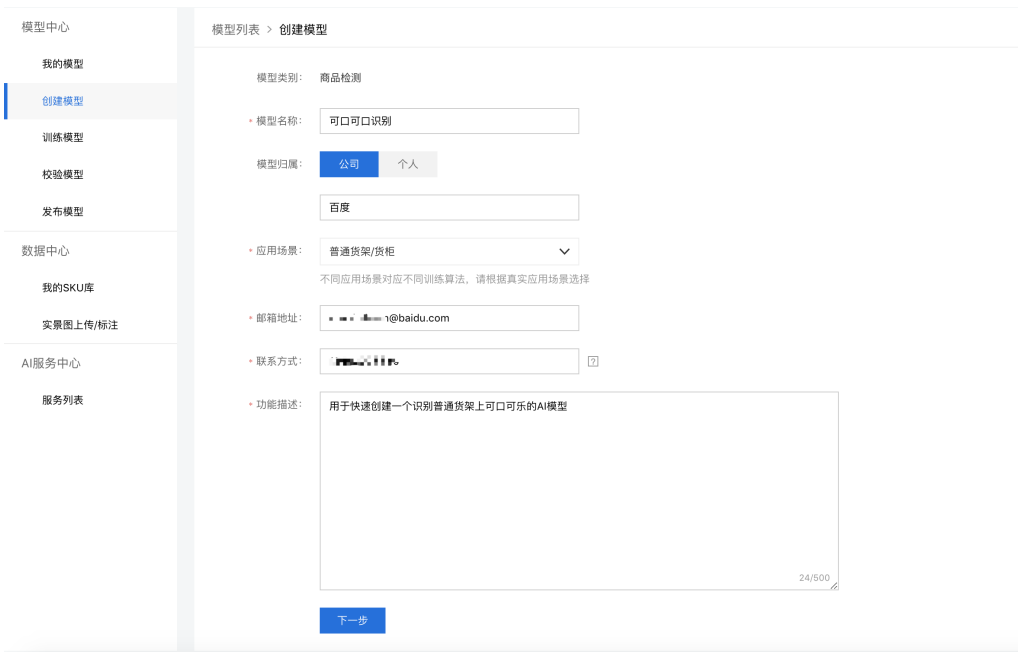
- 模型名称  
模型的名称
- 模型归属  
模型是属于公司的，还是属于个人的，如果是前者，请填写公司名称
- 应用场景

提示：选择模型将要应用的场景，请根据真实应用场景选择，选择的场景将会关联后端数据增强算法，若不确定，请选择“其他”

可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

- 邮箱地址  
用于联系到您的邮箱地址
- 联系方式  
有效的联系方式将有助于后续模型上线的人工快速审核，以及更快的百度官方支持，推荐填写个人手机号码
- 功能描述  
描述改模型将要应到的业务场景，详细的描述，在获取官方支持时，能帮助我们为您提供准确的使用建议

像下图展示的一样完成所有填写项后点击【下一步】按钮完成模型创建，创建完成后会跳转到【我的模型】页面。



## 数据准备

### SKU创建

#### 在Web页面创建SKU

#### 单个创建SKU

在**模型训练页面**，点击左侧列表中的【我的SKU】进入**SKU管理页面**，点击【创建SKU】按钮进入**创建SKU页面**。



您会看到如下图展示的内容：



提示：SKU识别结果中，SKU的名字是以“SKU名称\_品牌名称\_规格参数”的形式返回的，在填写SKU名称、品牌名称和规格参数时，请避免这三项内容重复。

需要填写的项目如下：

- **SKU名称**  
SKU的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等
- **品牌名称**  
SKU的品牌名称，如可口可乐，乐事，伊利等
- **规格参数**  
SKU的规格，如330ml，500g，20片等
- **商品品类**  
可选择的有饮品、药品、保健品、零食、香烟、调味品、日用品和其他
- **包装类型**  
可选择的有瓶装、罐装、袋装、盒装和其他
- **商品编号**  
如果您自身的业务系统中有现成SKU对应的商品编码，比如商品条形码，可以填在该填写框中，之后模型接口将支持返回该内容，用于您快速匹配SKU
- **SKU单品图**  
SKU的单品图将用于商品增强合成，拍摄角度和上传张数基本原则是覆盖实际检测场景可能出现的角度，请参考「SKU单品图数据要求」文档中进行单品图采集。如果不上传，将会降低模型的识别效果，可以点击页面上的【示例图片】查看SKU单品图样张。

完成填写和上传SKU单品图上传后，页面内容显示如下图所示

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

数据中心

我的SKU库

实景图上传/标注

AI服务中心

服务列表

我的SKU库 > 创建SKU

创建的SKU将以“SKU名称\_品牌名称\_规格参数”的形式作为识别时返回的结果。命名时请避免重复

\* 商品名称: 原味可乐

\* 品牌名称: 可口可乐

\* 规格参数: 500ml

\* 商品品类: 饮品

\* 包装类型: 瓶装

商品编号: 用于匹配商品系统中的编号。若无可不填

SKU单品图: [上传图片](#) 用于有效提高模型效果 [示例图片](#)

您已上传12张SKU单品图

[创建SKU](#) [返回](#)

点击【创建SKU】按钮完成创建，点击后回到【我的SKU库】，SKU列表中的SKU图数需要大约5秒的时间进行计算，刷新页面即可显示SKU单品图片数。

### 批量创建SKU

在模型训练页面，点击左侧列表中的【我的SKU】进入SKU管理页面，点击【本地批量上传】按钮进入批量上传页面。

我的SKU库

[创建SKU](#) [本地批量上传](#) [API上传](#)

SKU名称	品牌名称	规格参数	SKU单品图数	被标注次数	状态	操作
番茄味薯片-袋装2	乐事	75g	0	1	正常	<a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a>
可乐-瓶装1	可口可乐	300ml	0	3	新建	<a href="#">查看</a> <a href="#">编辑</a> <a href="#">删除</a>

您会看到如下图展示的内容：

我的SKU库 > 本地批量上传

选择本地文件：[上传文件](#)

文件要求说明：[下载示例文件](#)

1. 支持xls和xlsx格式的Excel文件，以及CSV文件（分隔符为英文逗号）
2. 文件编码支持：UTF-8，可参考 [编码查看和修改方法文档](#)
3. 文件必须包含以下表头，各列的内容规范如下：

表格名称	内容规范
sku_name	不得多于30个字符，支持中文、英文、数字和- & ' + * ( ) % /
brand_name	不得多于20个字符，只支持中文、英文、数字和- & ' + * ( ) % /
specs	不得多于20个字符，只支持中文、英文、数字和- & ' + * ( ) % /
product_code	不得多于40个字符，支持英文、数字和- & ' + * ( ) % /
category	数字代表商品类别：1为饮品，2为药品，3为保健品，4为零食，5为香烟，6为调味品，7为日用品，99为其它
package_type	数字代表商品包装类型：1为瓶装，2为罐装，3为袋装，4为盒装，5为桶装，99为其它

[确认](#) [取消](#)

按照要求上传SKU标签信息文件即可，[建议下载示例文件修改后上传](#)，以保证编码为UTF-8。如果是自建的文件，可以参考[编码查看和修改方法文档](#)进行编码修改。

提示：将文档上传前，请确认UTF-8编码的文档内的文字没有乱码。

### 使用API管理SKU数据

SKU的创建、删除、查询和上传SKU单品图均可以通过调用API实现，API使用方法请参考文档[SKU管理API](#)。

### 🔗 文档编码查看和修改方法

本文档介绍如何查看一个文档的编码以及修改编码的方法。

#### 使用示例文件

进入[SKU本地批量上传页面](#)，点击页面上的[下载示例文件](#)，下载后在源文件中删除示例SKU后，再添加进需要创建的SKU信息。

我的SKU库 > 本地批量上传

选择本地文件：[上传文件](#)

文件要求说明：[下载示例文件](#)

1. 支持xls和xlsx格式的Excel文件，以及CSV文件（分隔符为英文逗号）
2. 文件编码支持：UTF-8，可参考 [编码查看和修改方法文档](#)
3. 文件必须包含以下表头，各列的内容规范如下：

#### 查看和修改CSV文件的编码

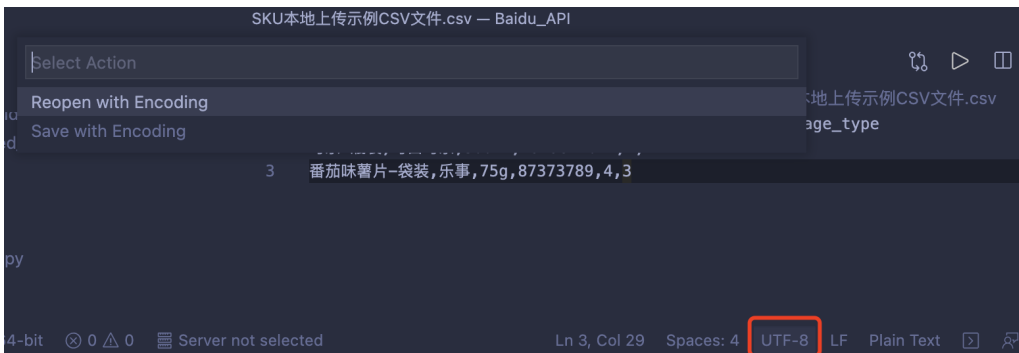
提示：将文档上传前，请确认UTF-8编码的文档内的文字没有乱码。

下载[Visual Studio Code](#)，在Visual Studio Code打开csv文档，右下角可看到文件的编码格式，如下图所示：





如果不是UTF-8，点击显示编码格式的区域，输入框会弹出两个选项，如下图所示：



点击「Save with Encoding」，会弹出编码选择，选择「UTF-8」后即可。



#### 查看和修改Excel文件的编码

**提示：**将文档上传前，请确认UTF-8编码的文档内的文字没有乱码。

由于Excel的默认编码和操作系统以及软件版本有关，建议使用示例文件修改后上传，如果没有使用，又无法正常上传至平台，可以将源文件另存为CSV文件后，参考修改CSV文件的编码的方法转换成UTF-8的CSV文件进行上传。

#### SKU标签组

##### 创建标签组

##### 直接创建标签组

在[模型训练页面](#)，点击左侧列表中的【我的SKU】进入[SKU管理页面](#)，点击【当前标签组】下拉列表中【+创建分组】创建空白标签组，创建成功后，可在【当前标签组】下拉列表中显示新建的标签组。



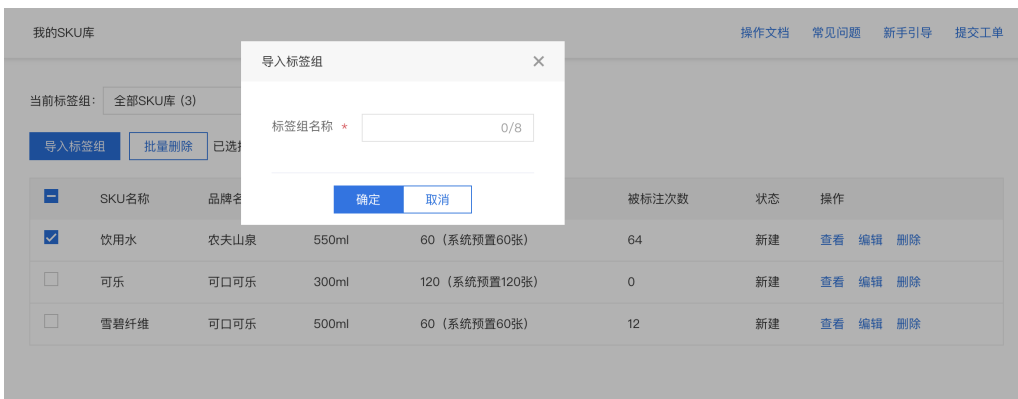
提示：此时创建的标签组内没有SKU记录。

### 导入SKU时创建标签组

点击【当前标签组】下拉列表中【全部SKU库】，在SKU列表中左侧勾选SKU，再点击【导入标签组】。



如果您之前还没有创建过标签组，您会看到如下图所示的内容。填写标签组名称，点击【确定】，创建新的标签组并自动导入选中SKU记录到该标签组，成功创建的标签组可以在【当前标签组】下拉列表中显示。



如果您之前已经创建过标签组，您会看到如下图所示的内容：



点击【创建新分组】状态栏为【ON】状态，再填写标签组名称，点击【确定】，创建新的标签组并自动导入选中SKU记录到该标签组，成功创建的标签组可以在【当前标签组】下拉列表中显示。



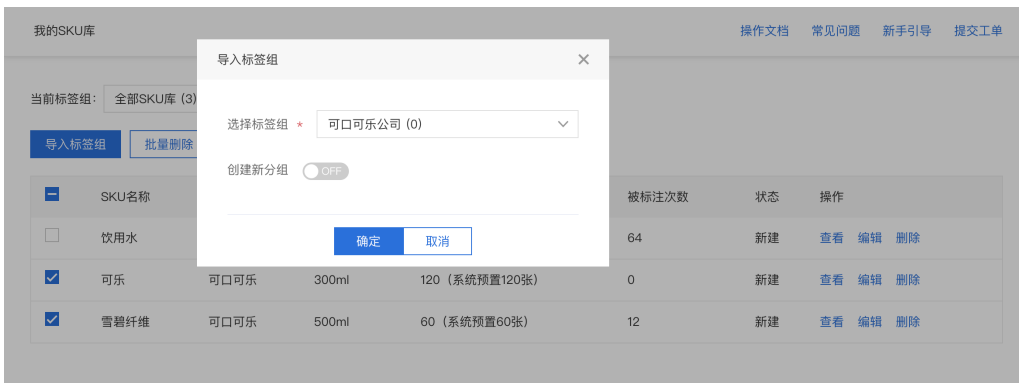
## 编辑标签组

### 导入SKU

点击【当前标签组】下拉列表中【全部SKU库】，在SKU列表中左侧勾选SKU，再点击【导入标签组】。



选择需要导入SKU的标签组，点击【确定】，完成导入，此时点击【当前标签组】下拉列表，选中想要查看的标签组，便可以浏览该标签组中的SKU。



## 移除SKU

### 移除单个SKU

点击【当前标签组】下拉列表，选中一个标签组，在SKU列表中「操作」列点击【移除】。



确认后即可将该SKU从改标签组中移除，移除SKU并不会删除SKU库中的SKU，该SKU仍然会被保留在“全部SKU库”中。



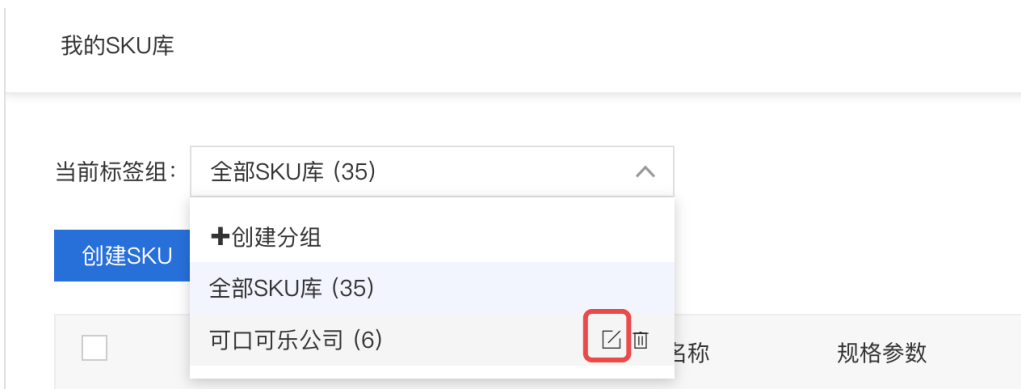
### 批量移除SKU

同样的，您也可以选中该标签组中一个或多个SKU的，再点击批量移除，实现对批量SKU从该标签组中移除的作用。移除SKU并不会删除SKU库中的SKU，移除的SKU仍然会被保留在“全部SKU库”中。

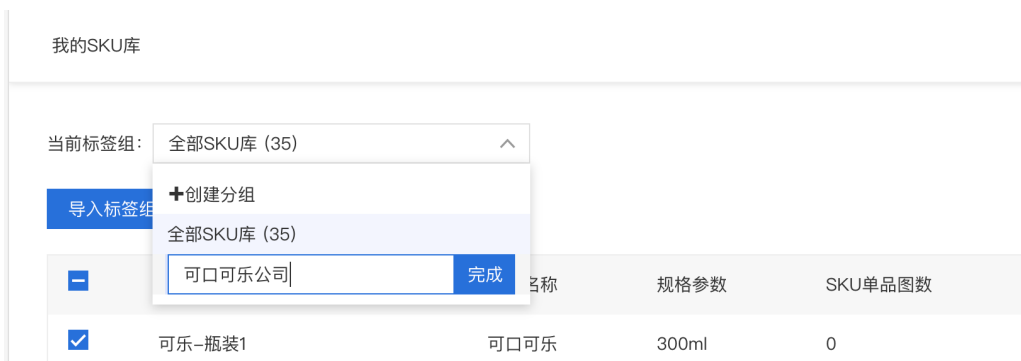


### 重名标签组

点击【当前标签组】下拉列表，点击【编辑】按钮修改标签组的名称。

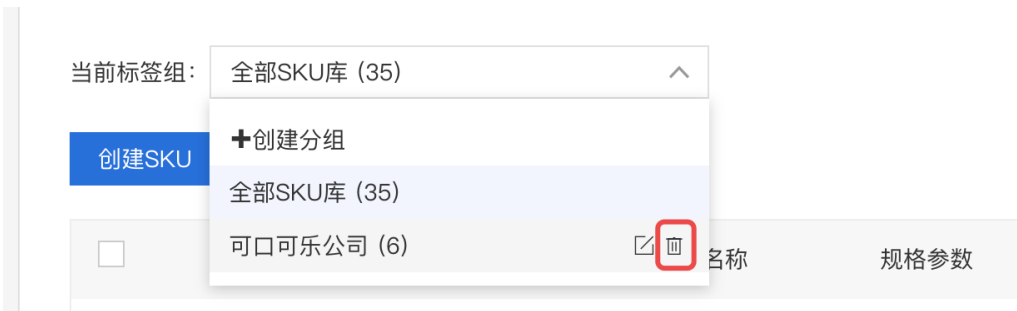


点击【完成】即可完成编辑，如下图所示。

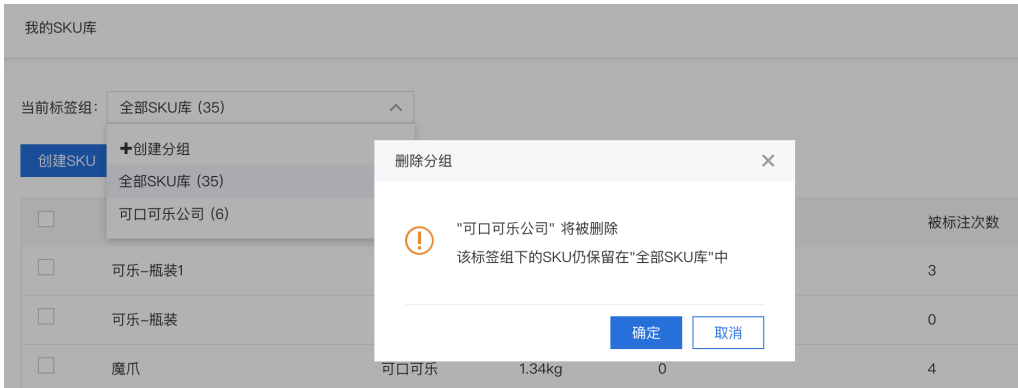


### 删除标签组

点击【当前标签组】下拉列表，点击【删除】按钮删除此标签组。



确认后删除该标签组，删除SKU标签组后，该标签组下的SKU仍然会保留在“全部SKU库”中。



## SKU单品图数据要求

### 简介

训练一个定制商品检测模型需要准备两类数据：SKU单品图片和实景图图片。本文档将详细介绍训练一个定制商品检测模型的数据要求，如规格、大小、尺寸等，并提供相应的图片样例。另外，可以参考[实景图标注规范文档](#)文档内容，了解各个业务场景的数据如何正确标注。

点击下载[数据采集与标注规范长图](#)，一张图看懂如何采集和标注数据，让您不走弯路，获得一个高精度的商品检测模型。

SKU单品图指的是单个商品的图片，**不是模型训练必须的数据**，SKU单品图的作用是用来合成实景图，连同手工标注的实景图一起用于训练，降低实景图即训练数据采集和标注成本。

当每个SKU的实景图大于20张时，可以先不上传SKU单品图进行训练，后续提升模型以补充实景图为主，如果无法提供足量的实景图数据，可以通过上传SKU单品图来提升模型效果。

为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分的成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且单品图无需进行任何标注，即可让模型学习到这个SKU各个角度的样子。

平台上预置了近千个SKU，每个预置的SKU已匹配了50张左右各角度的单品图，绝大多数情况下无需再自行上传单品图，可根据训练结果补齐识别效果不好角度的单品图。

### 格式要求

图片的格式为：jpg、jpeg、png、bmp，图片大小不超过4M。

### 分辨率大小要求

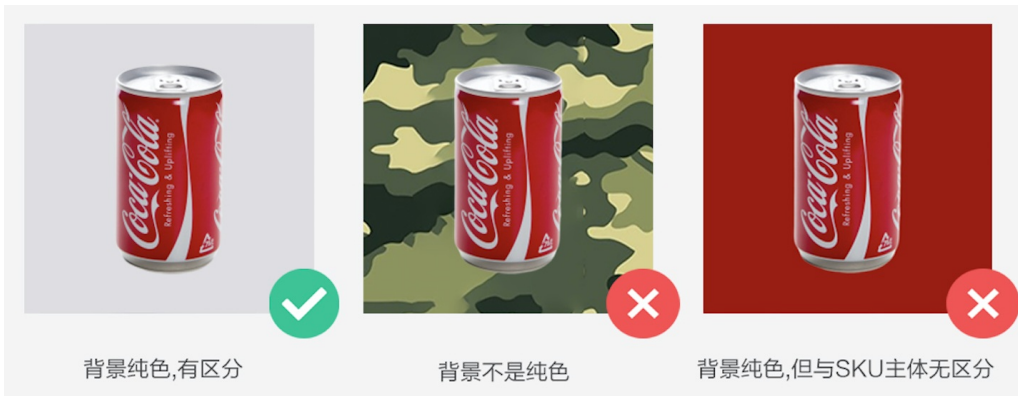
图片分辨率大小建议达到1920×1080以上，拍摄相机像素200W以上，保证图片上的SKU清晰不模糊。

### 图片内容要求

项目	要求
图中SKU数量	图片上仅可出现一个SKU
背景	纯色，且背景颜色与SKU主体有区分度
角度	覆盖到实际检测场景中SKU可能出现的差异性较大的所有角度
光照	覆盖到实际检测场景中SKU可能出现的差异性较大的光照条件，比如灯光的颜色

下面举两个例子：

1. 当要检测的SKU是罐装可口可乐时，背景不可以是非纯色或是纯红色，如下图所示：



2. 当业务场景是货架陈列审核，且货架上的商品无确定的展示面时，单品图需要覆盖到可能在货架上出现的所有差异性较大的角度，考虑到拍摄角度，一个SKU需要覆盖到水平视角、俯视视角和仰视视角，如下图所示：



#### 各场景单品图推荐拍摄角度和上传图片数

拍摄角度和上传张数基本原则是覆盖实际检测场景可能出现的角度，请根据实际业务场景的情况灵活调整单品图的拍摄角度和上传图片数。

场景	推荐拍摄角度	推荐上传图片数
普通货架/货柜审核	水平视角、俯视视角和仰视视角	10 张
无人零售货柜	俯视视角	10 张
智能结算台	水平视角、俯视视角，可根据实际情况增加仰视视角	每个角度各10 张
地堆商品审核	可尽量在实景图覆盖需要检测的角度	无需上传单品图

#### 单品图片样例

以一个货架上瓶装商品为例，如下图所示：



### 使用API管理线上SKU数据

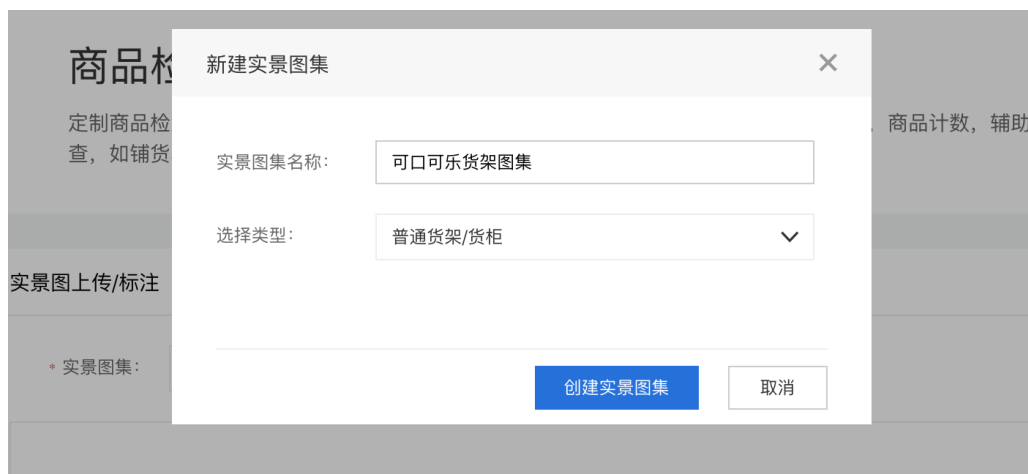
SKU的创建、删除、查询和上传SKU单品图均可以通过调用API实现，API使用方法请参考文档[SKU管理API](#)。

### 🔗 实景图上传

#### 使用平台在线上传图片

SKU在货架上的实景图是模型训练需要用到的训练数据，需要客户从真实的业务场景中采集，这些图片在被正确标注中，可以用于训练成模型。

在完成[SKU创建](#)后，可在[模型训练页面](#)左侧列表中点击【实景图上传/标注】进入[上传和标注页面](#)，在上传前请在实景图集选择栏内创建实景图集，如下图所示



需要填写的项目如下：

- 实景图集名称

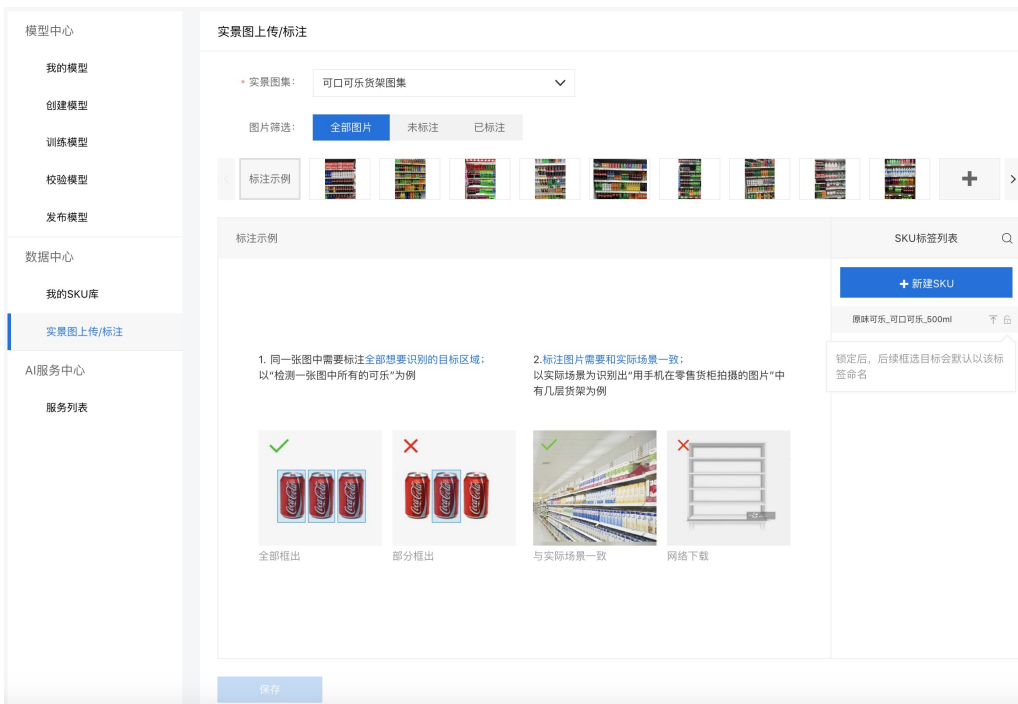
实景图集的名称，可适当填入SKU细节，例如：原味可乐，番茄味薯片，奥运版纯牛奶等

- 选择类型

实景图集的类型，请与创建模型时选择的应用场景保持一致，上传时只上传跟选择类型相同的实景图。可选项为普通货架/货柜、智能结算台、无人零售柜、地堆商品和其他

完成创建实景图集后，页面显示为如下图所示的内容





点击页面上【标注】为该实景图集上传作为训练数据的实景图，点击【标注示例】右侧的加号上传实景图。

实景图基本要求如下：

实景图的详细采集要求，请参考[实景图数据要求文档](#)

- 实景图图片需要是从真实业务场景中采集来的数据
- 支持上传的图片格式为jpg，png，jpeg，bmp，大小限制为4M
- 建议图片尺寸：最长不超过4096px，最小不低于30px，长宽比3：1以内

标注基本要求如下：

标注工具的使用方式，请参考[实景图标注文档](#)，实景图的具体标注要求，请参考[实景图标注规范文档](#)

- 完整并仅仅框选要识别的SKU
- 标注框不要框选到其它SKU或是价目标签等非要识别的SKU的干扰信息
- 在实景图中出现的所有要识别的SKU必须全部标注，不能遗漏

完成所有实景图的标注后，返回到【我的SKU库】可以查看到SKU列表中【实景图数】列显示标注了该SKU的实景图的数量，如下图所示



使用API线下上传图片

实景图也可以使用线下标注工具标注好后，通过API上传至EasyDL零售版训练平台，也可以通过API创建和删除实景图集，API使用方法请参考文档[实景图管理API](#)。



## 实景图数据要求

### 简介

训练一个定制商品检测模型需要准备两类数据：SKU单品图片和实景图片。本文档将详细介绍训练一个定制商品检测模型的数据要求，如规格、大小、尺寸等，并提供相应的图片样例。另外，可以参考[实景图标注规范文档](#)内容，了解各个业务场景的实景图片如何正确标注。

点击下载[数据采集与标注规范长图](#)，一张图看懂如何采集和标注数据，让您不走弯路，获得一个高精度的商品检测模型。

实景图指的是从业务场景中采集的图片，这些图片需要手动标注，只有标注的图片才会被用于训练，下面给出一些场景中的图片样例：

### 格式要求

图片的格式为：jpg、jpeg、png、bmp，图片大小不超过4M。

### 图片内容要求

上传标注的图片内容需要跟实际业务检测图片来源一致，比如货架上商品陈列审核业务，上传标注的图片是业务员巡店时拍摄的图片；无人货柜业务，上传标注的图片是货柜里摄像头采集的实际投放时摆放了商品的图片；智能结算台业务，上传标注的图片是结算台日常结算时拍摄的图片。

### 采集设备要求

采集设备推荐与实际业务中拍摄图片的设备一致。比如，智能结算台业务场景，采集设备推荐为结算台；无人货柜业务场景，采集设备推荐为货柜；普通货架/货柜审核业务场景，采集设备推荐为手机。

### 分辨率大小要求

实景图中能够清晰看清每一个要识别的SKU，各场景的推荐图片分辨率如下：

场景	推荐图片分辨率
普通货架/货柜审核	1920×1440以上
地堆商品审核	1920×1440以上
无人零售货柜	1280×720以上
智能结算台	1280×720以上

以普通货架/货柜审核场景为例，如下图所示：



### 拍摄角度要求

在保证清晰度的前提下，实景图采集时的拍摄角度建议与实际检测时保持一致。普通货架/货柜审核场景需要注意，图片尽量从正面拍摄，角度可以少量倾斜，但不要倾斜过大，如下图所示：



俯角拍摄,清晰可辨



水平倾斜,清晰可辨



拍摄角度过大,以至增加商品特征识别难度



### 推荐上传标注图片数

在第一次训练时,建议每个SKU至少有20张实景图,上传的实景图,只有标注过的图片会被训练,所有训练的图片中,系统会随机抽取70%作为训练集,剩余的30%作为测试集,如果标注的训练数据不足,可能会导致某个SKU的精确度远低于其它SKU,或是训练结果出现mAP、精确率、召回率全都为0的情况。

第一次训练后,通过调取服务接口测试模型效果,根据测试结果,不断补足识别效果达不到的需求的SKU实景图,这个过程可参考[模型优化方法文档](#)和使用[模型优化工具](#)处理识别效果不佳的实景图图片。

### 实景图图片样例

#### 普通货架/货柜陈列审核场景





地堆商品陈列审核场景



无人货柜场景





智能结算台场景



#### 使用API管理实景图数据

实景图也可以使用线下标注工具标注好后，通过API上传至EasyDL零售版训练平台，也可以通过API创建和删除实景图集，API使用方法请参考文档[实景图管理API](#)。

#### 🔗 SKU管理API

##### 简介

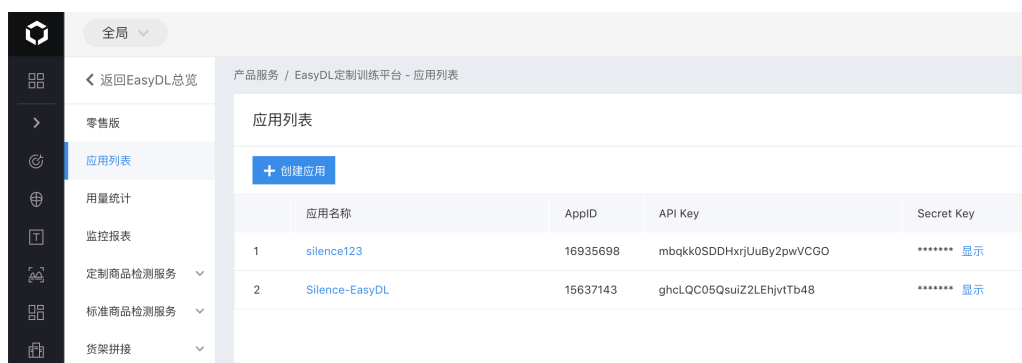
本文档主要说EasyDL零售版的定制商品检测服务中的SKU管理API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择人工智能服务
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

- 加入EasyDL零售版官方QQ群（群号:1009661589）联系群管

## 接口鉴权

- 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



- 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权认证机制文档](#)，使用API Key(AK)和Secret Key(SK)获取access\_token

## SKU管理API概览

SKU管理API包含以下API：

接口名称	HTTP方法	API URL	说明
SKU创建	POST	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/create">https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/create</a>	用于创建SKU
SKU更新	POST	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/update">https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/update</a>	用于更新SKU
SKU列表	POST	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/list">https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/list</a>	用于列出所有SKU，可获得SKU的所有信息
SKU删除	POST	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/delete">https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/delete</a>	用于删除指定SKU
SKU数据添加API	POST	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/addentity">https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/addentity</a>	用于为指定SKU上传单品图片

## SKU创建API

### 接口描述

该接口可用于创建SKU

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/create>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

### 请求参数

字段	必选	类型	说明
sku_name	是	string	SKU名称，长度限制100个字符，支持中文、英文、数字和- & ' . + * ( ) % / #
brand_name	是	string	品牌名称，长度限制100个字符，支持中文、英文、数字和- & ' . + * ( ) % / #
specs	是	string	规格参数，长度限制100个字符，支持中文、英文、数字和- & ' . + * ( ) % / #
category	是	string	商品品类：饮品 drink，药品 medicine，保健品 healthcare products，零食 snacks，香烟 cigarette，调味品 condiment，日用品 daily necessities，其它 other
package_type	是	string	包装类型：瓶装 bottled，罐装 canned，袋装 bagged，盒装 boxed，桶装 barrel，其它 other
product_code	否	string	商品编号，长度限制100个字符，支持英文、数字和- & ' . + * ( ) % / #
package_image	否	string	图片数据，将图片转化为base64编码上传，要求图片大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式，注意请去掉头部。

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位
sku_id	是	int	创建的SKU ID

### SKU更新API

#### 接口描述

该接口可用于更新SKU

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/update

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

#### 请求参数

字段	必选	类型	说明
sku_id	是	int	SKU的ID, SKU完成创建时刻获取, 或可通过SKU列表接口查询
sku_name	是	string	SKU名称, 长度限制100个字符, 支持中文、英文、数字和- & ' . + * ( ) % / #
brand_name	是	string	品牌名称, 长度限制100个字符, 支持中文、英文、数字和- & ' . + * ( ) % / #
specs	是	string	规格参数, 长度限制100个字符, 支持中文、英文、数字和- & ' . + * ( ) % / #
category	是	string	商品品类: 饮品 drink, 药品 medicine, 保健品 healthcare products, 零食 snacks, 香烟 cigarette, 调味品 condiment, 日用品 daily necessities, 其它 other
package_type	是	string	包装类型: 瓶装 bottled, 罐装 canned, 袋装 bagged, 盒装 boxed, 桶装 barrel, 其它 other
product_code	否	string	商品编号, 长度限制100个字符, 支持英文、数字和- & ' . + * ( ) % / #
package_image_url	否	string	和package_image二选一, 当package_image字段存在时, 该字段输入失效, 以package_image字段为准。图片大小不超过4M, 最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式。通过URL上传SKU包装图时, 请确保图片链接有外网访问权限, 否则图片将会上传失败, 若需要补充SKU包装图, 可使用SKU更新API。
package_image	否	string	和package_image二选一, 当package_image字段存在时, 以package_image字段为准。图片数据, 将图片转化为base64编码上传, 要求图片大小不超过4M, 最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式, 注意请去掉头部。

## 返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id, 用于问题定位
sku_id	是	int	创建的SKU ID

## SKU列表API

## 接口描述

该接口可用于获取已创建的SKU列表

## 请求说明

## 请求示例

HTTP 方法: POST

请求URL: <https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/list>

URL参数:

参数	值
access_token	通过API Key和Secret Key获取的access_token, 参考 <a href="#">鉴权认证机制文档</a>

Header如下:

参数	值
Content-Type	application/json

## 请求参数

字段	必选	类型	说明
start	是	int	起始位置
num	是	int	结果数量, 最大数量为100

## 返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位
total_num	是	int	返回结果数
results	是	array	返回结果
+sku_id	是	int	SKU ID
+sku_name	是	string	SKU名称
+brand_name	是	string	品牌名称
+specs	是	string	规格参数
+category	是	string	商品品类
+package_type	是	string	包装类型
+product_code	否	string	商品编号
+status	是	string	SKU状态：新建 new，上传SKU单品图片中 uploading，错误 error，正常 normal，训练中 training
+entity_count	是	int	SKU单品图数量

## SKU删除API

## 接口描述

该接口可用于删除SKU

## 请求说明

## 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/delete

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

## 请求参数

字段	必选	类型	说明
sku_id	是	string	SKU的ID，SKU完成创建时刻获取，或可通过SKU列表接口查询

## 返回说明

## 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位

## SKU数据添加API

## 接口描述

该接口可用于为SKU上传单品图片



### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/retail/sku/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token， <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

### 请求参数

字段	必选	类型	说明
sku_id	是	int	SKU ID
entity_content	是	string	SKU单品图的base64编码

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	无效的参数xx，请检查相关参数
406002	dataset not exist	实景图集不存在
406003	dataset already exists	实景图集已存在
406004	dataset can not be modified temporarily	实景图集暂时不能被修改
406006	no permission to modify the dataset	没有修改实景图集的权限
406008	[xx] quota exceeded	xx配额超限
406009	sku does not exist	SKU不存在
406010	sku already exists	SKU已存在
406011	sku cannot be modified temporarily	SKU暂时不能被修改

## 🔗 实景图管理API

### 简介

本文档主要说EasyDL零售版的定制商品检测服务中的实景图管理API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择**人工智能服务**
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL零售版官方QQ群（群号:1009661589）联系群管

### 接口鉴权

1. 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权认证机制文档](#)，使用API Key(AK)和Secret Key(SK)获取access\_token

## 实景图集管理API

实景图集管理API包含以下API：

接口名称	HTTP方法	请求Body	API URL	说明
实景图集创建	POST	JSON	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/create">https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/create</a>	用于创建实景图集
实景图集列表	POST	JSON	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/list">https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/list</a>	用于列出所有实景图集，可获得实景图集的所有信息
实景图集删除	POST	JSON	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/delete">https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/delete</a>	用于删除指定实景图集
实景数据添加API	POST	JSON	<a href="https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/addentity">https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/addentity</a>	用于为指定实景图集上传实景图片

## 实景图集创建API

### 接口描述

该接口可用于创建实景图集

### 请求说明

### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/create

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

### 请求参数

字段	必选	类型	说明
type	是	string	实景图集类型，SKU_DETECTION
dataset_name	是	string	实景图集名称，长度不超过20个utf-8字符
scene	是	string	应用场景，不同场景对应内容为：普通货架/货柜 general shelf/container，智能结算台 smart self-checkout，无人零售柜 smart vending machine，地堆商品 type genus，其它 other

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位
dataset_id	是	int	创建的实景图集ID

#### 实景图集列表API

#### 接口描述

该接口可用于获取已创建的实景图集列表

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/list

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

#### 请求参数

字段	必选	类型	说明
start	是	int	起始位置
num	是	int	结果数量
type	是	string	实景图集类型，SKU_DETECTION

#### 返回说明

#### 返回参数

字段	必选	类型	说明
total_num	是	int	返回结果数
results	是	array	返回结果
+dataset_id	是	int	实景图集id
+dataset_name	是	string	实景图集名称
+type	是	string	实景图集类型
+status	是	string	实景图集状态：新建 new，上传实景图片中 uploading，错误 error，正常 normal，训练中 training
+special_status	是	string	特殊状态，商品检测模型值为空
+scene	是	string	实景图集场景

## 实景数据添加API

### 接口描述

该接口可用于为实景图集上传标注好的图片

### 请求说明

### 请求示例

HTTP 方法：**POST**

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/addentity>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

### 请求参数

字段	必选	类型	说明
dataset_id	是	int	实景图集id
type	是	string	实景图集类型，SKU_DETECTION
entity_content	是	string	实景图片的base64编码
entity_name	是	string	实景图片名称
appendLabel	否	boolean	确定添加标签的行为：追加(true)、替换(false)。默认为追加(true)
labels	否	array	SKU标签数据，如果不传该参数，则上传的为无标注信息的图片
+label_name	否	string	SKU标签名称，请先参考 <a href="#">SKU创建文档</a> 完成SKU创建，格式为：SKU名称_品牌名称_规格参数，例如：雪碧_可口可乐_500ml
+left	否	int	标注框左上角到图片左边界的距离(像素)
+top	否	int	标注框左上角到图片上边界的距离(像素)
+width	否	int	标注框的宽度(像素)
+height	否	int	标注框的高度(像素)

### 返回说明

### 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位

### 实景图集删除API

#### 接口描述

该接口可用于删除实景图集

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/retail/dataset/delete>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

#### 请求参数

字段	必选	类型	说明
dataset_id	是	string	实景图集ID
type	是	string	实景图集类型，SKU_DETECTION

#### 返回说明

#### 返回参数

字段	必选	类型	说明
log_id	是	int	唯一的log id，用于问题定位

#### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
406000	internal server error	服务错误
406001	param[xx] invalid	无效的参数xx，请检查相关参数
406002	dataset not exist	实景图集不存在
406003	dataset already exists	实景图集已存在
406004	dataset can not be modified temporarily	实景图集暂时不能被修改
406006	no permission to modify the dataset	没有修改实景图集的权限
406008	[xx] quota exceeded	xx配额超限
406009	sku does not exist	SKU不存在
406010	sku already exists	SKU已存在
406011	sku cannot be modified temporarily	SKU暂时不能被修改

## 🔗 数据相关常见问题

### 为什么建议每个SKU至少出现在20张实景图中？

上传的实景图，只有标注过的图片会被训练，所有训练的图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，如果标注的训练数据不足，可能会导致某个SKU的精确度远低于其它SKU，或是训练结果出现mAP、精确率、召回率全都为0的情况。

### SKU单品图是用来做什么的？

SKU单品图用来降低实景图即训练数据采集和标注成本的，为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分的成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且单品图无需进行任何标注，即可让模型学习到这个SKU各个角度的样子。由百度提供的SKU预置了50张左右的单品图，绝大多数情况下无需再自行上传单品图。

## SKU单品图需不需要标注？

SKU单品图不要标注，只需要参考「SKU单品图数据要求」文档采集并上传至相应的SKU即可。

## SKU单品图和实景图分别是怎样的图片？

SKU单品图是单个商品的摆拍图，要求背景为纯色；实景图是商品在真实业务场景里的图片，比如商品在超市货架上的图片。具体两类图片的数据要求，请参考「SKU单品图数据要求」和「实景图数据要求」文档。

## 每个账号允许创建多少个SKU？

每个账号默认允许创建的SKU数量为50个，如果需要增加SKU数量，请加入官方QQ群（群号:1009661589）咨询解决。

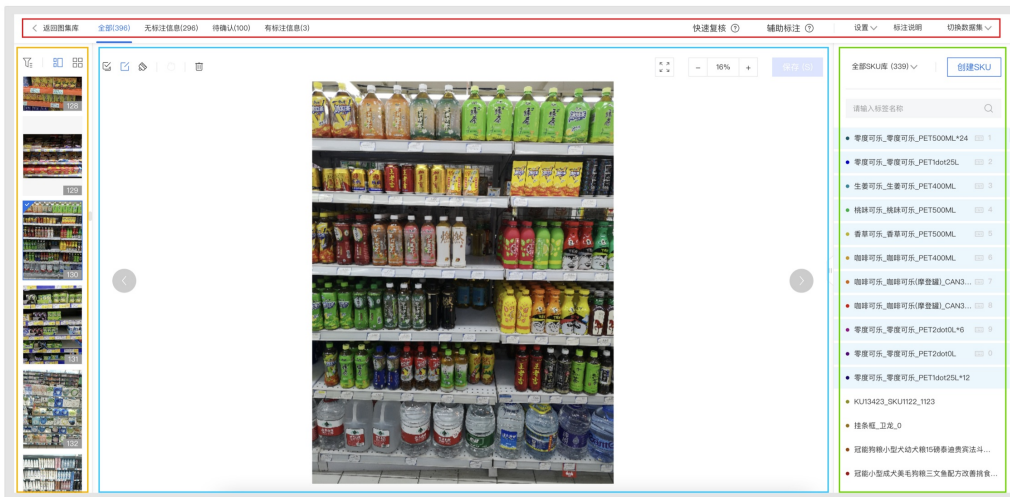
## 数据标注

### 在线标注工具介绍

本文档介绍EasyDL零售版在线标注工具中各个功能的简介。

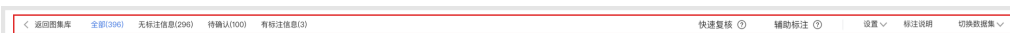
### 在线标注工具页面布局

可以使用EasyDL零售版在线标注工具进行标注，在线标注工具提供的多种功能提高标注效率，如下图所示，下面会对Banner区域（红）、实景图预览区（黄）、图片标注区（蓝）、SKU标签区（绿）四个区域的功能进行介绍。



### 在线标注工具介绍

#### Banner区域



从左至右依次为：

#### 1. 返回图集库

点击后返回**实景图集库**，如果当前图片的标注信息未保存，将会丢失

#### 2. 全部

该Tab显示图集下的所有图片，包含无标注信息、待确认、有标注信息的所有图片

#### 3. 无标注信息

该Tab显示图集下的所有未标注的所有图片，**该Tab下的图片不会参与模型训练**

#### 4. 待确认

该Tab显示**辅助标注**功能预标注后的图片，**该Tab下的图片不会参与模型训练**，关于辅助标注的功能详情请见[产品文档](#)

#### 5. 有标注信息

该Tab显示手动标注和在「待确认」Tab下确认后的图片，**该Tab下的图片会参与模型训练**

#### 6. 快速复核

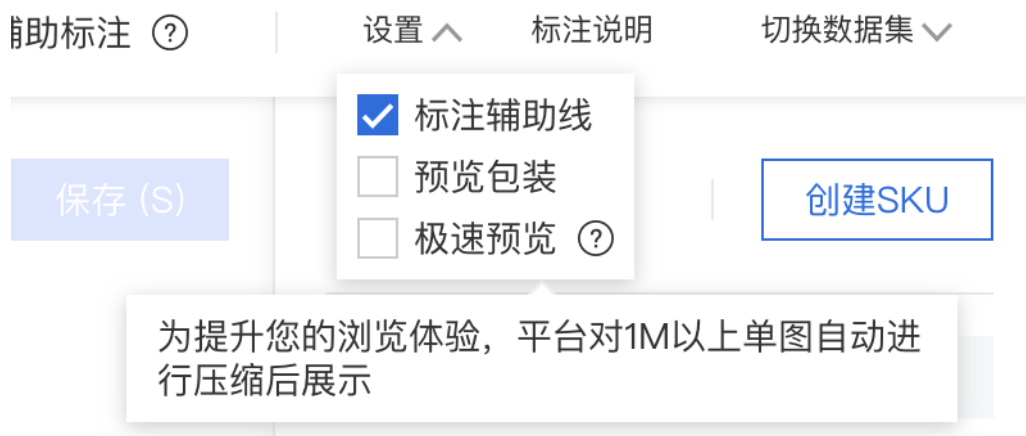
该功能可将「有标注信息」下图片上所有标注框，按照SKU分类后汇总展示，并支持修改或删除标注信息，可用于快速复核标注是否准确，关于快速复核的功能详情请见[产品文档](#)



## 7. 辅助标注

该功能可预先给未标注的实景图图上无标签的橙色辅助标注框和有标签的非橙色标注框，关于辅助标注的功能详情请见[产品文档](#)

## 8. 设置



- 标注辅助线，勾选后，标注时会有辅助线更精确地画标注框
- 预览包装，勾选后，鼠标HoverSKU标签或是标注框时，右侧标签栏会显示SKU包装图，SKU包装图
- 极速预览，勾选后，平台对1M以上单图自动进行压缩后展示，可以一定程度提升图片加载速度

## 8. 标注说明

- 实景图上传API，可以通过API上传未标注和已标注的实景图，点击后跳转至该说明文档
- 实景图集标注规范，详细说明实景图如何正确标注，点击后跳转至该说明文档
- 标注工具使用方法，详细说明在线标注工具的使用方法，点击后跳转至该说明文档

## 9. 切换数据集

可在此处切换实景图集

## 实景图预览区



从左至右依次为：

## 1. 按SKU筛选

可以按照SKU标签筛选包含该SKU标注信息的实景图

## 2. 卡片视图

默认状态，仅在页面左侧显示实景图图片缩略图，点击可以切换图片，切换时不会自动保存该图片上标注信息的修改内容，如需保存该图片上的标注信息，可以通过「保存」按钮或是「左右切换」按钮进行保存

## 3. 拓展卡片视图

点击后，可以预览所有图片，该视图下可批量删除图片，或快速定位图片，切换图片时不会自动保存该图片上标注信息的修改内容

## 图片标注区



从左至右依次为：

## 1. 多选标注

支持同时选择多个标注框，点击右侧标签后，为选中的多个标注框一次性附上标签

## 2. 单选标注

可在图片上进行画框标注，可选择一个已存在的标注框，点击右侧标签后，为选中的一个标注框一次性附上标签

## 3. 清除

可清除单张图片上的橙色辅助标注框

## 4. 拖动

可以拖动图片，配合放大、缩小工具使用调整图片显示

## 5. 删除

可删除当前显示的实景图片

## 6. 全屏

开启全屏标注

## 7. 缩小

将图片缩小显示

## 8. 放大

将图片放大显示

## 9. 保存

当实景图标注或是已有标注发生变化时，可点击，保存当前图片的标注信息，快捷键S

## 10. 左右切换

上一张/下一张图片，切换图片时会自动保存该图片上标注信息的修改内容

## SKU标签区

### 1. 切换标签组

选择标签组后，标签栏内仅显示该标签组包含的SKU，可用于缩小SKU标签显示范围，方便查找。标签组功能详情请见[产品文档](#)



### 2. 创建SKU

点击后，可在当前页面创建SKU

### 3. 搜索栏

可以搜索包含关键字的SKU，支持按SKU名称、品牌名称、规格参数进行搜索，方便找到常用的标注标签，可配合锁定和置顶工具使用

### 4. SKU标签列表

标签显示优先级：锁定的标签 > 置顶的标签 > 图上有标注框的标签 > 最新创建的标签，优先级越高，展示位置越靠前。

鼠标放在标签上停留几秒可以显示完整的标签名。

● 零度可乐\_零度可乐\_PET1dot25L\*12 锁定 置顶  
🔒 ⬆

● KU13423\_SKU 完整的标签名 零度可乐\_零度可乐\_PET1dot25L\*12

- 置顶

可将所选标签置顶，方便标注时选择需要标注的常用标签，置顶后，标签底色为蓝色

● 零度可乐\_零度可乐\_PET1dot25L\*12 置顶的标签

● KU13423\_SKU1122\_1123 非置顶的标签

- 锁定

用于锁定该SKU标签，锁定后，单选标注和多选标注选择中的标注框都将以锁定的标签命名，锁定后的标签，底色为蓝色，位于标签栏最上方，优先级高于「置顶」的标签

- 快捷键

位于标签列表的前10个标签，快捷键依次为1至0，在画完或选中标注框后，可通过键盘上快捷键快速给标注框选上标签

● 零度可乐_零度可乐_PET500ML*24	☰ 1
● 零度可乐_零度可乐_PET1dot25L	☰ 2
● 生姜可乐_生姜可乐_PET400ML	☰ 3
● 桃味可乐_桃味可乐_PET500ML	☰ 4
● 香草可乐_香草可乐_PET500ML	☰ 5
● 咖啡可乐_咖啡可乐_PET400ML	☰ 6
● 咖啡可乐_咖啡可乐(摩登罐)_CAN3...	☰ 7
● 咖啡可乐_咖啡可乐(摩登罐)_CAN3...	☰ 8
● 零度可乐_零度可乐_PET2dot0L*6	☰ 9
● 零度可乐_零度可乐_PET2dot0L	☰ 0

● KU13423\_SKU1122\_1123

#### 快捷键总览

S键：保存当前标注内容

方向左键：切换至上一张图片，并保存当前图片标注内容

方向右键：切换至下一张图片，并保存当前图片标注内容

数字键1~9：给标注框打上对应键位的标签

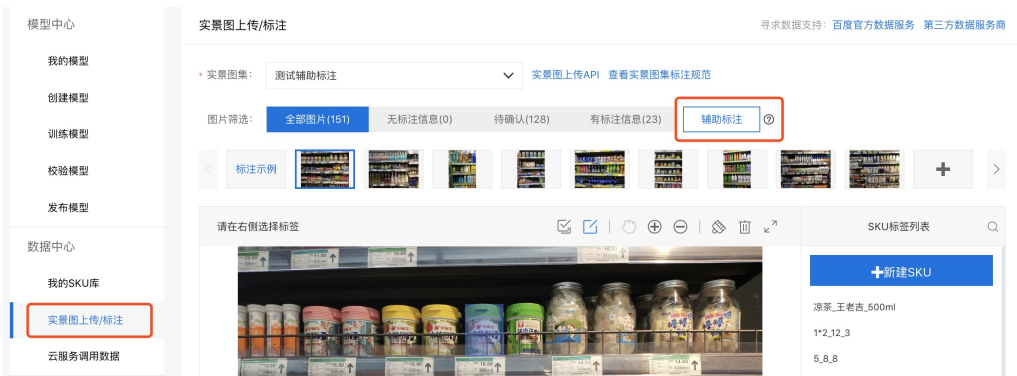
#### 🔗 实景图标注

实景图的标注工作可以在EasyDL零售版平台上完成，也可以在线下完成后上传到平台上，下面将分别介绍两种方法。

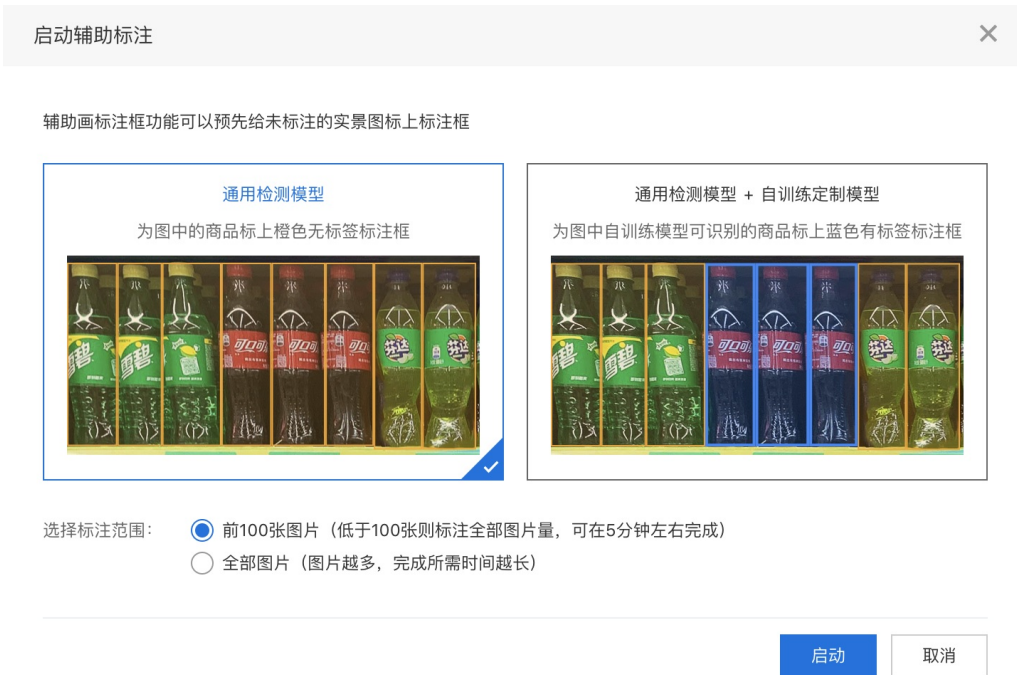
#### 在线标注-辅助标注

##### 辅助标注介绍

EasyDL零售版的在线标注工具提供了辅助标注功能，该功能可以使用平台预置模型和用户自训练定制模型预先为未标注的图片进行预标注，来降低整体标注工作的成本。在EasyDL零售版的[实景图上传/标注页面](#)可以使用该功能进行标注，如下图所示：



辅助标注功能提供两种预标注方式的选择，一种是通用检测模型，另一种是通用检测模型+自训练定制模型，如下图所示：



### 启动辅助标注

#### 选项1. 通用检测模型

选择「通用检测模型」可预先给未标注的实景图标上无标签的橙色辅助标注框，如下图所示：



注：只为需要识别的SKU的所有橙色辅助标注框附上标签，不需要识别的不处理不影响训练

在参考[实景图上传文档](#)将图片上传至平台后，在页面选择需要辅助标注的图片量，由于图片量越大，完成辅助标框所需时间越长，两个选项的相应建议如下：

前100张图片（低于100张则标注全部图片量，可在5分钟左右完成）



- 在尝试第一次标注不同包装类型的商品时（如瓶装、袋装、盒装等），建议选该选项
- 需要立刻开始标注图片时，建议选该选项

全部图片（图片越多，完成所需时间越长）

- 未标注的图片量很大，可以等待数小时后在进行标注时，建议选该选项

注：图片量和所需完成时间的关系大致为：所需时长=未标注图片量/100 \* 5 (分钟)

选项2. 通用检测模型+自训练定制模型

选择「通用检测模型」可预先给未标注的实景图标上无标签的橙色标注框和有标签的蓝色标注框，如下图所示：



此选项需要选择自训练定制模型的版本，建议选择mAP值高的版本。

注：支持选择2019年9月1日后完成训练的模型版本，建议选择mAP数值高的模型版本

启动辅助标注

辅助画标注框功能可以预先给未标注的实景图标上标注框

通用检测模型

为图中的商品标上橙色无标签标注框

通用检测模型 + 自训练定制模型

为图中自训练模型可识别的商品标上蓝色有标签标注框

选择标注模型： 27185: silencessecond V4 - mAP: 56.73%

支持选择2019年9月1日后完成训练的模型版本，建议选择mAP数值高的模型版本

选择标注范围：  
 前100张图片（低于100张则标注全部图片量，可在5分钟左右完成）  
 全部图片（图片越多，完成所需时间越长）

点击启动后，页面如下图所示，后台开始根据所选择的图片范围标注图片，此时可以去往其它页面或是退出该页面，比如标注其它实景图集中的图片，但是只允许同时对一个图集启动辅助标注框功能。

实景图上传/标注 寻求数据支持：[百度官方数据服务](#) [第三方数据服务商](#)

实景图集： [实景图上传API](#) [查看实景图集标注规范](#)

🕒 正在为图片画标注框，预计5分钟左右完成，请稍候...

完成所需时间与未标注图片量有关，[了解功能详情](#)

[终止](#)

**标注橙色辅助标注框**

为需要识别的SKU的所有橙色辅助标注框附上标签，无需识别的不处理不影响训练



**还原所有未确认图片**

可将剩余未确认的图片还原到未标注，图片上橙色辅助标注框将被清空



**清空单张图片辅助标注框**

如果对单张图片辅助标注框不满意，可清除单张图片中的辅助标注框



过程中，可以点击「终止」按钮终止辅助标注框进程，终止后图片将恢复至启动前状态。

### 完成辅助标注

辅助标注完成后，选择到相应启动辅助标注功能的图集后，显示如下页面：

实景图上传/标注 寻求数据支持：[百度官方数据服务](#) [第三方数据服务商](#)

实景图集： [实景图上传API](#) [查看实景图集标注规范](#)

✅ 辅助画标注框已完成，请在“待确认”图片分类下进行标注

[前往标注](#)

**标注橙色辅助标注框**

为需要识别的SKU的所有橙色辅助标注框附上标签，无需识别的不处理不影响训练



**还原所有未确认图片**

可将剩余未确认的图片还原到未标注，图片上橙色辅助标注框将被清空



**清空单张图片辅助标注框**

如果对单张图片辅助标注框不满意，可清除单张图片中的辅助标注框



点击「前往标注」按钮后，会进入到标注页面，使用辅助标注功能处理的图片会被归类到待确认的分类Tab下，如下图所示：



可以看到图片上会出现橙色的无标签标注框，如果选择了「通用检测模型+自训练定制模型」，则部分可识别的商品会被打上蓝色的有标签标注框。

### 确认辅助标注图片

对于两类辅助标注的结果，需要做如下两个操作：

#### 1. 标注橙色标注框

对于橙色无标签的标注框，标注方式为选中一个标注框，点击右侧标签进行标注。

注：只为需要识别的SKU的所有橙色辅助标注框附上标签，不需要识别的不处理不影响训练。

#### 2. 核对蓝色标注框

对于蓝色有标签的标注框，需要核对每一个标注框的标签是否正确，如果不正确，需要选中标注错误的标注框，再在右侧为标注框选择正确的标签。

核对标注好图片后，点击「确认并保存当前图片」或「切换并保存」按钮，该张图片会被保存到「有标注信息」Tab下。

标注栏工具也提供了多选标注功能，可以选中多个标签后进行标注，详情见「标注工具的使用」部分。

### 异常处理：还原所有未确认图片

如果发现很多图片上大量的辅助标注框位置都不准确，可以点击「还原所有未确认图片」按钮，按钮位置如下图所示，将所有还没有确认的图片全部还原到「未标注」的图片分类Tab下，图片上的橙色辅助标注框也将被清空，这种情况建议不使用辅助标注框功能，用一般标注进行标注，也可以加入EasyDL零售版官方QQ群（群号1009661589），将不准确的情况反馈给我们，帮助我们优化提升该功能。



### 在线标注-手动标注

对于辅助标注功能无法满足的图片，可以参考[标注工具使用方法](#)文档，使用在线标注工具手动标注。

### 线下标注

#### 线下标注数据上传



如果您打算使用线下的标注工具（如标注精灵、labelme等）标注数据或是已有一些标注好的数据，可以参考[实景图管理API](#)文档将线下标注好的数据上传至EasyDL零售版训练平台。

## 🔗 货架场景实景图标注规范

### 简介

数据的标注质量决定了模型的效果，本文档详细介绍各个场景采集的实景图应该如何标注。可以参考[实景图数据要求文档](#)，了解各个业务场景的数据采集规范。

点击下载[数据采集与标注规范长图](#)，一张图看懂如何采集和标注数据，让您不走弯路，获得一个高精度的商品检测模型。

### 实景图标注原则

#### 标注框的要求

标注框是标注时的最小单位，需要能够完全覆盖目标SKU的最小矩形框，如下图所示：



#### SKU被遮挡时的标注方法

无人零售货架和智能结算台业务场景中，只标注商品露出来的部分。普通货架/货架审核场景，建议只标注露出部分超过70%且具备识别特征的SKU，如下图所示：



- 1** 不需标注:虽然商品露出部分超过70%但无明显特征
- 2** 需要标注:露出超过70%,且具备明显识别特征  
标注方法:最小矩形框框选未被覆盖部分
- 3** 需要标注:露出超过70%,且具备明显识别特征  
标注方法:因底部未被全部遮盖,故最小矩形框框选整体商品
- 4** 不需标注:露出部分不足70%
- 5** 需要标注:露出超过70%,且具备明显识别特征  
标注方法:最小矩形框框选未被覆盖部分
- 6** 不需标注:虽然商品露出部分超过70%但无明显特征

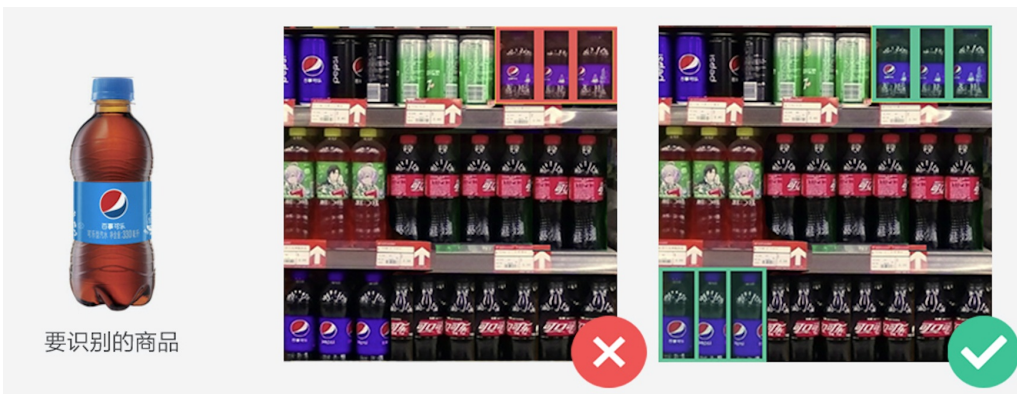
**相似SKU需要标注**

普通货架/货柜审核场景中，实景图上容易出现与目标SKU很相似的SKU，这些SKU也建议进行标注，能够降低模型将这些相似的SKU识别为目标SKU的可能性。相似SKU分为以下几种情况：

**1. 同品牌不同系列、口味、包装**



**2. 不同规格的同款产品**



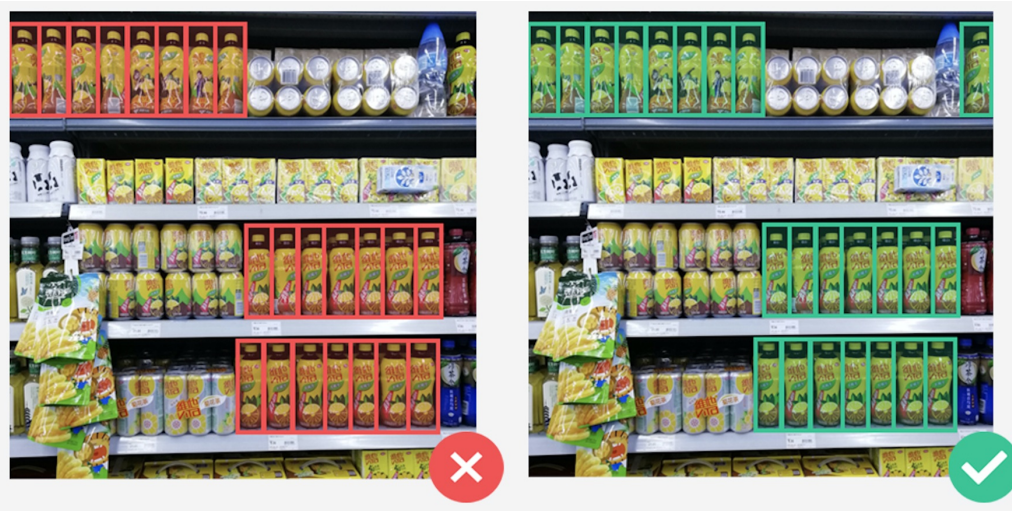
**3. 过于相似的竞品**





避免误标注和漏标注

误标注即将非目标SKU标注为目标SKU，漏标注即实景图上目标SKU没有全部被标注出来，标注的时候尽量避免这些情况发生，较多的误标注和漏标注会严重影响模型效果。下图为漏标情况：



地堆场景实景图标注规范

简介

数据的标注质量决定了模型的效果，本文档详细介绍各个场景采集的实景图应该如何标注。可以参考[实景图数据要求文档](#)，了解各个业务场景的数据采集规范。

实景图标注原则

以下图举例，如果需要识别的是「青岛纯生」、「青岛白啤」、「青岛经典10度」三种箱装SKU。



正确的标注方式如下图所示：



## 1. 区分顶箱SKU和非顶箱SKU创建标签

需要创建如下四个SKU，【SKU名称】、【品牌名称】、【规格参数】通过【\_】连接：

- 1) 箱装经典10度\_青岛啤酒\_500ml
- 2) 箱装经典10度\_顶箱\_青岛啤酒\_500ml
- 3) 箱装白啤\_青岛啤酒\_500ml
- 4) 箱装白啤-顶箱\_青岛啤酒\_500ml
- 5) 箱装纯生\_青岛啤酒\_500ml
- 6) 箱装纯生-顶箱\_青岛啤酒\_500ml

注：【SKU名称】、【品牌名称】、【规格参数】中「顶箱」或「topbox」需用符号「-」连接在文字最后，如【原味可乐-顶箱】或【原味可乐-topbox】

## 2. 区分顶箱SKU和非顶箱SKU标注

### 3. 完全覆盖目标SKU的最小矩形框

### 4. 避免误标注和漏标注

误标注即将非目标SKU标注为目标SKU，漏标注即实景图上目标SKU没有全部被标注出来，标注的时候尽量避免这些情况发生，较多的误标注和漏标注会严重影响模型效果。

## 快速复核标注

本文档介绍EasyDL零售版在线标注工具中快速复核功能，在完成标注后，可以使用该功能对标注的图片进行复核，检查标注的是否正确。

该功能可将【有标注信息】下图片上所有标注框，按照SKU分类后汇总展示，并支持修改或删除标注信息，可用于快速复核标注是否准确。

### 使用流程

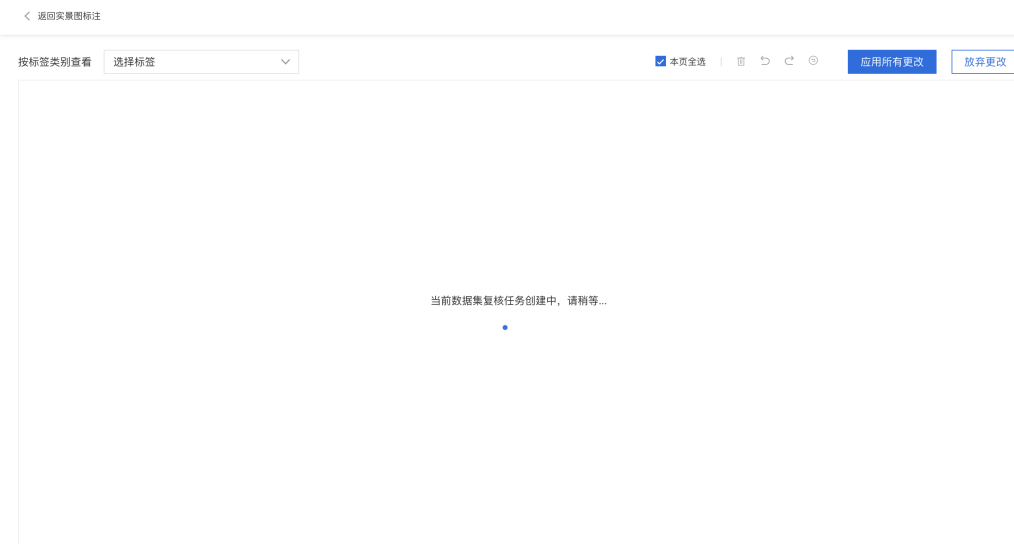
#### 1. 功能入口

进入标注页面后，在下图所示位置，点击「快速复核」

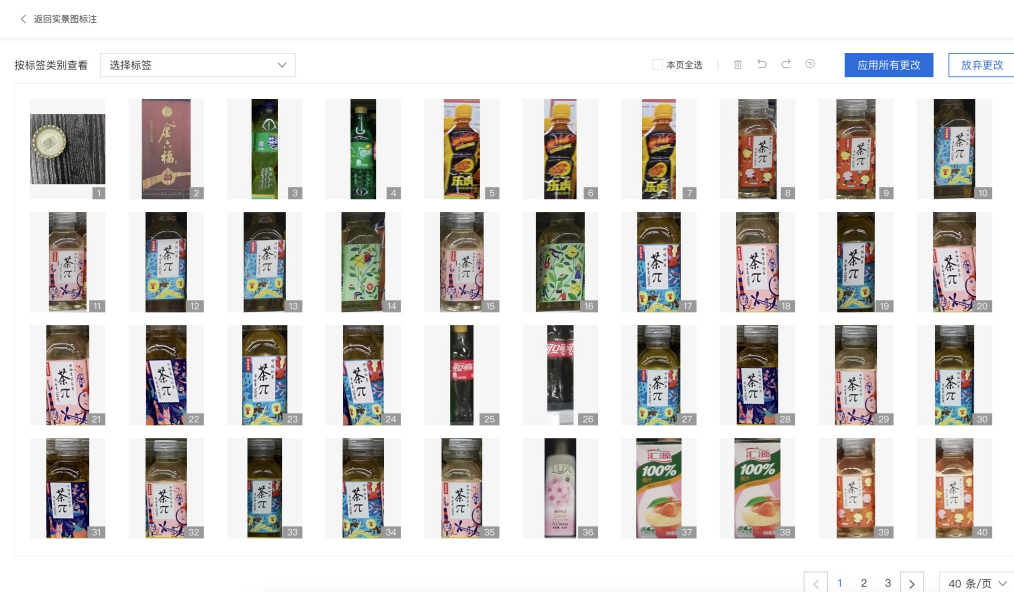


#### 2. 启动快速复核

点击「快速复核」后即启动该功能，启动后需等待一段时间，等待过程中，不可返回实景图标注和中断流程，可以通过新开页面的方式进行其他操作，如等待时间超过20分钟，可以[提交工单](#)进行反馈。



启动完成后, 您将看到如下页面, 每一个标注框将以小图的形式显示。



### 3. 复核标注

#### Step1. 选择需要复核的标签

可以滑动或是通过搜索框查找需要复核的SKU



按标签类别查看

选择标签

请输入标签名称

全部

KU13423\_SKU1122\_1123

挂条框\_卫龙\_0

冠能狗粮小型犬幼犬粮15磅泰迪贵...

冠能小型成犬美毛狗粮三文鱼配方...

冠能猫粮室内猫粮优护益肾控毛球...

冠能猫粮幼猫1-12月奶猫奶糕孕猫...

选择SKU标签后，下方将仅显示这个SKU的小图（即该SKU标注过的每一个标注框内容）

按标签类别查看 茶兀柠檬红茶\_农夫山泉\_500ml

全部SKU (339)

创建SKU

请输入标签名称

- 零度可乐\_零度可乐\_PET500ML\*24 1
- 零度可乐\_零度可乐\_PET1.5L\*24 2
- 生姜可乐\_生姜可乐\_PET400ML 3
- 麒麟可乐\_麒麟可乐\_PET500ML 4
- 香茅可乐\_香茅可乐\_PET500ML 5
- 麒麟可乐\_麒麟可乐\_PET400ML 6
- 麒麟可乐\_麒麟可乐(摩登罐)\_CAN3... 7
- 麒麟可乐\_麒麟可乐(摩登罐)\_CAN3... 8
- 零度可乐\_零度可乐\_PET2.0L\*6 9
- 零度可乐\_零度可乐\_PET2.0L 0
- 零度可乐\_零度可乐\_PET1.5L\*12 11
- 卫龙312g大面筋香辣味\_卫龙\_312g\*12袋
- KU13423\_SKU1122\_1123
- 挂条框\_卫龙\_0
- 冠能狗粮小型犬幼犬粮15磅泰迪贵...

鼠标悬停在小图上，可以显示该小图（标注框）的SKU标签名称

茶兀柠檬红茶\_农夫山泉\_500ml



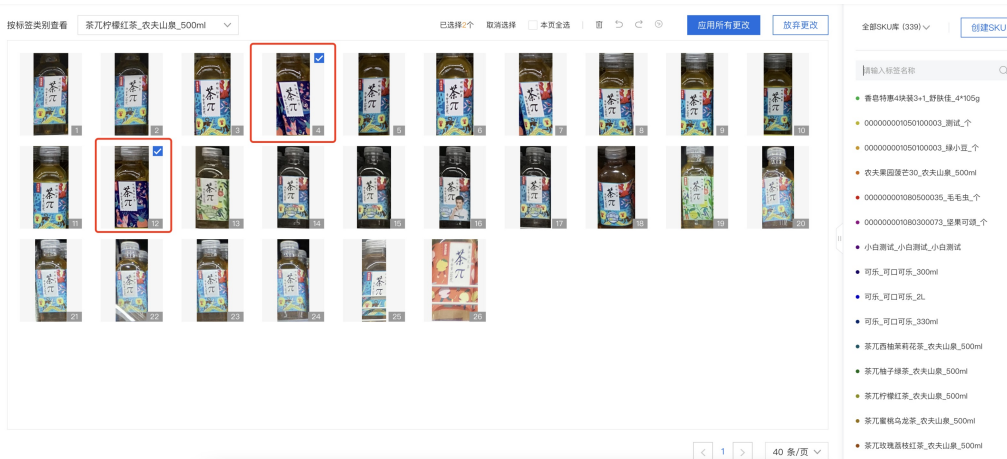
也可以通过点击「放大」，查看大图，以查看更多商品的细节



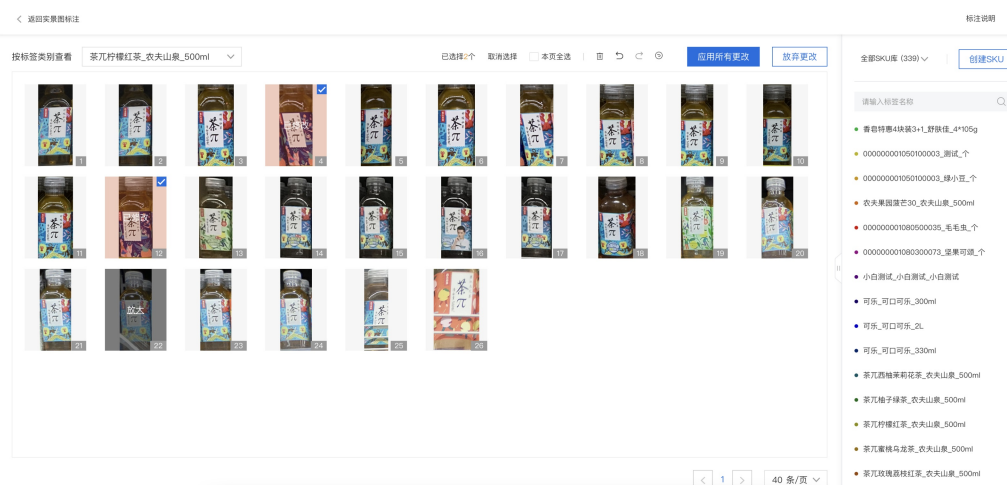
### Step2. 修正或删除标注

#### 1) 修正标注

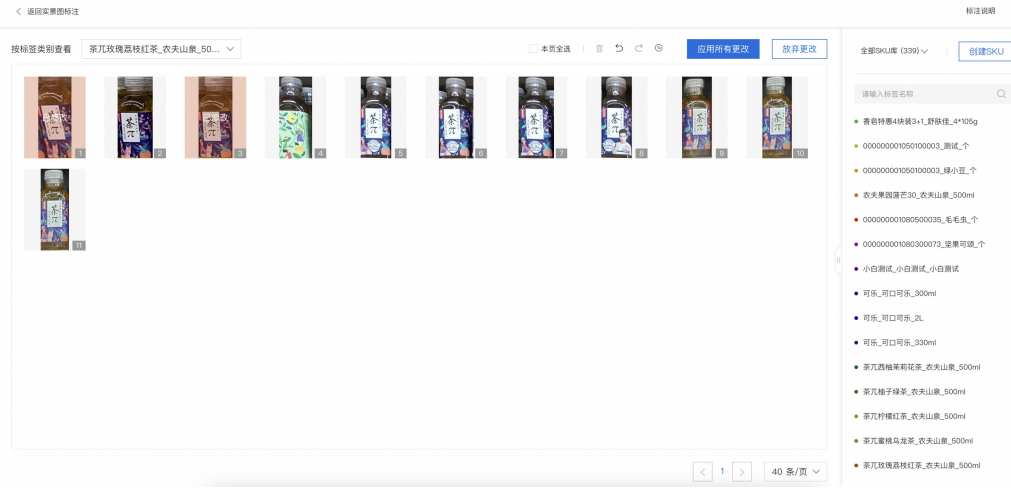
选中单个或是多个同类标注错误的图片



在右侧标签栏选择正确SKU标签，选择后，图片会变为「已修改」状态



此时，在切换到修改后的SKU标签时，可以发现修改过的小图（标注框）



### 2) 删除标注

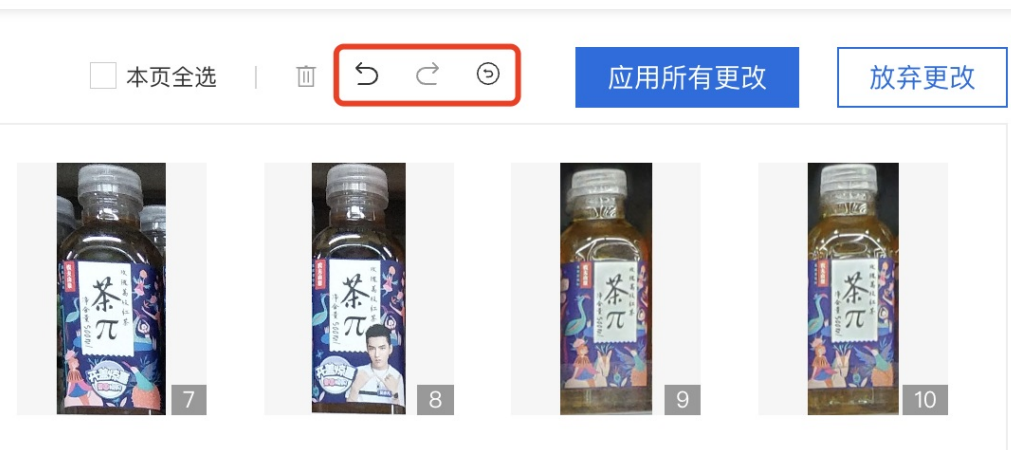
如果发现误标注且不属于需要标注的小图，可以删除标注。选中单个或多个小图，点击删除按钮删除



删除后，图片的状态变为「已删除状态」



### 3) 一些其它支持的操作



从左至右分别是：

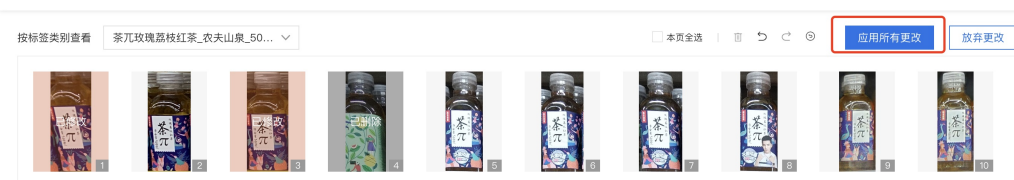
- 撤回上一步：取消上一步的操作（修改、删除）
- 重做上一步：重做上一步的操作（修改、删除）
- 还原本页所有操作：取消当前页上的所有操作（修改、删除）

### 4. 完成复核

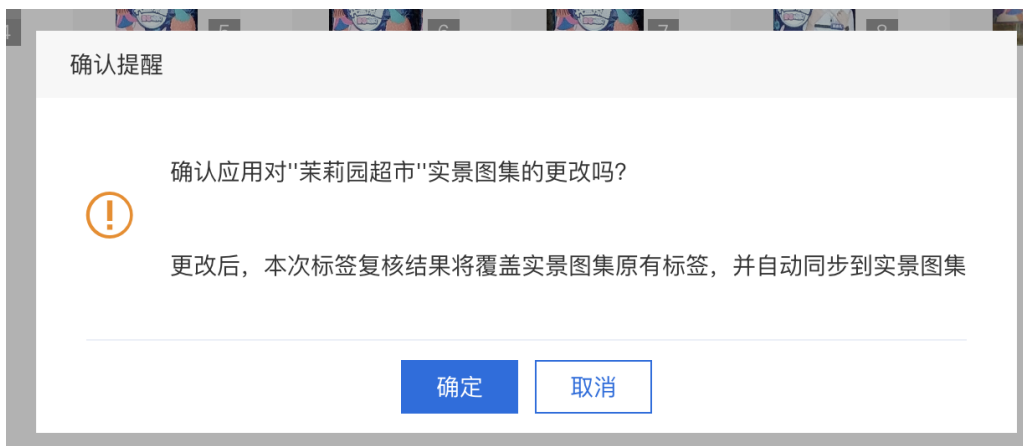
#### 应用所有更改

当复核完所有SKU时，可以点击「应用所有更改」完成复核工作。



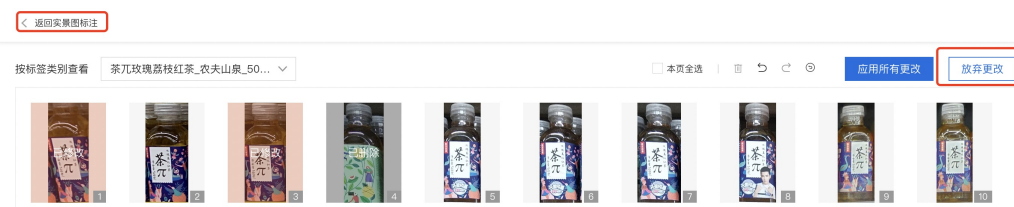


确认后，本次标签复核结果将覆盖实景图集原有标签，并自动同步到实景图集。

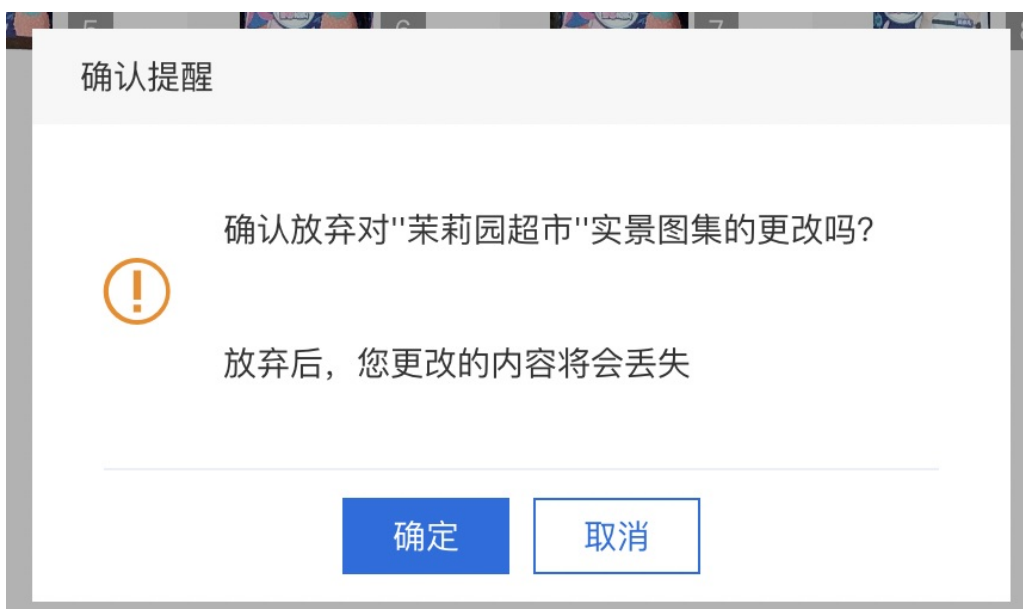


### 放弃所有更改

如果不想将复核所做的修改应用到实景图集中，可以点击「放弃更改」或是「返回实景图标注」。



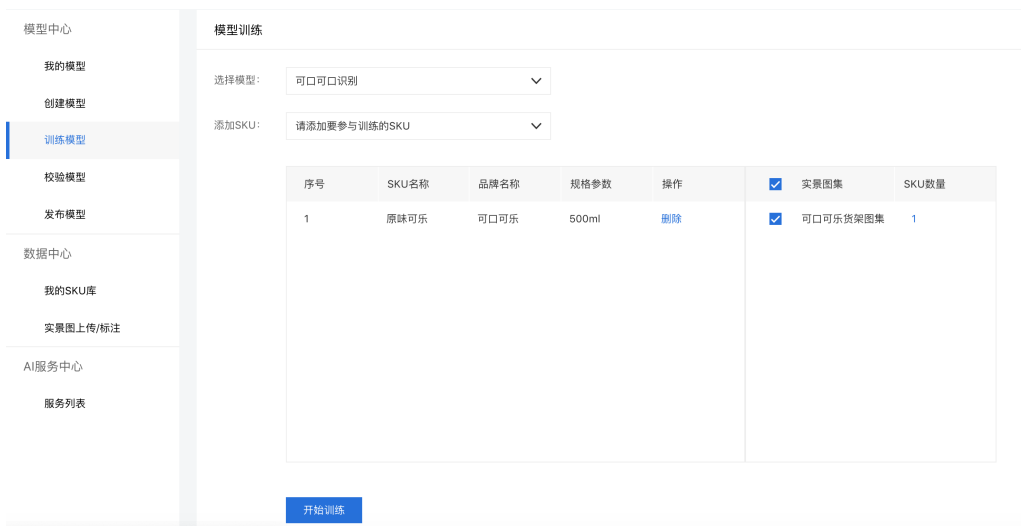
确认后，所有修改的内容将会丢失。



## 模型训练

### 🔗 模型训练操作说明

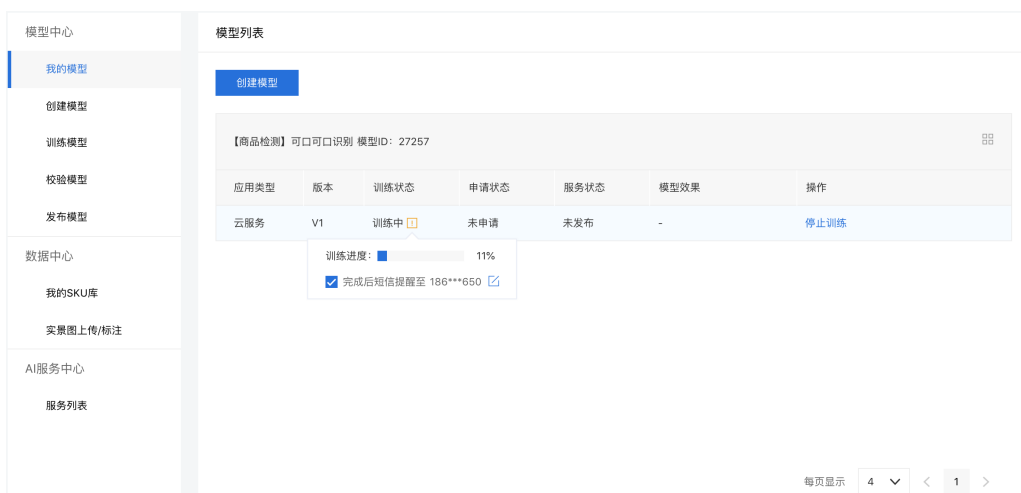
在完成[创建模型](#)和[实景图上传和标注](#)后，即可开始训练模型。在[模型训练页面](#)，点击左侧列表中的【训练模型】进入[模型训练页面](#)，您会看到如下图所示的内容：



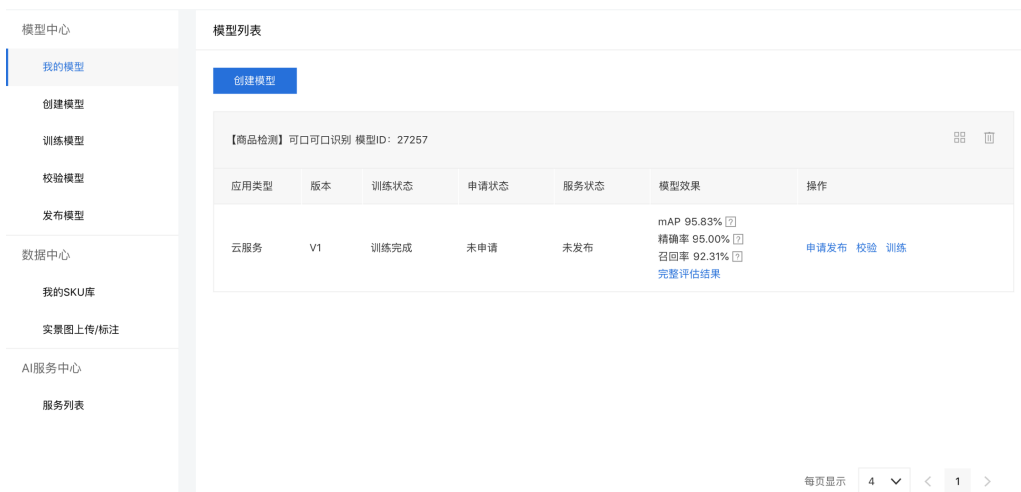
如上面图片所示，点击左侧列表中的【训练模型】，需要先后完成下面三项选择：

1. 选择要训练的模型
2. 选择需要模型支持检测的SKU，选择完成后，下方左侧会显示已添加的SKU，右侧会显示包含已添加SKU的实景图集
3. 选择要参与训练的实景图集

完成选择后，点击【开始训练】按钮页面跳转至【我的模型】页面，如下图所示，可以看到模型已进入训练状态，将鼠标移至状态"训练中"右边的小问号上，可以查看训练进度，训练进度数值只是作为参考，所以推荐打开短信通知功能，这样就第一时间知晓模型训练完成了。



训练完成后，可以使用[校验模型](#)功能上传少数几张实景图验证模型效果，也可以直接申请[发布模型](#)。



## 🔗 训练时长与等待时间说明

EasyDL训练平台各类模型均是使用GPU集群进行训练，一个模型训练通常需要几十分钟至几个小时不等，在EasyDL零售版中，训练时长与参与训练的SKU单品图和实景图数量有关，下表为各种训练数据量级所需要的大致训练时间：

训练实景图片数	SKU是否上传单品图	训练平均时长
6000以上	是	10小时以上，24小时以内
6000以上	否	10小时以上，24小时以内
4000~6000	是	10小时左右
4000~6000	否	8小时左右
2000~4000	是	6小时左右
2000~4000	否	4小时左右
1000~2000	是	5小时左右
1000~2000	否	3小时左右
500~1000	是	4小时左右
500~1000	否	2小时左右
100~500	是	3小时左右
100~500	否	1小时左右
100以内	是	3小时左右
100以内	否	1小时以内

注：上述训练时长为多次训练的平均值，仅供参考，建议开启短信通知（如下图），以便模型训练完成时，我们能够第一时间通知到您。

## 模型效果评估

### 简介

在参考[模型训练操作说明文档](#)完成模型训练后，可参考此文档了解模型效果。

### 模型训练结果

#### 模型的训练结果是如何得到的？

上传的实景图，只有标注过的图片会被训练，所有训练图片中，系统会随机抽取70%的标注数据作为训练数据，剩余的30%作为测试数据，训练数据训练出的模型去对测试数据进行检测，检测得到的结果跟人为标注的结果进行比对，得到页面显示的mAP，精确率和召回率。

提示：训练数据，即上传标注的实景图图片越接近真实业务里需要预测的图片，模型训练结果越具有参考性。

### 模型版本详情

模型列表

创建模型

【商品检测】李锦记 模型ID: 88538						训练	历史版本	删除
部署方式	版本	训练状态	申请状态	服务状态	SKU F1-score 分布	操作		
公有云API	V2	训练完成	未申请	未发布	[85%, 100%] 4个 [60%, 85%) 12个 [0%, 60%) 5个	<a href="#">查看版本详情</a>	<a href="#">申请发布</a>	<a href="#">校验</a>

模型训练好后，可以在模型列表中看到SKU的F1-score分布情况，如果需要了解更为详细的模型效果表现，可以在模型列表中点击「查看版本详情」。

## 基础信息

模型列表 &gt; 李锦记 &gt; V2

### 基础信息

模型ID 88538 版本 V2 图片数 85 SKU数 21

训练完成时间: 2020-12-14 18:50 训练算法: 公有云API-默认算法

模型版本的基础信息包含以下内容：

- 模型ID
- 训练版本
- 训练图片数
- 训练SKU数
- 训练完成时间
- 训练算法

### SKU F1-score分布



直观展示模型中SKU F1-score的分布，F1-score是模型中一个SKU的精确率和召回率的调和平均数，可以作为判断模型中各SKU效果的指标，通常情况下：

- F1-score>85%时，可满足商品计数需求
- F1-score>60%时，可满足统计商品分销需求

### 模型整体效果

#### 模型整体效果



页面上显示的分销准确率、mAP、召回率和精确率数值，是模型里所有SKU在建议阈值下的平均值，建议阈值可以在模型的「模型整体F1-score走势图」中查看。四项指标的含义分别为：

#### • 分销准确率

按图片粒度统计分销准确率，即单张评估图片中正确识别出所有SKU种类的平均概率。

分销准确率 = 一张图片内正确识别的SKU种类数 / (人工标注出的SKU种类数 + 模型识别出的SKU种类数)

#### • mAP

mAP在[0,1]区间，越接近1模型效果越好，mAP不高也不说明模型里所有的SKU识别效果不好。

mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务，每一类object都可以计算出其精确率(Precision)和召回率(Recall)，在不同阈值下多次计算/试验，每个类都可以得到一条P-R曲线，曲线下的面积就是average precision(AP)的值。“mean”的意思是对每个类的AP再求平均，得到的就是mAP的值。

#### ● 精确率

对于一个SKU而言，精确率越高，说明模型识别出是这个SKU的所有结果中，正确数量的占比越高。如果精确率为1，说明识别出的所有结果都是对的，但不说明该SKU全部都被识别出来了，可能会存在漏识别。

精确率 Precision = 模型正确预测为该SKU的数量/模型预测为该SKU的总数

#### ● 召回率

对于一个SKU而言，召回率越高，说明模型越完整地识别出这个SKU。如果召回率为1，说明这个SKU全部都被模型识别出来了，但不表示识别出是这个SKU的结果都是对的，可能会存在误识别。

召回率 Recall = 模型正确预测为该SKU的数量/SKU客观存在的总数

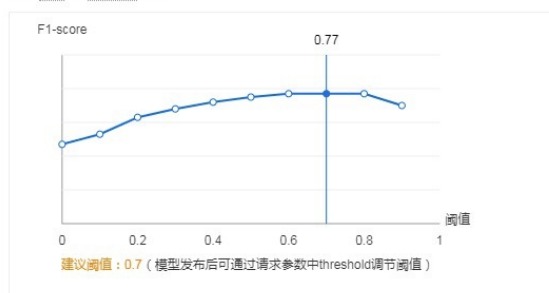
#### 模型整体F1-score走势图

F1-score是模型中一个SKU的精确率和召回率的调和平均数，在以相同权重考虑precision和recall的情况下，用来衡量一个模型的效果。

F1-score =  $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$

#### 模型整体F1-score走势图

不同 阈值 下 F1-score 表现



该曲线图展示了模型中各SKU在不同阈值下的F1-score平均值，根据该曲线可以得到阈值的最优值，即图中显示的「建议阈值」。模型列表和整体评估展示的各项模型效果指标数据，均是模型在「建议阈值」下的结果。另外，在模型发布后，调用服务API时可通过请求参数中threshold调节阈值，默认为建议阈值。

**阈值 (threshold)**，是正确结果的判定标准，例如阈值是0.6，置信度大于0.6的识别结果会被当作正确结果返回。

#### 训练及评估数据明细

##### 训练及评估数据明细

全部明细下载

No.	名称	训练图片数	训练标注框数	评估图片数	评估标注框数	F1-score	操作
1	薄盐生抽_李锦记_500ml	30	159	13	26	92.86%	<a href="#">查看详情</a>
2	薄盐生抽_李锦记_1.75L	9	19	1	3	66.67%	<a href="#">查看详情</a>
3	薄盐醇味鲜酱油_李锦记_500ml	9	21	6	11	95.65%	<a href="#">查看详情</a>
4	凉拌汁_李锦记_207ml	3	8	2	6	50.00%	<a href="#">查看详情</a>
5	蒸鱼豉油_李锦记_410ml	8	23	3	10	37.84%	<a href="#">查看详情</a>

每页显示  < 1 2 3 4 5 >

由于训练过程中，系统会随机抽取70%的标注数据作为训练数据，剩余的30%作为测试数据，所以模型训练存在一定的随机性。明细数据列表中展示了该模型版本训练时的图片数据和标注框的分布情况，帮助用户更具针对性的分析各个SKU的指标对应的数据量，以便针对性补充训练数据来优化模型。该表单支持下载，以便用作模型报告的制作。表单中的各列含义如下：

- No. : SKU在表单中的序号

- 名称：SKU的标签名称，名称`品牌规格`
- 训练图片数：实际用于训练模型的各个SKU的图片数量（训练模型时选择所有图集中含有图片总量的70%）
- 训练标注框数：实际用于训练模型的图片中，各个SKU的标注框数
- 评估图片数：用于评估个SKU训练效果的图片数量（训练模型时选择所有图集中含有图片总量的30%），即用来得到模型各项指标（mAP、Precision、Recall、F1-score）的评估集
- 评估标注框数：用于评估的图片中，各个SKU的标注框数
- F1-score：每个SKU的F1-score，F1-score是模型中一个SKU的精确率和召回率的调和平均数，可以作为判断模型中各SKU效果的指标，通常情况下：
  - F1-score>85%时，可满足商品计数需求
  - F1-score>60%时，可满足统计商品分销需求
- 操作：查看详情，可查看所有参与训练和评估的图片

## 模型优化

### 🔗 优化方法

#### 简介

模型训练好后，必经的一个过程是模型优化，EasyDL零售版已专门根据零售业务场景中的数据调优了模型算法，所以优化EasyDL零售版训练的模型，不需要理解和调优模型算法中的各种参数，仅需要优化训练数据即可。优化一个EasyDL零售版的商品检测模型，可以分为以下几个步骤进行：

1. 前提 - 正确采集实景图 and 单品图，并正确标注实景图
2. 补充实景图 - 使用EasyDL零售版提供的模型优化工具 ([云服务数据回流](#)) 补充实景图
3. 补充SKU单品图 - 上传SKU识别不好的角度的单品图
4. 重新训练模型 - 补充好数据后用新旧数据一起重新训练模型
5. 重新发布模型 - 将新训练的模型版本发布为API后测试模型效果
6. 重复优化 - 根据测试结果重复2-5步骤直到模型效果可商用

#### 1. 前提

数据质量是保证模型效果的前提，在EasyDL零售版中，数据质量涉及实景图 and 单品图的图片质量，以及实景图的标注质量，**开始模型优化前，请先学习如何采集合格的图片和进行合格的标注**，各个参考文档如下：

- 实景图采集：[实景图数据要求](#)
- 实景图标注：[实景图标注规范](#)
- SKU单品图：[SKU单品图数据要求](#)

点击下载[数据采集与标注规范长图](#)，一张图看懂如何采集和标注数据，让您不走弯路，获得一个高精度的商品检测模型。

#### 2. 补充实景图

推荐使用EasyDL零售版提供的模型优化工具 - [云服务数据回流](#)工具，参考使用文档[优化工具](#)补充实景图来优化模型。

#### 3. 补充SKU单品图

在上一步使用[云服务数据回流](#)工具时，发现SKU识别效果较差的角度，参考文档[SKU单品图数据要求](#)采集SKU相应角度的单品图并上传到对应SKU。

#### 4. 重新训练

在补充了相应数据后，在模型列表中「操作」列中点击「训练」，选择需要检测的SKU和所有需要训练的实景图集，包括上一次训练的图集和新增的图集（如果优化模型的数据是存在新的实景图集中），确认选择无误后开始训练。

提示：重新训练之前训练或发布过的模型，训练过程中和训练后均不会影响之前训练的模型版本，训练之前发布的版本API接口依旧有效，也可以选择发布之前训练过的其他模型版本。

同一个模型所有训练的的版本均可以在模型列表中进行查看，如下图所示：

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

数据中心

我的SKU库

实景图上传/标注

云服务调用数据

AI服务中心

服务列表

增值服务中心

货架拼接服务

模型列表

创建模型

【商品检测】 饮料检测 模型ID: 27899

训练 历史版本 删除

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V1	训练完成	审核成功	已发布	mAP : 96.95% 精确率: 88.46% 召回率: 92.00%	查看版本详情 配置服务功能 校验 体验H5 完整评估结果

【商品检测】 饮料检测 模型ID: 27888

训练 历史版本 删除

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V2	训练完成	未申请	未发布	mAP : 93.70% 精确率: 83.33% 召回率: 80.00%	查看版本详情 申请发布 校验 完整评估结果

每页显示 4 < 1 >

在模型列表中点击「全部版本」图标后进入到模型全部版本管理页面，如下图所示，可在页面上查看所有版本的训练集和选择发布任意已经成功训练好的版本。

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

数据中心

我的SKU库

模型列表 > 饮料检测

饮料检测全部版本

版本	训练状态	申请状态	服务状态	模型效果	操作
V2	训练完成	未申请	未发布	mAP : 93.70% 精确率: 83.33% 召回率: 80.00%	查看版本详情 申请发布 校验 完整评估结果
V1	训练完成	未申请	未发布	mAP : 95.72% 精确率: 85.19% 召回率: 92.00%	查看版本详情 申请发布 校验 完整评估结果

## 5. 重新发布模型

模型重新训练好后，如果之前发布过的模型版本还在测试，可以直接重新发布，用新的版本执行下一步；如果之前发布过的模型版本已经上线到生产环境，请参考下面注意内容，确保不影响线上生产环境的情况下，执行重新发布。

注意：重新发布模型后，服务API URL不会变化，模型自动切换为新发布的训练版本，如果服务API已上线生产环境，发布新版本前，请先在模型列表和完整评估报告中，确认新版本的模型指标和各SKU的精确度与已上线版本是更优或者相差不大的，以确保新发布版本的模型效果不会对线上业务不会产生不好的影响。

## 6. 重复优化

根据每次优化数据后的模型测试结果，重复2-5步骤直到模型效果可商用。

### 优化工具

#### 简介

本文档介绍如何使用云服务数据回流功能来优化商品检测模型。

#### 云服务调用数据管理

EasyDL零售版云服务数据回流功能，可用于查找云服务模型识别错误的的数据，纠正结果并将其加入模型迭代的训练集，实现训练数据的持续丰富和模型效果的持续优化。

提示：模型发布成功后，才可以开通该功能，如果模型还未发布，可以参考文档模型发布发布一个模型。

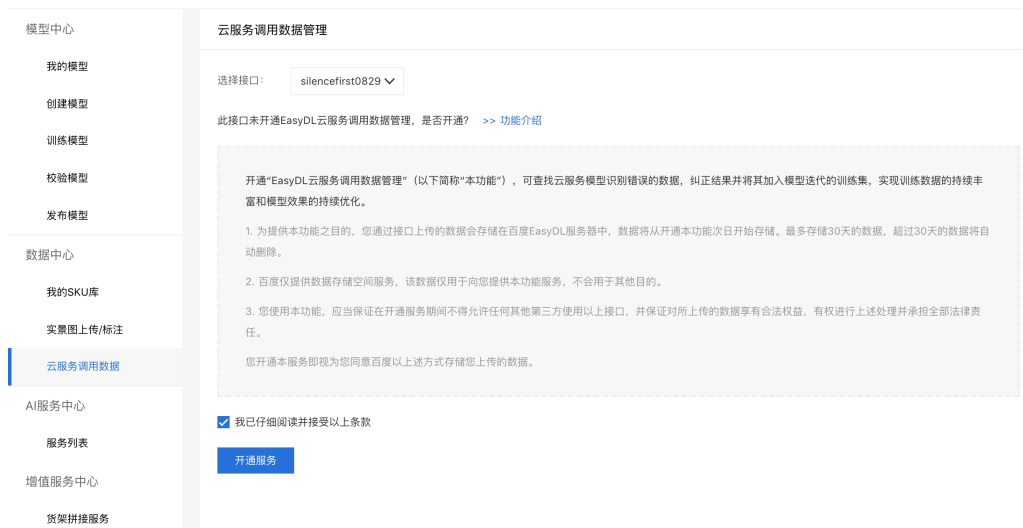
#### 使用步骤

该功能的使用步骤如下：

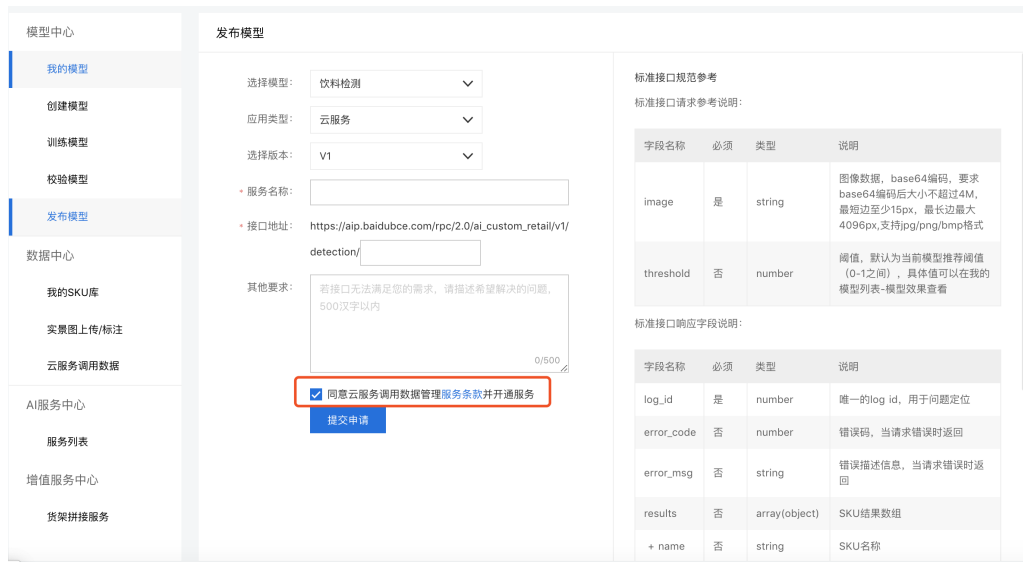
1. 开通功能
2. 筛选数据
3. 修正标注
4. 优化模型

### 步骤1. 开通功能

开通此项功能有两种方式，一是在**发布模型**页面，发布模型时勾选「同意云服务调用数据管理服务条款并开通服务」，发布后即可开通这项功能；二是在左侧导航栏「数据中心」点击「云服务调用数据」，在页面上选择**已发布**的定制商品检测模型接口，选择后仔细阅读服务条款，接受后即可开通这项功能。如下两张图所示：



「云服务调用数据」页面



「发布模型」页面

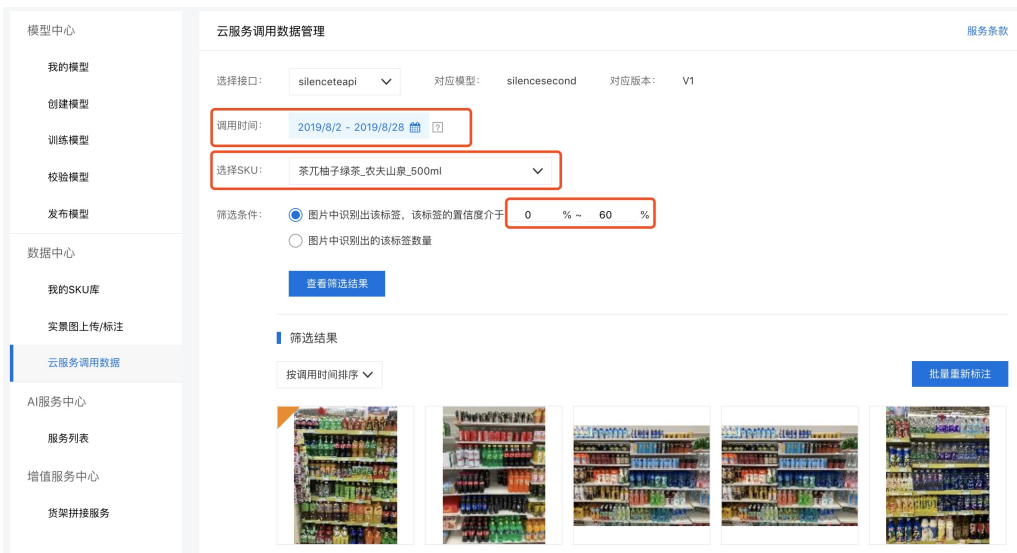
### 数据查看说明

服务开通后，次日服务开始生效，生效后接口调用的数据可在次日通过网页查看。

举例：如果您在2019年8月29日开通该功能，该功能将于2019年8月30日生效，如果在30日调用过接口，那么30日使用该接口识别过的图片，将在31日0点后可以再在网页上查看到。

### 步骤2. 筛选数据

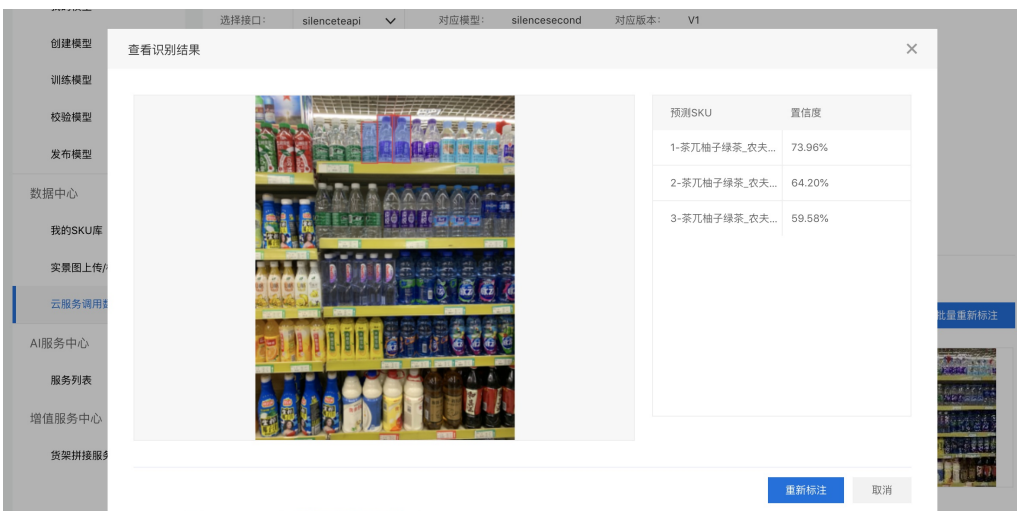




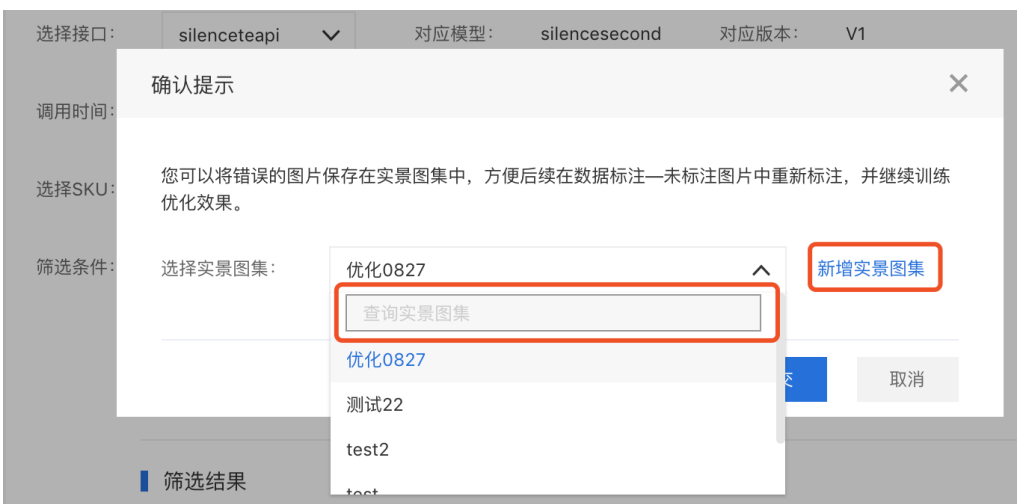
如上图所示，选择想要查询的接口调用时间和要筛选的SKU标签，筛选条件可以根据模型的阈值和业务的情况而定，筛选出的图片会显示在页面上。举个例子，比如调用接口识别图片时，设置的阈值（threshold）是0.6，业务上认为置信度达到80%以上才是可以接受的结果，那么这里置信度填写的标签应该是0~80%。这样，所有调用识别的图片中，含有该SKU标签且置信度在60%~80%的图片就会被筛选出显示在页面上。

阈值（threshold），是正确结果的判定标准，例如阈值是0.6，置信度大于0.6的识别结果会被当作正确结果返回。在调用接口时，可以通过参数「threshold」设定，如果不填，则默认设置为推荐阈值，推荐阈值可以在「我的模型」页模型的「完整评估结果」里查看。

### 步骤3. 修正标注



如上图所示，选择一张图片点击查看，可以看到图中三个识别结果都是误识别，点击「重新标注」后，在弹窗内选择需要将该图片添加到的实景图集，如下图所示：



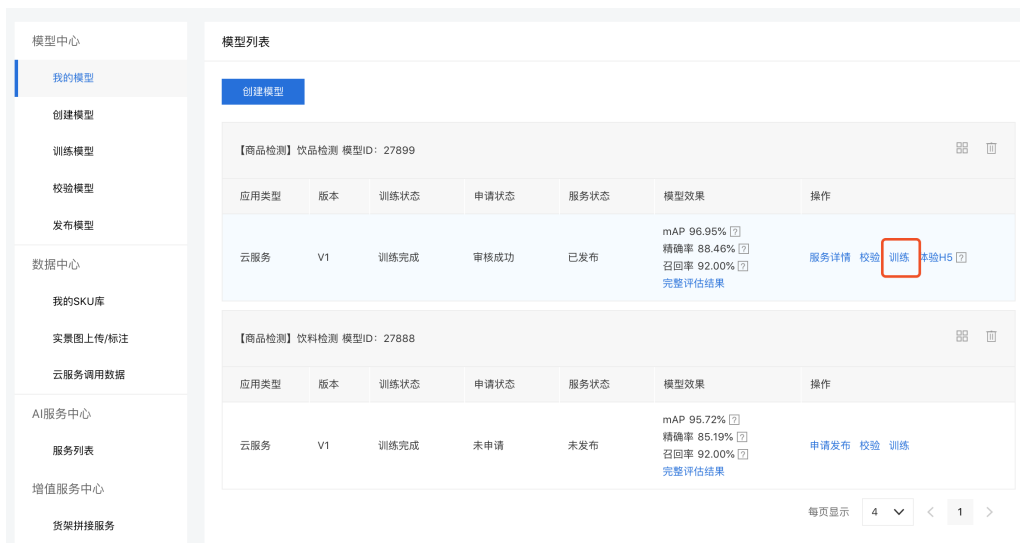
推荐每次将用于优化模型的图片都添加至一个新的实景图集中，可以点击「新建实景图集」新增一个图集，比如新建一个名称叫「优化

0827」，表示用于放8月27日优化模型的图片，这样便于标注和在训练模型时直接勾选上这个实景图集。点击「提交」后，可以点击实景图集的名称立刻跳转去标注页面，也可以点击「继续处理数据」留在该页面继续处理其它的图片，如下图所示：

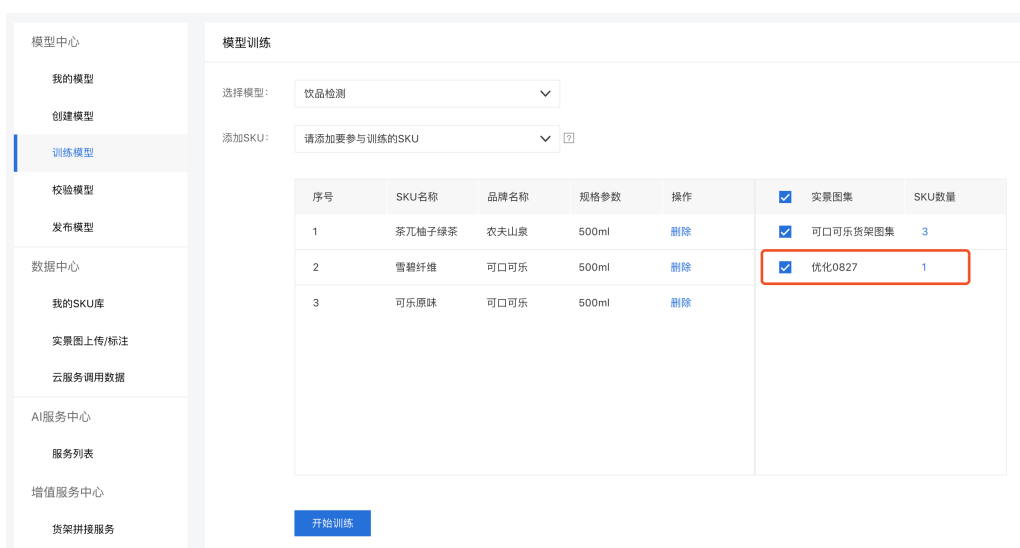


#### 步骤4. 迭代模型

将所有需要修正标注的图片都标注好后，去到[我的模型](#)页，在需要优化的模型表里，点击「训练」，如下图所示：



点击后会跳转至训练页面，如下图所示，新增勾选添加了这部分修正标注的图片的图集即可，比如如果是将这些图片放在实景图集「优化0827」，那么勾选最初训练这个模型的实景图集的同时勾选这个新的图集，确认选择无误后，点击开始训练，训练完成后即完成了一次模型优化。



#### 模型发布

在完成[模型训练](#)后，即可将模型发布为云服务API，发布成功后调用API即可获得模型支持的商品检测能力。模型训练完成后，可以在[我的模型列表](#)中发起模型上线申请，如下图所示：

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V2	训练完成	未申请	未发布	mAP : 93.70% 精确率: 83.33% 召回率: 80.00% 完整评估结果	查看版本详情 <b>申请发布</b> 校验

点击模型列表内对应模型「操作」列中的「申请发布」，或是在左侧导航栏点击「发布模型」可以进入发布模型页面，如下图所示：

模型中心

- 我的模型
- 创建模型
- 训练模型
- 校验模型
- 发布模型**
- 数据中心
- 我的SKU库
- 实景图上传/标注
- AI服务中心
- 服务列表

### 发布模型

- 选择模型: 饮品检测
- 选择版本: V1
- 服务名称: kelexuebijiance
- 接口地址: https://aip.baidubce.com/rpcj2.0/ai\_custom\_retail/v1/detection/kelexuebixianwei
- 其他要求: 可乐和雪碧纤维检测

[提交申请](#)

如果有私有化部署需求, 请点击此申请

#### 标准接口规范参考

标准接口请求参考说明:

字段名称	必须	类型	说明
image	是	string	图像数据, base64编码, 要求base64编码后大小不超过4M, 最短边至少15px, 最长边最大4096px, 支持jpg/png/bmp格式
threshold	否	number	阈值, 默认为当前模型推荐阈值(0-1之间), 具体值可以在我的模型列表-模型效果查看

标准接口响应字段说明:

字段名称	必须	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码, 当请求错误时返回

在对应选项中选择和输入相应内容发起模型发布的申请：

1. 选择模型 (必选)

选择需要发布的模型，只能选择已经完成训练的模型

2. 选择版本 (必选)

选择需要发布的模型版本，只能选择完成训练且没有发布过的版本

3. 服务名称 (必填)

为发布的服务命名，服务名称不得多于20个字符

4. 接口地址 (必填)

自定义服务的API URL，接口地址需要多于5个字符但不能超过20个字符，仅限英文

5. 其他要求

如果有其他要求可以输入要求描述

填写完上述信息后，点击「提交申请」完成发布模型申请。提交申请后，模型列表内该模型的申请状态和服务状态为有以下几种情况：

申请状态	服务状态	状态描述
审核中	未发布	服务刚申请发布，模型在审核中，申请发布后，会自动通过审核
审核成功	发布中	服务通过审核，进入系统自动发布阶段，约5分钟左右完成发布
审核成功	已发布	服务发布成功

发布成功后，可以点击模型列表内「操作」列中的「配置服务功能」，如下图：

模型中心

- 我的模型**
- 创建模型
- 训练模型
- 校验模型
- 发布模型
- 数据中心
- 我的SKU库

### 模型列表

[创建模型](#)

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V1	训练完成	审核成功	已发布	mAP : 96.95% 精确率: 88.46% 召回率: 92.00% 完整评估结果	查看版本详情 <b>配置服务功能</b> 校验 体验H5

点击后弹出下图所示窗口，可以获取模型的云服务API URL。

我的模型 > 1111V1的公有云API服务详情

服务名称: 111  
模型版本: V1  
服务状态: 已发布  
接口地址: [https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/detection/silence02](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/detection/silence02)  
服务功能: 已选功能不同, 单次接口调用价格不同, 详细收费方式请见 [计费文档](#)

基础功能服务

商品基本信息识别  
商品标签 (名称、品牌、规格)、编号、坐标、置信度

可选服务功能

商品陈列层数识别  
商品所在货架层数及货架总层数, 支持区分不同货架  OFF

商品陈列场景识别  
支持五类场景: 货架、端架、立式冰柜、地堆和割箱  ON

商品排面占比统计  
商品排面数/占比、未识别商品数、空位数及货架利用率  ON

提示: 开启或关闭功能后, 约5分钟后生效

[立即使用](#) [返回](#)

服务功能描述:  
模型服务接口使用方法请见 [API文档](#)

服务功能	接口返回字段	内容
商品基本信息识别	name	商品标签, 包含商品名称、品牌、规格
	sku_code	商品编号
	score	识别结果的置信度
	location	商品检测框在图片上的像素坐标
商品陈列场景识别	scenes	图片中包含的陈列场景类型, 支持货架、端架、立式冰柜、地堆和割箱
	scene	每个商品所在的陈列场景
商品陈列层数识别	shelf	商品所在的货架编号, 从左往右依次递增
	layer	商品所在货架层数编号, 从上往下依次递增
	layer_count	统计各货架的总层数
	layer_top	判断货架顶层是否拍摄完整

在该页面可以为模型的云服务API配置服务功能, 详情见[服务功能文档](#)

点击右侧「API文档」可以快速跳转至[API文档](#), 参考文档调用API获取商品检测AI能力。

## 模型使用

### 服务功能

在完成[模型发布](#)后, 可以点击模型列表内「操作」列中的「配置服务功能」, 如下图:

EasyDL零售版 < 模型列表 [操作文档](#) [常见问题](#) [教学视频](#) [新手引导](#) [提交工单](#) [专家定制](#)

我的模型 [创建模型](#)

【商品检测】饮品Demo  模型ID: 30443 [训练](#) ... 更多

部署方式	版本	训练状态	服务状态	SKU F1-score 分布 <sup>②</sup>	操作
公有云API	V1	训练完成	已发布	-	<a href="#">查看版本详情</a> <a href="#">配置服务功能</a> <a href="#">校验</a> <a href="#">体验H5 <sup>②</sup></a>

点击后弹出下图所示窗口, 可以获取模型的云服务API, API使用方式请参考[API调用方法文档](#)。

我的模型 &gt; 饮品DemoV1的公有云API服务详情

操作文档 常见问题 教学视频 新手引导 提交工单 专家定制

服务名称: 111

模型版本: V1

服务状态: 已发布

接口地址: [https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/detection/site\\_nce02](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/detection/site_nce02)

服务功能: 已选功能不同, 单次接口调用价格不同, 详细收费方式请见 [计费文档](#)

基础功能服务

**商品基本信息识别**  
商品标签 (名称、品牌、规格)、编号、像素坐标

可选服务功能 (开关状态变更会改变API返回参数, 详细请见 [API文档](#))

**商品陈列层数识别**  
商品所在货架层数、陈列顺序及货架总层数

**商品陈列场景识别**  
支持十类场景: 货架、端架、立式冰柜、冷风柜、地堆、割箱等

**商品排面占比统计**  
商品排面数/占比、未识别商品数、空位数及货架利用率

**商品陈列翻拍识别**  
识别陈列商品图片是对电子屏幕翻拍的可能性

提示: 开启或关闭功能后, 约5分钟后生效

[立即使用](#) [返回](#)

服务功能描述:

模型服务接口使用方法请见 [API文档](#)

服务功能	接口返回字段	内容
商品基本信息识别	name	商品标签, 包含商品名称、品牌、规格
	sku_code	商品编号
	score	识别结果的置信度
商品陈列场景识别	location	商品检测框在图片上的像素坐标
	scenes	支持十类场景: 货架、端架、立式冰柜、卧式冰柜、冷风柜、挂钩货架、斜口篮货架、地堆、割箱、地龙
	scene	每个商品所在的陈列场景
商品陈列层数识别	shelf	商品所在的货架编号, 从左往右依次递增
	layer	商品所在货架层数编号, 从上往下依次递增
	layer_count	统计各货架的总层数
	layer_top	判断货架最顶层是否拍摄完整
	sku_sn	每一层货架上商品的陈列顺序

在该页面可以为模型的云服务API配置服务功能, 支持以下三项功能:

- **商品基本信息识别 (必选)**  
接口支持识别商品信息 (商品名称、品牌、规格)、编号和置信度
- **商品陈列层数识别 (可选)**  
接口支持识别商品陈列所在货架层数, 货架总层数以及商品的陈列顺序, 货架类型支持: 货架、端架、冷风柜和立式冰柜内货架
- **商品陈列场景识别 (可选)**  
接口支持识别商品陈列的场景, 场景类型支持: 普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台
- **商品排面占比统计 (可选)**  
接口支持统计商品排面数/占比、未识别商品数、总空位数、每货架层的空位数及货架利用率
- **商品陈列翻拍识别 (可选)**  
识别商品陈列照片是对手机屏幕翻拍的可能性

接口单次调用的费用, 根据开启的功能不同而不同, 详情可见[购买指南](#)文档。

可在页面随时开启和关闭可选的功能, 变更功能后约5分钟生效, 生效后单次调用费用按变更后的功能计费, 接口将返回变更后的功能字段, 详情可见[API调用方法](#)文档。

## 🔗 体验H5

### 简介

在完成[模型发布](#)后, 可以将模型云服务API快速集成进H5页面中体验模型效果。

### 生成体验H5页面

可以点击模型列表内「操作」列中的「体验H5」, 如下图:

模型中心

我的模型

创建模型

训练模型

校验模型

发布模型

数据中心

我的SKU库

模型列表

创建模型

【商品检测】 饮品检测 模型ID: 27899

训练 历史版本 删除

部署方式	版本	训练状态	申请状态	服务状态	模型效果	操作
公有云API	V1	训练完成	审核成功	已发布	mAP : 96.95% 精确率: 88.46% 召回率: 92.00% 完整评估结果	查看版本详情 配置服务功能 校验 体验H5

点击后弹出下图所示窗口，选择一个已创建的APPID授权继续。

体验H5说明

H5中的定制化商品检测功能将使用你的app进行调用。

调用APP: EasyDL-APPID: 16096...

温馨提示: 每次体验识别将消耗个人账号下的调用次数。

继续

如果还未在百度智能云创建任何APP，请按照页面弹窗提示，前往[EasyDL零售版控制台应用列表](#)，点击「创建应用」按钮进行创建。

产品服务 / EasyDL定制训练平台 - 应用列表

应用列表

+ 创建应用

应用名称	AppID	API Key	Secret Key	创建时间	操作
silence123	16935698	mbqkk0SDDHxjUuBy2pwVCGO	***** 显示	2019-08-01 18:45:10	报表 管理 删除
Silence-EasyDL	15637143	ghcLQC05QsuIZLEHjxTb48	***** 显示	2019-02-27 11:01:26	报表 管理 删除

创建完成后，即可继续下一步，配置H5相关页面，如下图所示：

自定义H5首页样式

模型名称: 饮品识别

模型介绍: 货架Demo

开发者署名: Silence

H5分享文案: 饮品识别Demo

8/50

生成H5

完成配置后，即可扫描弹框内的二维码进行体验模型服务的效果。

体验H5



用百度或微信APP扫以下二维码，在手机端体验模型效果



[修改已配置的H5页面](#)

注：每次体验识别将消耗个人账号下的调用次数

#### H5页面功能介绍

扫描二维码后，可看到下图页面：



点击上传图片后，可以选择本地相册照片，也可以拍摄照片，上传后，将会使用发布的模型云服务API对图片进行识别，结果如下图所示：





## 1 陈列场景：端架

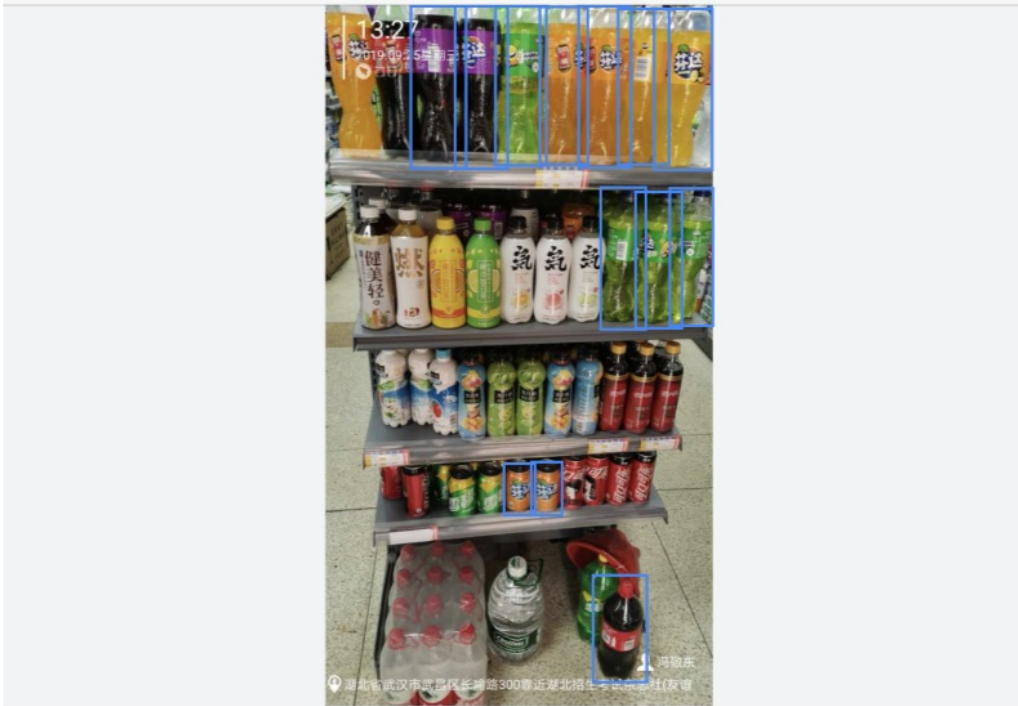
2 置信度  66%

序号	名称	3 数量
1	芬达橙_可口可乐_500ml	4
2	芬达苹果_可口可乐_500ml	4
3	芬达橙_可口可乐_500ml	2
4	美年达葡萄_百事_600ml	2

上传图片

4 查看商品层数

1. 场景识别结果，在[服务功能](#)开启了「商品陈列场景识别」的云服务API的H5页面可以支持返回该结果
2. 置信度过滤设置，根据调整的值，查看图片上的商品识别结果，比如图中设定的66%，置信度大于0.66的识别结果将返回，体现在页面上的是会出现蓝色的检测框
3. 商品数量统计结果，在[服务功能](#)开启了「商品基本信息识别」的云服务API的H5页面可以支持返回该结果
4. 商品数量/层数结果切换按钮，在[服务功能](#)开启了「商品陈列层数识别」的云服务API的H5页面可以支持该选项，点击「查看商品层数」后，可以查看各个商品陈列所在货架层数，显示如下图结果页面：



### 陈列场景：端架

置信度 

 66%

序号	名称	层数
1	芬达橙_可口可乐_500ml	1
2	美年达葡萄_百事_600ml	1
3	芬达橙_可口可乐_500ml	2
4	芬达橙_可口可乐_330ml	4

上传图片

查看商品数量

☁ 云服务API

在完成模型发布后，即可调用云服务API获取商品检测AI能力，可以点击模型列表内「操作」列中的「服务详情」获取API URL，如下图：



点击后弹出下图所示窗口，可以获取模型的云服务API URL。



API使用方式详细介绍请参考[API调用方法文档](#)

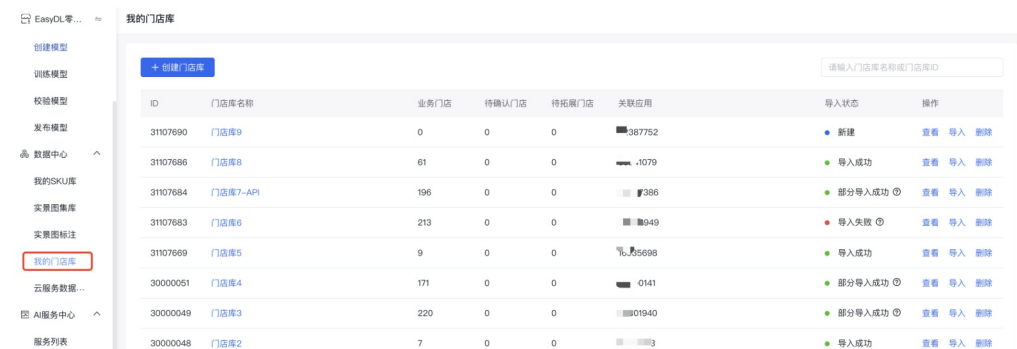
## 门店管理

### 门店库创建

本文档介绍门店管理功能中的门店库创建流程。

### 在页面创建门店库

在左侧导航栏点击「我的门店库」进入[门店管理页面](#)，如下图所示



点击「创建门店库」按钮，在弹框中按要求输入信息

### 创建门店库

\* 门店库名称

\* 控制台应用

选择的应用用于关联门店库, 如未创建, 请前往 [控制台](#) 进行创建

- 门店库名称：定义门店库的名称，20字以内，创建后可在门店库列表中修改
- 控制台应用：选择在[百度云EasyDL零售版控制台](#)创建的应用，用于关联门店库，如未创建，请前往创建后再进行创建

门店库关联应用说明：

一个门店库可以关联多个应用，一个应用只能关联一个底库

按要求完成填写后，点击「创建」按钮即可完成门店库的创建，创建的门店库可在门店库列表中看到。

## 门店导入

本文档介绍门店管理功能中的门店导入流程。

### 门店导入

创建好门店库后，在列表中「操作」列点击「导入」，可进入门店导入页面。

[< 返回](#) 导入门店

#### 门店库信息

门店库名称：	门店库9	门店库ID：	31107690	关联应用：	32387752
业务门店数：	-	待确认门店数：	-	待拓展门店数：	-

#### 导入门店

门店类型  业务门店  待确认门店

导入方式  批量导入  单店导入

#### 1. 门店类型

##### 业务门店

业务门店指的是客户业务系统中真实的业务门店，进入门店导入页面后，在「门店类型」选项中选择「业务门店」即为指定将门店导入业务门店列表中。

##### 待确认门店

待确认门店指的是待确认的未铺货门店，如百度智能拓店服务输出待拓展门店数据，在「门店类型」选项中选择「待确认门店」即为指定将门店导入业务门店列表中。

#### 2. 导入方式

##### 批量导入

该方式适用于将门店多条门店数据导入至门店库中，在「导入方式」选项中选择「批量导入」，按页面所提示内容将本地的门店文件上传至平台，点击「确认」即可开始导入。

< 返回 导入门店

**门店库信息**

门店库名称: 门店库9      门店库ID: 31107690      关联应用: 32387752  
 业务门店数: -      待确认门店数: -      待拓展门店数: -

---

**导入门店**

门店类型  业务门店  待确认门店

导入方式  批量导入  单店导入

\* 选择本地文件

文件要求说明 [下载示例文件](#)

- 支持xlsx和xls格式的Excel文件，以及CSV文件(分隔符为英文逗号)
- 文件编码支持:UTF-8，可参考 [编码查看和修改方法文档](#)
- 文件必须包含以下表头，各列的内容规范如下:

表头名称	是否必填	内容规范
user_store_id	否	用户业务系统中的真实的业务门店ID，长度限制128B，支持字母、数字、下划线
store_name	是	门店名称，长度限制1024B

### 单店导入

该方式适用于将单条门店数据导入至门店库中，在「导入方式」选项中选择「单店导入」，在页面上按要求完成各项目填写，点击「确认」即可开始导入。

< 返回 导入门店

**门店库信息**

门店库名称: 门店库9      门店库ID: 31107690      关联应用: 32387752  
 业务门店数: -      待确认门店数: -      待拓展门店数: -

---

**导入门店**

门店类型  业务门店  待确认门店

导入方式  批量导入  单店导入

业务门店ID

\* 门店名称

\* 门店经度

\* 门店纬度

\* 坐标类型

### 门店导入状态

在门店导入页面点击「确认」后，可在门店库列表中查看门店导入的状态。

ID	门店库名称	业务门店	待确认门店	待拓展门店	关联应用	导入状态	操作
31107690	门店库9	0	0	0	32387752	<span style="color: blue;">●</span> 新建	<a href="#">查看</a> <a href="#">导入</a> <a href="#">删除</a>
31107686	门店库8	61	0	0	33941079	<span style="color: green;">●</span> 导入成功	<a href="#">查看</a> <a href="#">导入</a> <a href="#">删除</a>
31107684	门店库7-API	196	0	0	33857386	<span style="color: green;">●</span> 部分导入成功 <input type="radio"/>	<a href="#">查看</a> <a href="#">导入</a> <a href="#">删除</a>
31107683	门店库6	213	0	0	33856949	<span style="color: red;">●</span> 导入失败 <input type="radio"/>	<a href="#">查看</a> <a href="#">导入</a> <a href="#">删除</a>

状态说如下：

- **新建**：新创建的门店库，未进行任何门店导入操作
- **导入中**：门店导入过程中的状态
- **导入成功**：所有门店都导入成功的状态
- **部分导入成功**：存在部分门店导入失败的状态，可在？内图标里查看详情，也可点击下载失败详情文件到本地
- **导入失败**：所有门店都导入失败的状态

## 门店查看和编辑

本文档介绍门店管理功能中的门店查看功能。

### 门店库详情

门店导入完成后，在门店库列表「操作」列点击「查看」即可进入门店库详情页面，页面包含门店库基本信息和门店列表。

< 返回 门店库详情
导入

**门店库信息**

门店库名称: 门店库2	门店库ID: 30000048	关联应用: 15637143
业务门店数: 7	待确认门店数: -	待拓展门店数: -

业务门店 待确认门店 待拓展门店

全部省 / 全部市
全部渠道

业务门店ID	门店名称	经度	纬度	坐标类型	操作
10002	晨晨超市	116.312803	40.047735	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
10003	绿丰果蔬超市	118	40	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
10004	王一烧烤	114	34	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
10005	利华平价超市	115	35	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
10006	盛源酒楼	116	36	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
10007	太河全羊馆	117.1	37.2	BD09II	<a href="#">查看/编辑</a> <a href="#">删除</a>
store_16869062714978_c91bd6cb1e1b	四川特产文化市集	121.54409	31.22114	GCJ02II	<a href="#">查看/编辑</a> <a href="#">删除</a>

### 门店信息

在门店列表的「操作」列中点击「查看/编辑」按钮，进入门店详情页。门店信息会展示门店导入时填入的内容，非必填项如果未填写，则会显示「-」。

**门店信息**

编辑

业务门店ID: 10002	待确认门店ID: -	门店名称: 晨晨超市
经度: 116.312803	纬度: 40.047735	坐标类型: BD09II
省份: -	城市: -	区县: -
地址: -	渠道类型: -	连锁类型: -
潜力评分: -		

点击「编辑」按钮，即可以对门店信息进行修改。

## 编辑门店



## 门店信息

业务门店ID	10002
待确认门店ID	定义待确定门店的唯一ID
* 门店名称	晨晨超市
* 门店经度	116.312803
* 门店纬度	40.047735
* 坐标类型	BD09II
省份	门店所在省份，如江苏省
城市	门店所在城市，如南京市
区县	门店所在区/县，如秦淮区
地址	门店所在地址，如中山南路1号新百商场2楼
渠道类型	门店所属渠道类型，如 超市

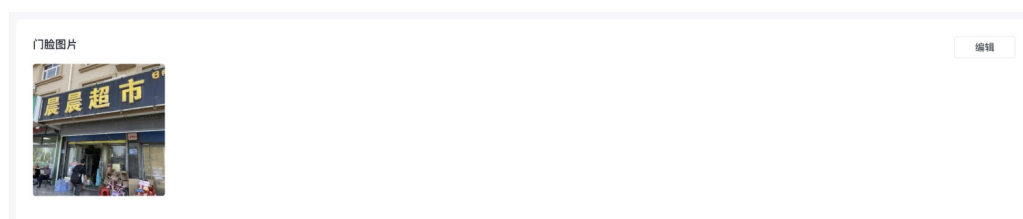
取消

确定

## 门脸图片

门脸图片为对应门店的店招图，图片要求如下：

- 1.至少上传1张图片，最多上传5张图片，建议为多角度拍摄的门脸图片
- 2.支持png、jpg、jpeg格式
- 3.图片最短边不小于15px，最长边不大于4096px



点击编辑，即可对已上传的门脸图片进行修改：

## 编辑门店图片



## 门脸图片



## 图片要求：

- 1.至少上传1张图片，最多上传5张图片，建议为多角度拍摄的门脸图片
- 2.支持png、jpg、jpeg格式
- 3.图片最短边不小于15px，最长边不大于4096px

取消

确定

注意：门脸图片修改后，保存修改内容需要一定时间，修改过程中，图片不能被修改，页面显示不会实时更新，可刷新页面观察是否完成修改，修改完成后，可再对图片进行修改。

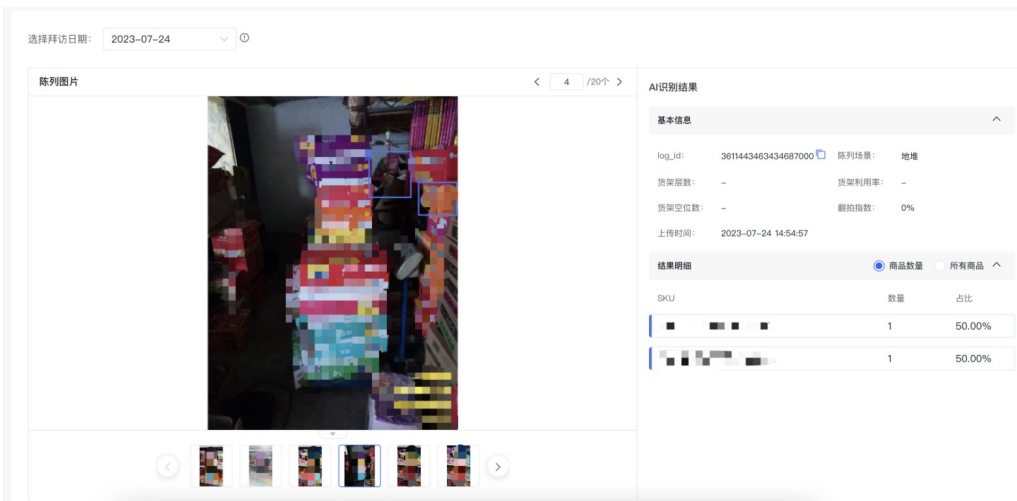
## 商品陈列信息

如果用户同时使用定制商品检测服务的云服务API，可在请求参数中传入门店ID，以建立商品检测识别到的陈列信息和门店信息的关联，建立后，可在此部分查看该门店最近一次被拜访时的商品陈列信息。

商品陈列信息 ⓘ		最近拜访日期: 2023-07-24		详情
商品分销量:	5	陈列场景:	地堆、货架、卧式冰柜	

点击「详情」按钮后，可进入详情页，支持查看最近30天内该门店调用定制商品检测服务API时上传的照片和对应的识别结果。

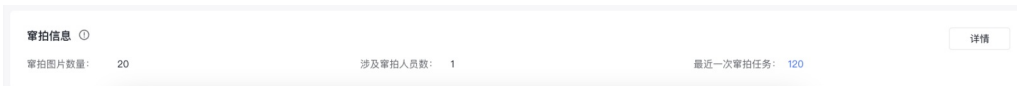




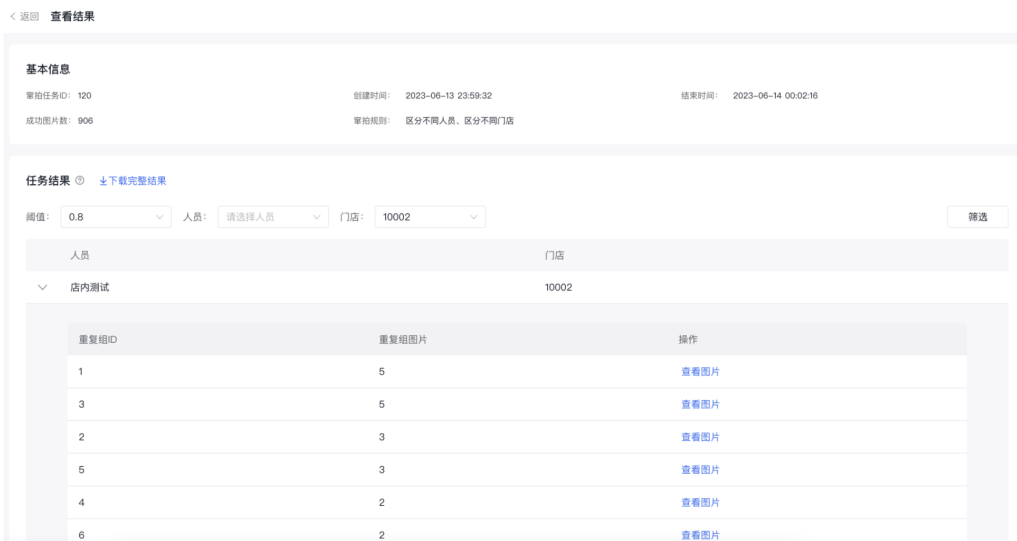
注意：通过定制商品检测服务的云服务API获取的图片可能会有缺失。

### 窜拍信息

如果用户同时使用窜拍识别服务，可在窜拍任务中定义门店ID，以建立窜拍和门店信息的关联，建立后，可在此部分查看该门店最近一次窜拍任务所设定最大阈值的消息。



点击「详情」按钮后，可进入窜拍任务详情页，查看窜拍任务具体的结果。

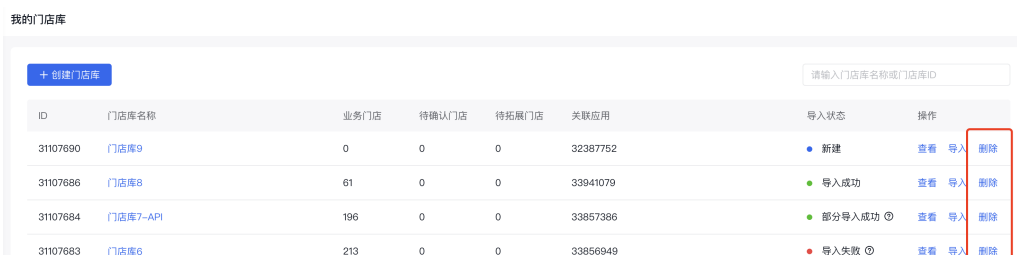


### 门店库和门店删除

该文档介绍如何删除门店库和门店。

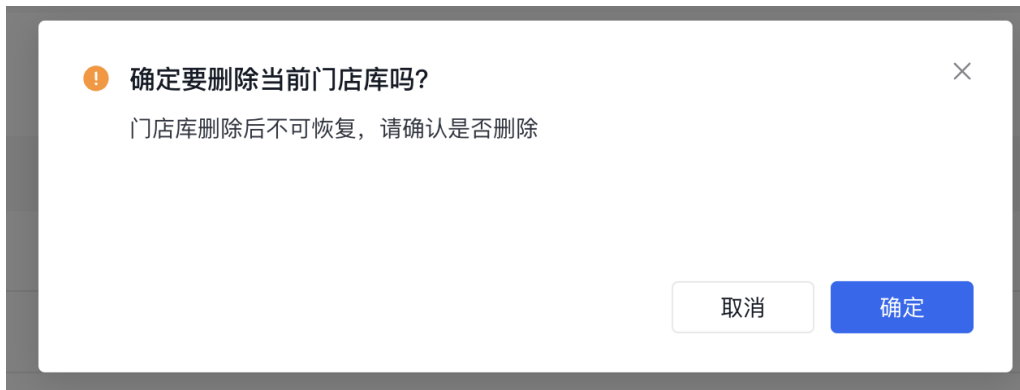
#### 删除门店库

进入[我的门店库](#)页面。



在门店库列表「操作」列点击「删除」，在弹窗中点击「确认」后即可将门店库删除。

注意：门店库删除后不可恢复，请谨慎操作



## 删除门店

进入我的门店库页面。

我的门店库

+ 创建门店库

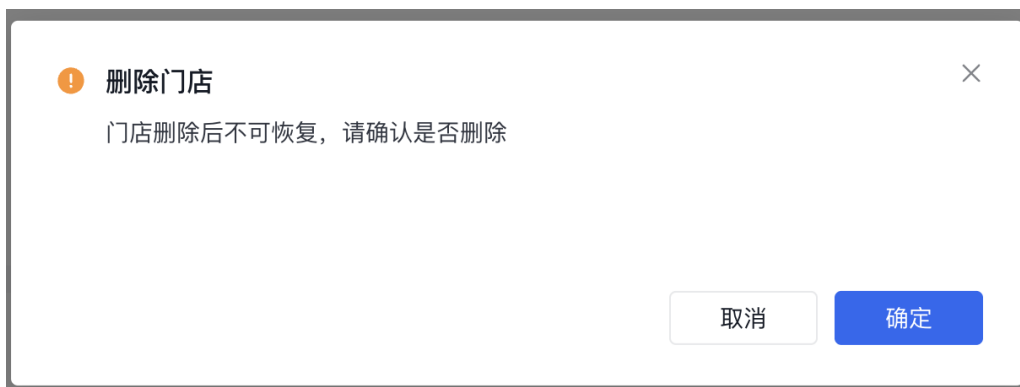
ID	门店库名称	业务门店	待确认门店	待拓展门店	关联应用	导入状态	操作
31107690	门店库9	0	0	0	32387752	● 新建	查看 导入 删除
31107686	门店库8	61	0	0	33941079	● 导入成功	查看 导入 删除
31107684	门店库7-API	196	0	0	33857386	● 部分导入成功	查看 导入 删除
31107683	门店库6	213	0	0	33856949	● 导入失败	查看 导入 删除

在门店库列表「操作」列点击「查看」，进入门店库详情页面。

业务门店ID	门店名称	经度	纬度	坐标类型	操作
10002	晨晨超市	116.312803	40.047735	BD09II	查看/编辑 删除
10003	绿丰果蔬超市	118	40	BD09II	查看/编辑 删除
10004	王一烧烤	114	34	BD09II	查看/编辑 删除
10005	利华平价超市	115	35	BD09II	查看/编辑 删除
10006	盛源酒楼	116	36	BD09II	查看/编辑 删除

在门店列表「操作」列点击「删除」，在弹窗中点击「确认」后即可将门店库删除。

注意：门店删除后不可恢复，请谨慎操作



## API文档

### 商品检测API调用方法

#### 接口描述

本文档主要说明定制化商品检测模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择人工智能服务
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

## 接口鉴权

1. 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权参考文档](#)，使用API Key(AK)和Secret Key(SK)获取 access\_token

## 请求说明

## 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL零售版](#)进行定制商品检测模型训练，完成训练后申请上线，上线成功后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001和336002的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

## 请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
threshold	否	number	0~1	可精确到小数点后两位，默认值为建议阈值，请在 <a href="#">我的模型列表-完整评估结果</a> 查看推荐阈值。阈值(threshold)，是正确结果的判定标准，例如阈值是0.6，置信度大于0.6的识别结果会被当作正确结果返回。
split_shelf	否	boolean	True/False	True表示区分货架节数，False表示不区分货架节数。默认为False。当未开通货架层数识别服务功能时，传该参数不生效。

提示：image参数中“去掉头部”指的是图片经base64编码后的头部信息「data:image/jpeg;base64,」，如下图所示：

点击这里选择要转换成Base64的图片

C:\fakepath\14458.jpg

复制

清空

```
data:image/jpeg;base64,[9j/4RCRRXhpZgAATU0AKgAAAQAEQEAAAMAAABDYAAAAEBAAMAAABEgAAAAECAAMAA
AADAAAA2gEGAAMAAABAAIAAAEOAAIAAAAEbm9yAAEPAAIAAAAHAAAA4AEQAIAAAAJAAAA5wESAAMAAABAAEAA
AEVAMAAABAAAMAAEAUAUAAAABAAA8AEbAAUAAAABAAAA+AEoAMAAABAAIAAAEAAIAAAKAAABAAEYAAIAA
AAUAAABJAITAAMAAABAAEAAlpAAQAAAAABAAABOlglAAQAAAAABAAADeAAABFgACAAIAAhIVUFXRUKASFdJLUFMMDA
AAAr8gAAAJxAACvyAAANeEFkb2JlIFBob3Rvc2hvcCBDQyAyMDE3IChNYWNpbnRvc2gpADlwMTk6MDQ6MTQgMjE6MTU
6MiaAACWCmaFAAAAAQAAAvCnQAFAAAAQAAAwKllaDAAAAAQACAACIjwADAAAAAQCaAACQAAAHAAABDAvMT
```

## 返回说明

排面数：同层同列去重后的SKU检测数量

开启模型服务功能可参考[服务功能文档](#)

## 返回参数

返回结果为JSON格式

字段	是否必选	类型	说明	需要开启的模型服务功能
recapture_score	否	float	图片是对手机屏幕翻拍的可能性评分。翻拍判定方法：设定一个判定为翻拍图片的阈值，即如果recapture_score大于这个值，则认为这张图片是翻拍。请结合业务实际情况和实测结果进行设定阈值	商品陈列翻拍识别
statistics	否	object{}	对于整张图片的综合识别统计结果	商品排面占比统计
+known_sku_num	否	int	定制模型识别的SKU总数量	商品排面占比统计
+unknown_sku_num	否	int	定制模型未识别的SKU总数量	商品排面占比统计
+known_sku_facing	否	int	定制模型识别的SKU总排面数量，如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
+unknown_sku_facing	否	int	定制模型未识别的SKU总排面数量，如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
+vacancy_num	否	int	货架上的空位数量，如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
+share_of_shelf	否	float	定制商品检测模型识别SKU的排面占比= $(\text{known\_sku\_facing})/(\text{known\_sku\_facing}+\text{unknown\_sku\_facing})$ 。 如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
+known_sku_proportion	否	float	定制商品检测模型识别SKU的数量占比= $(\text{known\_sku\_num})/(\text{known\_sku\_num}+\text{unknown\_sku\_num})$	商品排面占比统计

+utilization	否	float	货架利用率= $(\text{known\_sku\_num})/(\text{known\_sku\_num}+\text{unknown\_sku\_num}+\text{vacancy\_num})$ 。 如果图片为非货架陈列场景（如冰柜、端架、普通货架等），结果不具参考意义	商品排面占比统计
sku_count	否	array[object]	定制商品检测模型识别的各类SKU的总数和排面占比	商品排面占比统计
+name	否	string	定制商品检测模型识别的SKU标签，SKU名称_品牌名称_规格参数，为在EasyDL零售版上创建SKU时填写的内容	商品排面占比统计
+sku_code	否	string	定制商品检测模型识别的SKU编码，为在EasyDL零售版上创建SKU时填写的内容	商品排面占比统计
+sku_num	否	int	定制商品检测模型识别的各类SKU的总数	商品排面占比统计
+proportion	否	float	定制商品检测模型识别的各类SKU的数量占比= $\text{sku\_num}/(\text{known\_sku\_num}+\text{unknown\_sku\_num})$	商品排面占比统计
+sku_facing	否	int	定制商品检测模型识别的各类SKU的总排面数	商品排面占比统计
+sku_scores	否	float	定制商品检测模型识别的各类SKU的排面数量占比= $\text{sku\_facing}/(\text{known\_sku\_facing}+\text{unknown\_sku\_facing})$ 。 如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
shelf_info	否	array[object]	各组货架的每层货架的详细统计信息，如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品排面占比统计
+shelf	否	int	商品所在货架编号，“-1”表示未识别到货架，编号为图中货架最左从往右数依次增大	商品排面占比统计
+layer_info	否	array[object]	各货架层的详细空位数信息	商品排面占比统计
++layer	否	int	商品所在层数编号，“-1”表示未识别到层数，编号为从图中货架最上层往下依次增大	商品排面占比统计
++layer_vacancy	否	int	每一层的空位数量	商品排面占比统计
++layer_known_sku_num	否	int	每一层的可识别SKU数量	商品排面占比统计
++layer_unknown_sku_num	否	int	每一层的未知SKU数量	商品排面占比统计
layer_count	否	array[object]	图片中，各货架的总层数，如果图片为非货架陈列场景（如堆箱、割箱、地龙等），结果不具参考意义	商品陈列层数识别

+shelf	否	int	商品所在货架编号, "-1"表示未识别到货架, 编号为图中货架最左从往右数依次增大	商品陈列层数识别
+layer_num	否	int	货架的总层数, 如果图片为非货架陈列场景 (如冰柜、端架、普通货架等), 结果不具参考意义	商品陈列层数识别
layer_top	否	int	判断是否拍摄到货架最上一层, 0表示未拍摄到, 1表示拍摄到, -1表示图片中未识别到货架	商品陈列层数识别
layer_complete	否	int	表面货架是否拍摄完整, 0表示不完整, 1表示完整	商品陈列层数识别
scenes	否	array[string]	图片中包含的陈列场景类型。返回所有场景去重后的集合, "GE"表示端架, "shelf"表示货架, "freezer"表示冰柜, "TG"表示地堆, "cutbox"表示割箱, "DL"表示地龙, "HS"表示挂钩货架, "OBS"表示斜口篮货架, "SGE"表示小端架, "barcounter"表示吧台, "HF"表示卧式冰柜, "OACR"表示冷风柜, "HGE"表示挂钩端架, "unknown"表示未识别到场景	商品陈列场景识别
log_id	是	int	唯一的log id, 用于问题定位	商品基本信息识别
results	否	array[object]	图片中每个商品的详细信息	-
+name	否	string	SKU名称_品牌名称_规格参数	商品基本信息识别
+scene	否	string	表示该SKU所在的陈列场景。"GE"表示端架, "shelf"表示货架, "freezer"表示冰柜, "TG"表示地堆, "cutbox"表示割箱, "DL"表示地龙, "HS"表示挂钩货架, "OBS"表示斜口篮货架, "SGE"表示小端架, "barcounter"表示吧台, "HF"表示卧式冰柜, "OACR"表示冷风柜, "HGE"表示挂钩端架, "unknown"表示未识别到场景	商品陈列场景识别
+shelf	否	int	商品所在货架编号, "-1"表示未识别到货架, 编号为图中货架最左从往右数依次增大, 如果图片为非货架陈列场景 (如堆箱、割箱、地龙等), 结果不具参考意义	商品陈列层数识别
+layer	否	int	商品所在层数编号, "-1"表示未识别到层数, 编号为从图中货架最上层往下依次增大, 如果图片为非货架陈列场景 (如堆箱、割箱、地龙等), 结果不具参考意义	商品陈列层数识别
+sku_sn	否	string	商品的陈列排序序号, 返回"A-B"或"A-B-C", 如"2-1"或"2-1-1", 其中A、B、C分别为数字, A代表同一货架层的横向顺序, 从左至右依次增大; B代表同一货架的层纵向序号, 从下至上依次增大; 如果存在大商品上下陈列有小商品或包含小商品的情况, 会出现C, 从左至右依次增大。如果图片为非货架陈列场景 (如堆箱、割箱、地龙等), 结果不具参考意义	商品陈列层数识别
+sku_code	否	string	商品编号, 由用户在模型训练页面创建SKU时自定义	商品基本信息识别
+score	否	float	置信度	商品基本信息识别
+location	否	object{}	每个商品在图上的像素位置	商品基本信息识别
++left	否	int	检测到的目标主体区域到图片左边界的像素距离	商品基本信息识别
++top	否	int	检测到的目标主体区域到图片上边界的像素距离	商品基本信息识别

				识别
++width	否	int	检测到的目标主体区域的像素宽度	商品基本信息识别
++height	否	int	检测到的目标主体区域的像素高度	商品基本信息识别

### 建议翻拍判定方法

设定一个判定为翻拍图片的阈值，即如果recapture的score大于这个值，则认为这张图片是翻拍。通常有两中对应的业务模式：

注：以下数值均为建议值，实际应用的阈值请结合业务实际情况和实测结果进行设定

1. 业务里查翻拍的原则是宁可错杀一千，也不愿错放一个的，那么可以把认为是翻拍的阈值放在0.8~0.95。
2. 业务里查翻拍的原则是允许错放过一些翻拍的图片，但是查到的一定要，那么可以把认为是翻拍的阈值放在0.98甚至0.99。

### 地堆检测API调用方法

#### 接口描述

本文档主要说明定制化商品检测模型发布后获得的API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择人工智能服务
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

#### 接口鉴权

1. 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权参考文档](#)，使用API Key(AK)和Secret Key(SK)获取access\_token

#### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：请首先在[EasyDL零售版](#)进行定制商品检测模型训练，完成训练后申请上线，上线成功后可在服务列表中查看并获取url。

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001和336002的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
threshold	否	number	0~1	可精确到小数点后两位，默认值为建议阈值，请在 <a href="#">我的模型</a> 列表-完整评估结果 查看推荐阈值。阈值（threshold），是正确结果的判定标准，例如阈值是0.6，置信度大于0.6的识别结果会被当作正确结果返回。

提示：image参数中“去掉头部”指的是图片经base64编码后的头部信息「data:image/jpeg;base64,」，如下图所示：

C:\fakepath\14458.jpg

data:image/jpeg;base64,9j/4RCRRXhpZgAATU0AKgAAAQgAEQEAAAMAAAABDYAAAAEBAAMAAAABEgAAAAECAAMAAADAAAA2gEGAAMAAAABAAIAAAEOAAIAAAAEbm9yAAEPAAIAAAAHAAAA4AEQAIAAAAJAAAA5wESAAMAAAABAAEAAAEVAAMAAAABAAAMAAAEaAAUAAAABAAA8AEbAAUAAAABAAAA+AEoAAMAAAABAAIAAAExAAIAAAkAAABAAEyAAIAAAUAAAABJAITAAMAAAABAAEAAIdpAAQAAAAABAAABOIgIAAQAAAAABAAADeAAABFgACAAIAAhVUFXRUKASFdJLUFMMMDAAAAR8gAAAjxAACvyAAAAnEEFkb2JlIFBob3Rvc2hvcCBDQyAyMDE3IChNYWNpbmRvc2gpADlwMTk6MDQ6MTQgMjE6MTU6MTU6AACWCmaFAAAAAQAAAVaCnQAFAAAAQAAAwKIIaADAAAAAQACAAClJwADAAAAQCaAACQAAAHAAAABDAvMT

返回说明

开启模型服务功能可参考[服务功能文档](#)

排面数：同层同列去重后的SKU检测数量

返回参数

字段	是否必选	类型	说明	需要开启的模型服务功能
recapture_score	否	float	图片是对手机屏幕翻拍的可能性评分。翻拍判定方法：设定一个判定为翻拍图片的阈值，即如果recapture_score大于这个值，则认为这张图片是翻拍。请结合业务实际情况和实测结果进行设定阈值	商品陈列翻拍识别（可选）
log_id	是	number	唯一的log id，用于问题定位	商品基本信息识别（必选）
scenes	否	array(string)	图片中包含的陈列场景类型。返回所有场景去重后的集合，“GE”表示端架，“shelf”表示货架，“freezer”表示冰柜，“TG”表示地堆，“cutbox”表示割箱，“DL”表示地龙，“HS”表示挂钩货架，“OBS”表示斜口篮货架，“SGE”表示小端架，“barcounter”表示吧台，“unknown”表示未识别到场景	商品陈列场景识别（可选）
results	否	array(object)	识别结果数组	商品基本信息识别（必选）



	否	number	堆头的序号，从左至右依次增加。举例：如果模型检测出图片存在两个堆头，那边两个堆头从左至右的stack_sn分别为1和2	商品基本信息识别（必选）
+stack_info	否	array(object)	各个堆头的检测结果	商品基本信息识别（必选）
++type	否	number	堆头陈列的类型：1为KA可识别长宽；2为非KA可计数；-1为其它（不可计数和不可计长宽）	商品基本信息识别（必选）
++stack_height	否	number	该堆头的高度（Y轴数量）	商品基本信息识别（必选）
++stack_width	否	number	该堆头的宽度（列数，X轴数量）	商品基本信息识别（必选）
++stack_depth	否	number	该堆头的纵深（Z轴数量）	商品基本信息识别（必选）
++area	否	number	该堆头的占地面数量 = width * depth	商品基本信息识别（必选）
++stack_sku_num	否	number	该堆头中的可识别SKU的含纵深总数量	商品基本信息识别（必选）
++stack_sku_num_visible	否	number	该堆头中的可识别SKU的可见总数量	商品基本信息识别（必选）
++stack_sku_info	否	array(object)	堆箱内含有SKU的信息	商品基本信息识别（必选）
+++name	否	string	SKU标签名称	商品基本信息识别（必选）
+++sku_code	否	string	SKU编码	商品基本信息识别（必选）
+++num_with_dept	否	number	该类SKU在该堆头中的含纵深总数量	商品基本信息识别（必选）
+++num_without_dept	否	number	该类SKU在该堆头中的可见总数量	商品基本信息识别（必选）
+col_info	否	array(object)	各个堆头中，各列的检测结果	商品基本信息识别（必选）
++col_loc_info	否	array(object)	各列的深（Z轴）和高（Y轴）信息	商品基本信息识别（必选）
+++col_sn	否	array(object)	列的序号，从左至右依次增加	商品基本信息识别（必选）

+++col_depth	否	number	列的纵深 (Z轴数量) SKU数量	商品基本信息识别 (必选)
+++col_height	否	number	列的高度 (Y轴数量) SKU数量	商品基本信息识别 (必选)
++col_num_info	否	array(object)	各列中各类SKU信息和数量结果	商品基本信息识别 (必选)
+++col_sn	否	number	列的序号, 从左至右依次增加	商品基本信息识别 (必选)
+++col_sku_num	否	number	该列的可识别SKU的含纵深总数量	商品基本信息识别 (必选)
+++col_sku_num_visible	否	number	该列的可识别SKU的可见总数量	商品基本信息识别 (必选)
+++col_sku_info	否	array(object)	列内各类SKU的信息	商品基本信息识别 (必选)
++++name	否	string	SKU标签名称	商品基本信息识别 (必选)
++++sku_code	否	string	SKU编码	商品基本信息识别 (必选)
++++num_with_depth	否	number	该类SKU在该列中的含纵深总数量	商品基本信息识别 (必选)
++++num_without_depth	否	number	该类SKU在该列中的可见总数量	商品基本信息识别 (必选)
++s_col_info	否	array(object)	各列中, 各子列的检测结果, 每个纵深 (Z轴) 列为一个子列	商品基本信息识别 (必选)
+++col_sn	否	number	列的序号, 从左至右依次增加	商品基本信息识别 (必选)
+++s_col_sn	否	number	子列的序号, 从内到外, 依次增加	商品基本信息识别 (必选)
+++s_col_sku_num	否	number	该列的可识别SKU纵深总数量	商品基本信息识别 (必选)
+++s_col_sku_num_visible	否	number	该列的可识别SKU可见总数量	商品基本信息识别 (必选) (必选)
+++s_col_sku_info	否	array(object)	当前子列内包含SKU的信息	商品基本信息识别 (必选)

++++name	否	string	SKU标签名称	商品基本信息识别 (必选)
++++sku_code	否	string	SKU编码	商品基本信息识别 (必选)
++++num_with_depth	否	number	该类SKU在该子列中的纵深总数量	商品基本信息识别 (必选)
++++num_without_depth	否	number	该类SKU在该子列中的可见总数量	商品基本信息识别 (必选)
+sku_info	否	array(object)	该堆头中的所有商品信息	商品基本信息识别 (必选)
++name	否	number	SKU标签名称	商品基本信息识别 (必选)
++sku_code	否	number	SKU编码	商品基本信息识别 (必选)
++scene	否	string	该SKU的陈列场景，“GE”表示端架，“shelf”表示货架，“freezer”表示冰柜，“TG”表示地堆，“cutbox”表示割箱，“DL”表示地龙，“HS”表示挂钩货架，“OBS”表示斜口篮货架，“SGE”表示小端架，“barcounter”表示吧台，“unknown”表示未识别到场景	商品陈列场景识别 (可选)
++score	否	number	置信度	商品基本信息识别 (必选)
++location	否	array(object)	该SKU在原图中的像素位置信息	商品基本信息识别 (必选)
+++left	否	number	检测到的目标主体区域到图片左边界的像素距离 (px)	商品基本信息识别 (必选)
+++top	否	number	检测到的目标主体区域到图片上边界的像素距离 (px)	商品基本信息识别 (必选)
+++width	否	number	检测到的目标主体区域的像素宽度 (px)	商品基本信息识别 (必选)
+++height	否	number	检测到的目标主体区域的像素高度 (px)	商品基本信息识别 (必选)
++threed_location	否	array(object)	该SKU在堆头中的三维位置信息	商品基本信息识别 (必选)
+++sku_height	否	number	SKU所在堆头中的3D位置，Y轴高度 (从下往上数第几个SKU)	商品基本信息识别 (必选)
+++sku_width	否	number	SKU所在堆头中的3D位置，X轴宽度 (从左往右数第几个SKU)	商品基本信息识别 (必选)

+++sku_depth	否	number	SKU所在堆头中的3D位置，Z轴深度（从里往外数第几个SKU）	商品基本信息识别（必选）
--------------	---	--------	---------------------------------	--------------

### 建议翻拍判定方法

设定一个判定为翻拍图片的阈值，即如果recapture的score大于这个值，则认为这张图片是翻拍。通常有两中对应的业务模式：

注：以下数值均为建议值，实际应用的阈值请结合业务实际情况和实测结果进行设定

1. 业务里查翻拍的原则是宁可错杀一千，也不愿错放一个的，那么可以把认为是翻拍的阈值放在0.8~0.95。
2. 业务里查翻拍的原则是允许错放过一些翻拍的图片，但是查到的一定要正确，那么可以把认为是翻拍的阈值放在0.98甚至0.99。

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请通过QQ群（1009661589）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数或代码格式有误。有疑问请通过QQ群（1009661589）或工单联系技术支持团队
336003	Base64解码失败	图片格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请通过QQ群（1009661589）或工单联系技术支持团队
336004	输入文件大小不合法	图片或音频、文本格式有误，请根据接口文档检查入参格式，有疑问请通过QQ群（1009661589）或工单联系技术支持团队
336005	解码输入失败/分词错误	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（1009661589）或工单联系技术支持团队

## 翻拍识别服务

### 服务介绍

#### 简介

商品陈列翻拍识别能够识别出通过手机翻拍出的商品陈列照片，比如商品货架陈列图片和地堆商品陈列图片，可降低人工审核人力，高效审核零售业务中通过翻拍原有图片来造假的图片。

#### 适用场景

适用于识别以下类型的商品陈列翻拍图片：

- 货架/货柜上商品陈列图片
- 地堆商品陈列图片

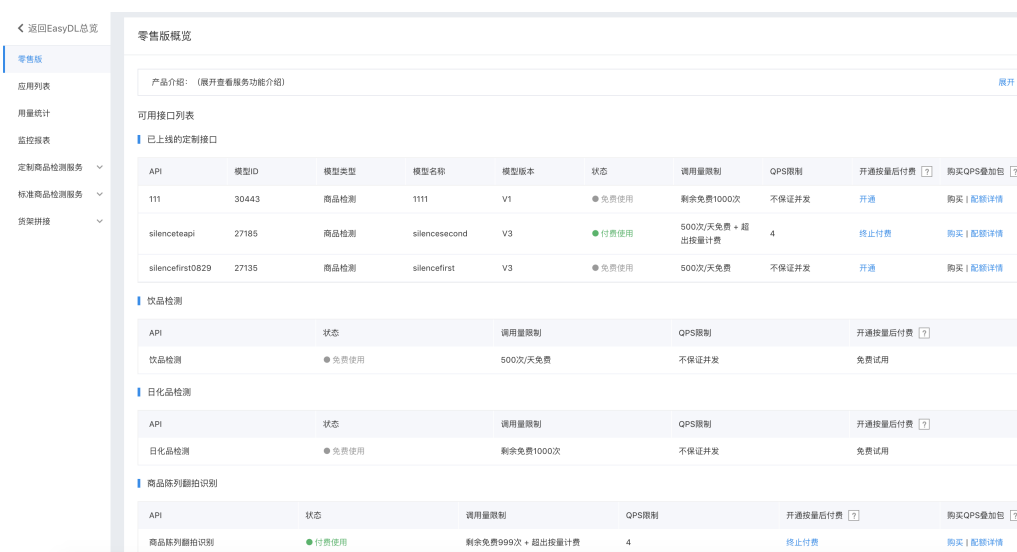
使用须知

- 服务价格详情请见[购买指南](#)
- 服务接口调用方法请见[API文档](#)

购买指南

开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。



计费方式

商品陈列翻拍识别目前支持下列三种计费方式:

1. 按调用量后付费
2. 调用量次数包预付费
3. QPS叠加包预付费

价目表 - 按调用量后付费

付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

## 价目表 - 调用量次数包

如果业务上对调用次数有预估，可以选择购买**单次调用价格更低**的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	490 元	4	1年 (366天)
10万次	4,800 元	4	1年 (366天)
100万次	45,000 元	4	1年 (366天)
500万次	212,500 元	4	1年 (366天)
1000万次	420,000 元	4	1年 (366天)
2000万次	800,000 元	4	1年 (366天)

**购买后不可退款**，次数包使用完后，开始按调用量每次0.05元收取费用

**特殊说明**，此计费方式仅限于单独调用翻拍模型接口，定制商品检测服务接口中的翻拍服务的计费不适用

## 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## API文档

### API调用方法

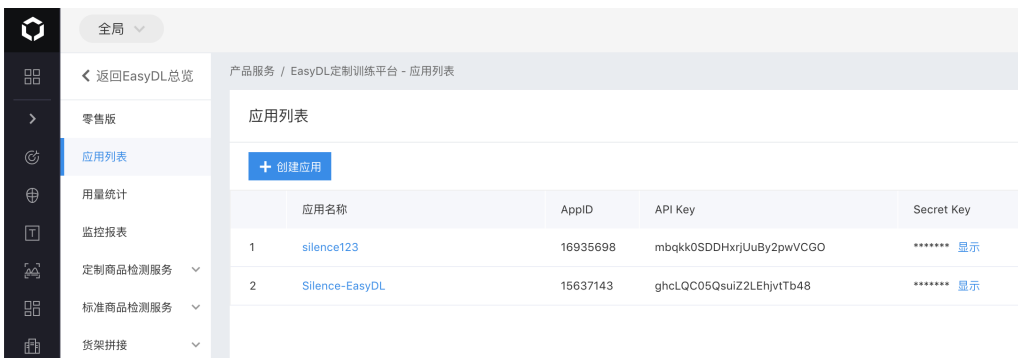
#### 简介

本文档主要说明如何使用翻拍识别API，如有疑问可以通过以下方式联系我们：

- 在百度云控制台内[提交工单](#)，咨询问题类型请选择**人工智能服务**

#### 接口鉴权

- 进入EasyDL零售版的百度云控制台[应用列表页面](#)，如下图所示：



- 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权参考文档](#)，使用API Key(AK)和Secret Key(SK)获取access\_token

接口调用

请求说明

请求示例

HTTP 方法：POST

商品陈列翻拍识别请求URL：<https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/recapture>

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token,参考“ <a href="#">Access Token获取</a> ”

Header如下：

参数	值
Content-Type	application/json

注意：如果出现336001和336002的错误码很可能是因为请求方式错误，与其他图像识别服务不同的是定制化图像识别服务以json方式请求。

Body请求示例：

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数，参数详情如下：

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部

提示：image参数中“去掉头部”指的是图片经base64编码后的头部信息「data:image/jpeg;base64,」，如下图所示：

C:\fakepath\14458.jpg

```
data:image/jpeg;base64,9j/4RCRRXhpZgAATU0AKgAAAQAEQEAAAMAAABDYAAAAEBAAMAAABEgAAAECAMAA
AADAAAA2gEGAAMAAABAAIAAAEOAAIAAAAEbm9yAAEPAIAIAAAAHAAAA4AEQAIAAAAJAAAA5wESAAMAAABAAEAA
AEVAAMAAABAAAMAAEAUAUAAAABAAA8AEbAAUAAAABAAAA+AEoAMAAABAAIAAAEAAIAAAkAAABAAEYAAIAA
AAUAAABJAITAAMAAABAAEAAIdpAAQAAAABAAABOlgIAAQAAAABAAADeAAABFgACAAIAAhIVUFXRUKASFdJLUFMMDA
AAAr8gAAAJxAACvyAAAAnEEFkb2JlIFBob3Rvc2hvcCBDQyAyMDE3IChNYWNpbmRvc2gpADlwMTk6MDQ6MTQgMjE6MTU
6MiaAACWCmaFAAAAAQAAAVaCnQAFAAAAQAAAwKllaADAAAAAQACAAClJwADAAAAAQCAAACQAAAAHAAAABDAvMT
```

- PHP
- JAVA
- Python3
- C++



```

<?php
/**
 * 发起http post请求(REST API), 并获取REST请求的结果
 * @param string $url
 * @param string $param
 * @return - http response body if succeeds, else false.
 */
function request_post($url = "", $param = "")
{
    if (empty($url) || empty($param)) {
        return false;
    }

    $postUrl = $url;
    $curlPost = $param;
    // 初始化curl
    $curl = curl_init();
    curl_setopt($curl, CURLOPT_URL, $postUrl);
    curl_setopt($curl, CURLOPT_POSTFIELDS, $curlPost);
}

```

#### 返回说明

#### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
results	否	array(object)	分类结果数组
+name	否	string	分类名称, 结果会返回“recapture”和“original”两类, recapture为翻拍, original为原图。
+score	否	number	置信度, 分别返回“recapture”和“original”两类的置信度

#### 建议翻拍判定方法

设定一个判定为翻拍图片的阈值, 即如果recapture的score大于这个值, 则认为这张图片是翻拍。通常有两中对应的业务模式:

注: 以下数值均为建议值, 实际应用的阈值请结合业务实际情况和实测结果进行设定

1. 业务里查翻拍的原则是宁可错杀一千, 也不愿错放一个的, 那么可以把认为是翻拍的阈值放在0.8~0.95。
2. 业务里查翻拍的原则是允许错放过一些翻拍的图片, 但是查到的一定要正确, 那么可以把认为是翻拍的阈值放在0.98甚至0.99。

#### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如Access Token失效返回:

```

{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}

```

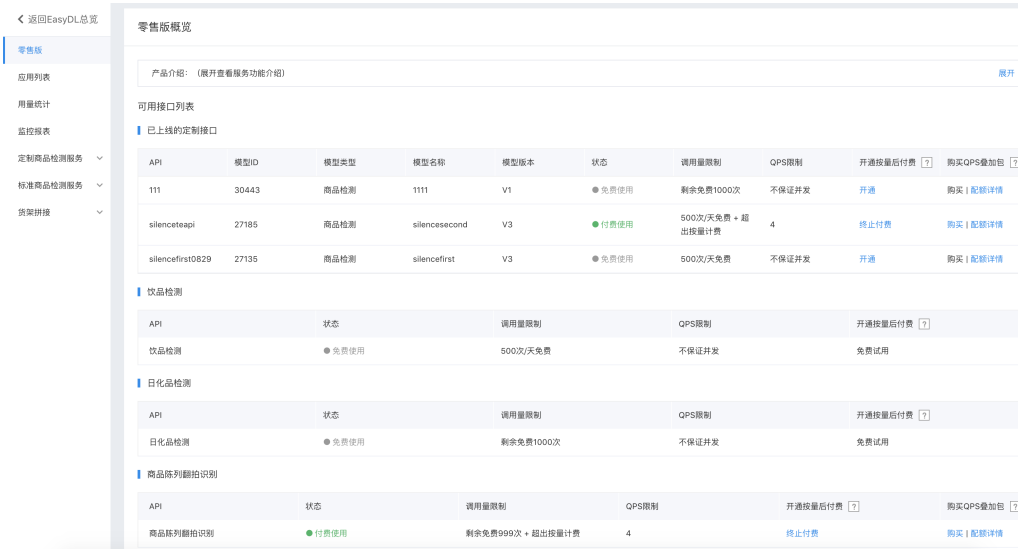
需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336003	Base64解码失败	图片格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336004	输入文件大小不合法	图片或音频、文本格式有误，请根据接口文档检查入参格式，有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336005	解码输入失败/分词错误	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队

## 购买指南

### ☞ 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。



🔗 计费方式

商品陈列翻拍识别目前支持下列三种计费方式:

1. 按调用量后付费
2. 调用量次数包预付费
3. QPS叠加包预付费

🔗 价目表 - 按调用量后付费

付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云EasyDL零售版控制台开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

🔗 价目表 - 调用量次数包

如果业务上对调用次数有预估，可以选择购买单次调用价格更低的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	490 元	4	1年 (366天)
10万次	4,800 元	4	1年 (366天)
100万次	45,000 元	4	1年 (366天)
500万次	212,500 元	4	1年 (366天)
1000万次	420,000 元	4	1年 (366天)
2000万次	800,000 元	4	1年 (366天)

**购买后不可退款**，次数包使用完后，开始按调用量每次0.05元收取费用

**特殊说明**，此计费方式仅限于单独调用翻拍模型接口，定制商品检测服务接口中的翻拍服务的计费不适用

## 🔗 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## 门店拜访SDK（原货架拼接服务）

### 服务介绍

#### 🔗 简介

百度门店拜访SDK主要包含货架拼接和门脸文字识别两个主要功能，下面是详细介绍。

#### 🔗 货架拼接

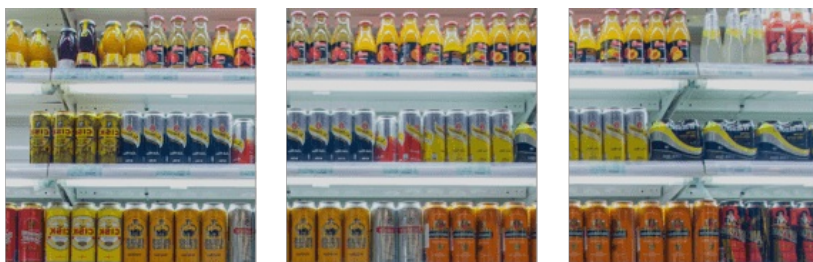
货架拼接服务基于百度EasyDL深度学习算法，支持将多个货架局部图片或视频，组合为完整货架图片。同时支持输出在完整货架图中的商品检测结果，包含SKU的名称和数量，适用于需要在长货架进行商品检测的业务场景。

#### 拼接方式介绍

货架拼接服务支持三种拼接方式：

1. 图片拼接-手机端实时拼接：拍摄图片进行拼接，可实时获得拼接结果
2. 图片拼接-云端非实时拼接：拍摄图片进行拼接，需要2~3分钟获得拼接结果
3. 视频拼接-云端非实时拼接：拍摄视频进行拼接，需要2~3分钟获得拼接结果

下面为货架拼接的效果图：



## 使用方式介绍

货架拼接服务提供以下三种使用方式：

- 云服务API，面向云端非实时拼接方式，可参考货架拼接文档[API调用方法](#)
- 可二次开发的SDK，支持iOS和Android，可参考文档[iOS SDK](#)和[Android SDK](#)
- 可以直接使用的体验APP，包含上述三种拼接方式的体验功能

## 使用须知

1. 使用货架拼接服务前请先参考文档[快速训练一个模型](#)或[模型发布](#)完成模型发布，货架拼接服务中需要使用商品检测服务输出完整货架图中的商品检测结果
2. 商品检测服务接口升级更新后，货架拼接服务会自动更新
3. 视频/图片-非实时拼接方式，每个账号共可免费使用200次货架拼接，SDK和体验APP均会消耗账号内的调用次数
4. 视频/图片-非实时拼接方式，每个账号仅支持同时进行一个拼接任务，超出需排队等待

## 体验APP功能简述

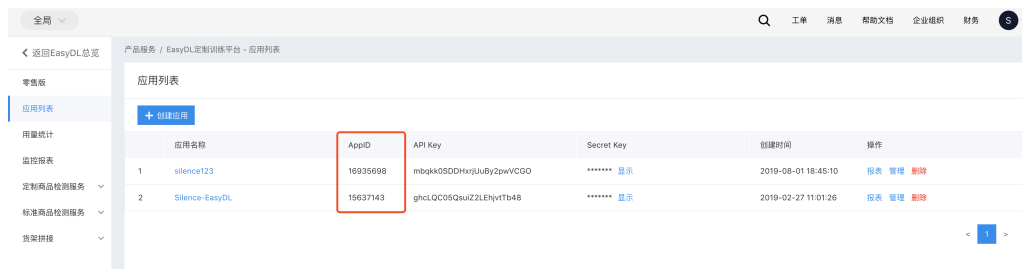
1. 支持拍摄一个或多个货架局部视频/图片进行拼接
2. 视频/图片-非实时拼接方式支持返回拼接后的完整图片，所有图片拼接去重后的商品识别结果
3. 图片拼接-手机端实时拼接支持实时拼接效果预览，所有图片拼接去重后的商品识别结果，不支持输出所有图片拼接的大图
4. 体验APP仅支持同时进行一个非实时拼接任务，超出需要排队等待

## 门脸文字识别

门脸文字识别功能支持识别图片中的门脸文字信息，包含门脸名称和描述文字。

## 使用须知

目前该服务处于邀请使用状态，请加入QQ群（群号:1009661589）联系管理员申请邀测，提供公司名称和在EasyDL零售版控制台[应用列表](#)创建应用的APPID，如下图：



## 使用方式介绍

门脸文字识别功能提供以下三种使用方式：

- 云服务API，需完成邀测申请
- 可二次开发的SDK，支持iOS和Android，可参考文档[iOS SDK](#)和[Android SDK](#)
- 可以直接使用的体验APP

## 购买指南

货架拼接服务支持按任务数后付费、任务次数包预付费和并发任务叠加包预付费三种计费方式。

## 开通付费及购买服务

货架拼接服务的「开通付费」、「购买任务次数包」、「购买并发任务叠加包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。



🔗 价目表 - 按任务数后付费

**付费调用**

每个账户享有累计200次免费调用额度，免费额度用尽后，请在百度智能云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

任务数	价格（元）	并发任务数限制	说明
每次拼接任务	0.2	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务启动后失败和运行前终止不计费，任务成功和运行后终止会计费用

**免费额度**

每个账号享有一定量免费调用额度，如下表：

服务	免费任务额度	并发任务数限制	说明
货架拼接	累计200次	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务成功与失败调用均消耗免费额度

🔗 价目表 - 任务次数包

如果对拼接任务次数有预估，可以选择购买**单次任务价格更低**的次数包，价格如下：

规格	价格	并发任务数限制	有效期
1千次	200 元	1	1年
1万次	1,900 元	1	1年
10万次	18,000 元	1	1年
100万次	150,000 元	1	1年
500万次	600,000 元	1	1年

购买后不可退款，任务次数包使用完后，开始按调用量每个任务0.2元收取费用

🔗 价目表 - 并发任务叠加包

开通付费后，并发任务数限制为1，如果有更多的并发请求需要，可以根据业务需求按天或按月购买并发任务叠加包，价格如下：

购买方式	每并发任务价格
按天购买	2元/天
按月购买	40元/月

购买 并发任务叠加包需保证已开通按量后付费或购买次数包

购买的并发任务叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## 体验APP

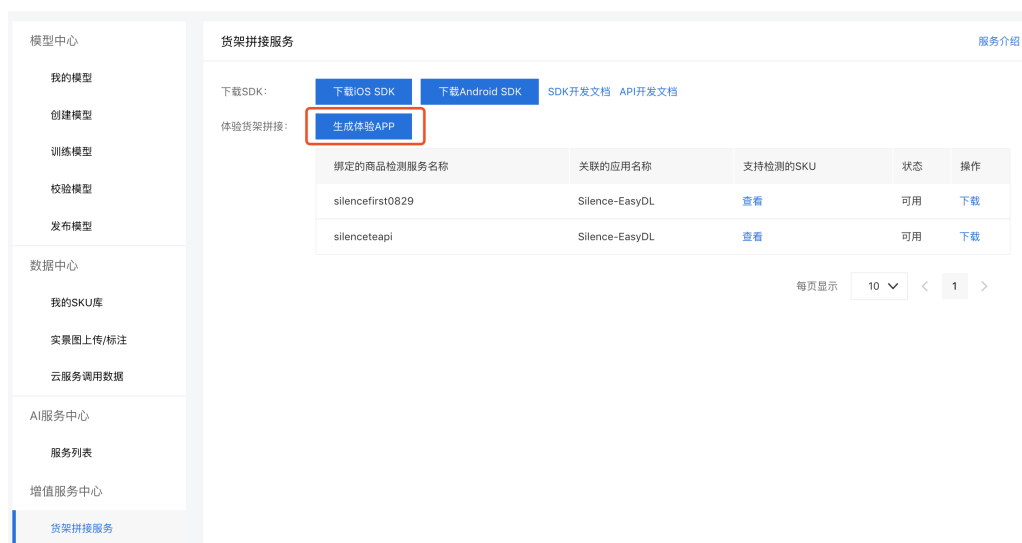
### 简介

本文档主要说EasyDL零售版的门店拜访体验APP（原货架拼接体验APP）如何获取和使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择**人工智能服务**
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:1009661589）联系群管

### 生成体验APP

在完成模型发布后，在EasyDL零售版上[货架拼接服务](#)页面上可以生成货架拼接体验APP，如下图所示：



点击「生成体验APP」按钮后弹出如下图所示弹窗：



需要选择两项内容：

- 选择服务：选择的服务用于对拼接的图片进行商品检测，该服务更新后，货架拼接服务会自动更新。一个商品检测服务只能被绑定到一个货架拼接体验APP上
- 选择应用：用于为体验APP鉴权，如未创建，请前往EasyDL零售版控制台[应用列表](#)进行创建

点击「生成体验APP」按钮后，可以看到页面上出现体验APP列表，可以在列表中看到该体验APP的状态为「生成中」，生成体验APP通常需要五分钟左右，如下图所示



## 使用体验APP

体验APP生成后，在体验APP列表中「操作」一列点击下载后会弹出下载二维码，使用手机摄像头扫码二维码后下载到手机使用。

## 更新体验APP

如果用于绑定体验APP的应用（在百度智能云控制台创建的应用）被删除，体验APP的鉴权将会失效，体验APP列表中的状态将会变为「不可用」，体验APP也将无法使用。如需继续使用该APP，可以在体验APP列表中更新该APP，点击「更新」后在弹窗内选择新的应用即可。

## API文档

### API调用方法

#### 简介

本文档主要说EasyDL零售版的货架拼接服务API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择人工智能服务
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

#### API总览

#### 接口列表

API名称	描述	API
创建任务	开始拼接整个流程	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create</a>
上传图片	上传货架局部图片	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload</a>
开始任务	启动货架拼接离线任务	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start</a>
查询结果	查询任务运行状态或者结果	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query</a>
终止任务	终止正在进行或者等待的任务	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate</a>
任务列表	列出所有状态的任务列表	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list</a>

#### 任务状态 task\_status

状态	描述
Created	已创建的任务
Queued	排队中的任务
Running	正在拼接的任务
Success	拼接成功的任务
Failure	拼接失败的任务
Terminated	被手动终止的任务

启动任务后，免费阶段，无论任务成功、失败、终止均会消耗免费任务数；付费使用阶段，仅对拼接成功和手动终止的任务进行计费。

#### 接口鉴权

- 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：





2. 如果还未创建应用, 请点击「创建应用」按钮进行创建。创建应用后, 参考[鉴权参考文档](#), 使用API Key(AK)和Secret Key(SK)获取 access\_token

## API使用方法

### 创建任务API

#### 接口描述

创建货架拼接任务, 开始整个拼接的流程。

#### 请求说明

#### 请求示例

HTTP 方法: POST

接口URL: [https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/create](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create)

URL参数:

参数	值
access_token	通过API Key和Secret Key获取的access_token, 参考 <a href="#">鉴权认证机制文档</a>

Header如下:

参数	值
Content-Type	application/json

Body中放置请求参数, 参数详情如下:

#### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
api_url	是	string	商品检测服务的url	无
row_image_nums	是	array[number]	各行待拼接货架图片的数量, array长度为货架图片的行数, array[i]为第i行的货架图片数量	行数不大于3, 行内图片数量不大于60
detection_threshold	否	float	商品检测服务的阈值	默认值为商品检测服务的阈值, 取值范围[0, 1]
nms_iou_threshold	否	float	检测框矫正去重的阈值	默认值为0.3, 取值范围[0.2, 0.8]

Body请求示例:

```
{
  "api_url": "http://xxxx",
  "row_image_nums": [3, 3, 4],
  "detection_threshold": 0.3,
  "nms_iou_threshold": 0.45
}
```

## 返回说明

## 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_id	否	string	新建任务对应的id
log_task_id	否	string	用于demo显示的任务id,用于问题定位
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxxxxx,
  "task_id": "xxxxx",
  "log_task_id": "xxx",
  "task_status": "Created"
}
{
  "log_id": xxxx,
  "error_code": 336204,
  "error_msg": "api name authentication failed"
}
```

## 上传图片API

### 接口描述

为指定任务上传待拼接的货架图

提示：只有在Created状态的任务才可以上传图片

### 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/upload](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无
row	是	number	图片对应的index	取值从0开始，需小于创建任务参数row_image_nums的长度
column	是	number	图片在行内所在的index	取值从0开始，需小于创建任务参数row_image_nums[row]的取值
image	是	string	上传图片的base64编码	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部

Body请求示例：

```
{
  "task_id": "xxxx",
  "row": 1,
  "column": 2,
  "image": "xxx=="
}
```

## 返回说明

接口返回参数：

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxx,
  "task_status": "Created"
}
{
  "log_id": xxx,
  "error_code": 336201,
  "error_msg": "unknown task id"
}
```

## 开始任务API

### 接口描述

开始执行货架拼接任务

提示：只有在Created状态的任务可以启动，若启动任务数到达用户的上限（默认为1，即同时只可以启动一个拼接任务），任务进入Queued状态。

### 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/start](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数名称	是否必需	参数类型	描述
task_id	是	string	货架拼接任务id

Body请求示例：

```
{
  "task_id": "xxxx"
}
```

返回说明

返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态
missed_location	否	array	缺失图片对应行和列的index

Body返回示例：

```
{ # 启动成功
  "log_id": xxxx,
  "task_status": "Running"
}
{ # 用户已运行的货架拼接任务已达上限，排队等待
  "log_id": xxxx,
  "task_status": "Queued"
}
{ # 货架图片未全部上传
  "log_id": xxxx,
  "error_code": 336211,
  "error_msg": "some images missed",
  "missed_location": [[0, 2], [1, 3]] # [[row, column]...]
}
```

查询结果API

接口描述

查询任务运行的状态或者结果信息

请求说明

请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/query](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

Body请求示例：

```
{
  "task_id": "xxxx"
}
```

返回说明

返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
log_task_id	否	string	用于demo显示的任务id,用于问题定位
task_status	否	string	任务状态
task_result	否	dict	任务拼接结果
+image_url	否	string	拼接成功后大图的url
+preview_image_url	否	string	拼接成功后大图预览图的url，压缩到1M以下，用于快速预览
+bbox	否	array	在拼接大图上的商品检测框
++name	否	string	商品名称
++score	否	float	检测框置信度
++sku_code	否	string	商品对应的sku code
++location	否	dict	检测框的位置
+++left	否	number	检测框的左上角像素点的横坐标
+++top	否	number	检测框的左上角像素点的纵坐标
+++height	否	number	检测框的高度
+++width	否	number	检测框的宽度
+sku_stat_info	否	array	在拼接大图上的商品检测框的统计信息
++name	否	string	商品名称
++sku_code	否	string	商品对应的sku code
++count	否	number	检测对应商品的数量
++proportion	否	float	统计商品在完整图片中的排面占比
+stitch_error_code	否	array[number]	拼接错误码
+fail_msg	否	string	拼接失败的错误信息

## stitch\_error\_code取值

stitch_error_code	描述
0	拼接成功
100	水平矫正失败
200	竖直矫正失败
300	拼接失败, 可能原因相邻图像重叠度不足30%
400	显存不足(OOM), 图片数量过多
500	GPU所能分配的单张图片的显存不足, 单张图片太大

Body返回示例 :

```
{ # 任务(已创建/排队中/运行中/已取消)
  "log_id": xxx,
  "log_task_id": "xxx",
  "task_status": "Created/Queued/Running/Terminated"
}
{ # 拼接任务运行成功
  "log_id": xxx,
  "log_task_id": "xxx",
  "task_status": "Success",
  "task_result": {
    "image_url": "https://xxxx",
    "preview_image_url": "https://xxxx",
    "bbox": [{
      "name": "xxx",
      "score": xxx,
      "sku_code": "xxx",
      "location": {
        "left": xxx,
        "top": xxx,
        "width": xxx,
        "height": xxx
      }
    }
  ], ... ],
  "sku_stat_info": [{
    "sku_code": "xxx",
    "name": "xxx",
    "count": n
  }, ...],
  "stitch_error_code": [100, ...]
}
{ # 拼接任务运行失败
  "log_id": xxx,
  "log_task_id": "xxx",
  "task_status": "Failure",
  "task_result": {
    "fail_msg": "image stitch job running timeout"
  }
}
{ # 请求错误
  "log_id": xxx,
  "error_code": 336201,
  "error_msg": "unknown task id"
}
```

## 终止任务API

## 接口描述

终止正在进行或者排队的任务

## 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/terminate](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

Body请求示例：

```
{
  "task_id": "xxxx"
}
```

### 返回说明

### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxx,
  "task_status": "Terminated"
}
```

### 任务列表API

### 接口描述

根据查询条件查询任务列表，多个条件取交集；按照创建时间倒序。

### 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/list](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_ids	否	array[string]	只返回指定id的任务信息	无
begin_time	否	number	只返回begin_time以后创建的任务信息	时间戳
end_time	否	number	只返回end_time之前创建的任务信息	时间戳

Body请求示例：

```
{
  "task_ids": ["xx", "xxx"],
  "begin_time": 1562763431,
  "end_time": 1562763842
}
```

返回说明

返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
tasks_info	否	array	任务列表
+task_id	否	string	任务id
+log_task_id	否	string	用于demo显示的任务id,用于问题定位
+task_status	否	string	任务状态
+create_time	否	number	任务创建时间

Body返回示例：

```
{
  "log_id": xxxx,
  "tasks_info": [
    {
      "task_id": "xxx",
      "log_task_id": "xxx",
      "task_status": "Created/Queued/Running/...",
      "create_time": 1562763842
    }, ...
  ]
}
```

🔗 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。



例如图片大小超4M导致报错，错误返回以下内容：

```
{
  "error_code": 336210,
  "error_msg": "invalid image size"
}
```

货架拼接服务错误码如下表：

error_code	error_msg	描述
336200	internal error	内部错误，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
336201	unknown task id	未知的任务id
336202	invalid param: 'param_name'	请求参数'param_name'的参数值不合法
336203	missing param: 'param_name'	请求参数'param_name'缺失
336204	api name authentication failed	api name和app_id认证失败
336205	current task status not support {action}.	当前任务状态不支持对应的操作：只有Created状态下的任务可以进行加图和启动任务，Created/Queued/Running状态下的任务可被终止
336206	invalid base64	加图操作：错误的base64图片编码
336207	failed loading image	加图操作：加载图片失败
336208	invalid image format	加图操作：不支持的图片格式，支持格式: bmp、jpg、jpeg、png
336209	invalid image shape	加图操作：不支持的图片形状，货架图片长宽需大于等于200
336210	invalid image size	加图操作：不支持的图片大小，图片大小不超过4M
336211	some images missed	启动拼接任务：货架图片缺失
336212	invalid json	请求数据格式不正确

## API调用方法

### 简介

本文档主要说EasyDL零售版的货架拼接服务API如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择[人工智能服务](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

### API总览

#### 接口列表

API名称	描述	API
创建任务	开始拼接整个流程	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create</a>
上传图片	上传货架局部图片	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload</a>
开始任务	启动货架拼接离线任务	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start</a>
查询结果	查询任务运行状态或者结果	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query</a>
终止任务	终止正在进行或者等待的任务	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate</a>
任务列表	列出所有状态的任务列表	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list</a>

#### 任务状态 task\_status

状态	描述
Created	已创建的任务
Queued	排队中的任务
Running	正在拼接的任务
Success	拼接成功的任务
Failure	拼接失败的任务
Terminated	被手动终止的任务

启动任务后，免费阶段，无论任务成功、失败、终止均会消耗免费任务数；付费使用阶段，仅对拼接成功和手动终止的任务进行计费。

## 接口鉴权

1. 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权参考文档](#)，使用API Key(AK)和Secret Key(SK)获取access\_token

## API使用方法

### 创建任务API

#### 接口描述

创建货架拼接任务，开始整个拼接的流程。

#### 请求说明

#### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/create](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/create)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
api_url	是	string	商品检测服务的url	无
row_image_nums	是	array[number]	各行待拼接货架图片的数量，array长度为货架图片的行数，array[i]为第i行的货架图片数量	行数不大于3，行内图片数量不大于60
detection_threshold	否	float	商品检测服务的阈值	默认值为商品检测服务的阈值，取值范围[0, 1]
nms_iou_threshold	否	float	检测框矫正去重的阈值	默认值为0.3, 取值范围[0.2,0.8]

Body请求示例：

```
{
  "api_url": "http://xxxx",
  "row_image_nums": [3, 3, 4],
  "detection_threshold": 0.3,
  "nms_iou_threshold": 0.45
}
```

返回说明

返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_id	否	string	新建任务对应的id
log_task_id	否	string	用于demo显示的任务id,用于问题定位
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxxxxx,
  "task_id": "xxxxx",
  "log_task_id": "xxx",
  "task_status": "Created"
}
{
  "log_id": xxxx,
  "error_code": 336204,
  "error_msg": "api name authentication failed"
}
```

上传图片API

接口描述

为指定任务上传待拼接的货架图

提示：只有在Created状态的任务才可以上传图片

请求说明

请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/upload](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/upload)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无
row	是	number	图片对应的index	取值从0开始，需小于创建任务参数row_image_nums的长度
column	是	number	图片在行内所在的index	取值从0开始，需小于创建任务参数row_image_nums[row]的取值
image	是	string	上传图片的base64编码	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少15px，最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部

Body请求示例：

```
{
  "task_id": "xxxx",
  "row": 1,
  "column": 2,
  "image": "xxx=="
}
```

返回说明

接口返回参数：

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxxx,
  "task_status": "Created"
}
{
  "log_id": xxxxx,
  "error_code": 336201,
  "error_msg": "unknown task id"
}
```

开始任务API

接口描述

开始执行货架拼接任务

提示：只有在Created状态的任务可以启动，若启动任务数到达用户的上限（默认为1，即同时只可以启动一个拼接任务），任务进入Queued状态。

请求说明

请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/start](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/start)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

请求参数

参数名称	是否必需	参数类型	描述
task_id	是	string	货架拼接任务id

Body请求示例：

```
{
  "task_id": "xxxx"
}
```

返回说明

返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态
missed_location	否	array	缺失图片对应行和列的index

Body返回示例：

```

{ # 启动成功
  "log_id": xxxx,
  "task_status": "Running"
}
{ # 用户已运行的货架拼接任务已达上限，排队等待
  "log_id": xxxx,
  "task_status": "Queued"
}
{ # 货架图片未全部上传
  "log_id": xxxx,
  "error_code": 336211,
  "error_msg": "some images missed",
  "missed_location": [[0, 2], [1, 3]] # [[row, column]...]
}

```

## 查询结果API

### 接口描述

查询任务运行的状态或者结果信息

### 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/query](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/query)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

Body请求示例：

```

{
  "task_id": "xxxx"
}

```

### 返回说明

### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id, 用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
log_task_id	否	string	用于demo显示的任务id,用于问题定位
task_status	否	string	任务状态
task_result	否	dict	任务拼接结果
+image_url	否	string	拼接成功后大图的url
+preview_image_url	否	string	拼接成功后大图预览图的url, 压缩到1M以下, 用于快速预览
+bbox	否	array	在拼接大图上的商品检测框
++name	否	string	商品名称
++score	否	float	检测框置信度
++sku_code	否	string	商品对应的sku code
++location	否	dict	检测框的位置
+++left	否	number	检测框的左上角像素点的横坐标
+++top	否	number	检测框的左上角像素点的纵坐标
+++height	否	number	检测框的高度
+++width	否	number	检测框的宽度
+sku_stat_info	否	array	在拼接大图上的商品检测框的统计信息
++name	否	string	商品名称
++sku_code	否	string	商品对应的sku code
++count	否	number	检测对应商品的数量
++proportion	否	float	统计商品在完整图片中的排面占比
+stitch_error_code	否	array[number]	拼接错误码
+fail_msg	否	string	拼接失败的错误信息

#### stitch\_error\_code取值

stitch_error_code	描述
0	拼接成功
100	水平矫正失败
200	竖直矫正失败
300	拼接失败, 可能原因相邻图像重叠度不足30%
400	显存不足(OOM), 图片数量过多
500	GPU所能分配的单张图片的显存不足, 单张图片太大

Body返回示例：

```

{ # 任务(已创建/排队中/运行中/已取消)
  "log_id": xxxx,
  "log_task_id": "xxx",
  "task_status": "Created/Queued/Running/Terminated"
}
{ # 拼接任务运行成功
  "log_id": xxxx,
  "log_task_id": "xxx",
  "task_status": "Success",
  "task_result": {
    "image_url": "https://xxxx",
    "preview_image_url": "https://xxxx",
    "bbox": [{
      "name": "xxx",
      "score": xxx,
      "sku_code": "xxx",
      "location": {
        "left": xxx,
        "top": xxx,
        "width": xxx,
        "height": xxx
      }
    }], ... ],
    "sku_stat_info": [{
      "sku_code": "xxx",
      "name": "xxx",
      "count": n
    }], ... ],
    "stitch_error_code": [100, ...]
  }
}
{ # 拼接任务运行失败
  "log_id": xxxx,
  "log_task_id": "xxx",
  "task_status": "Failure" ,
  "task_result": {
    "fail_msg": "image stitch job running timeout"
  }
}
{ # 请求错误
  "log_id": xxxx,
  "error_code": 336201,
  "error_msg": "unknown task id"
}

```

## 终止任务API

### 接口描述

终止正在进行或者排队的任务

### 请求说明

### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/terminate](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/terminate)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：



参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

Body请求示例：

```
{
  "task_id": "xxxx"
}
```

#### 返回说明

#### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
task_status	否	string	任务状态

Body返回示例：

```
{
  "log_id": xxxx,
  "task_status": "Terminated"
}
```

#### 任务列表API

#### 接口描述

根据查询条件查询任务列表，多个条件取交集；按照创建时间倒序。

#### 请求说明

#### 请求示例

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_stitch/list](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_stitch/list)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

参数名称	是否必需	参数类型	描述	参数值限制
task_ids	否	array[string]	只返回指定id的任务信息	无
begin_time	否	number	只返回begin_time以后创建的任务信息	时间戳
end_time	否	number	只返回end_time之前创建的任务信息	时间戳

Body请求示例：

```
{
  "task_ids": ["xx", "xxx"],
  "begin_time": 1562763431,
  "end_time": 1562763842
}
```

## 返回说明

### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
error_code	否	number	错误码
error_msg	否	string	错误描述
tasks_info	否	array	任务列表
+task_id	否	string	任务id
+log_task_id	否	string	用于demo显示的任务id,用于问题定位
+task_status	否	string	任务状态
+create_time	否	number	任务创建时间

Body返回示例：

```
{
  "log_id": xxxx,
  "tasks_info": [
    {
      "task_id": "xxx",
      "log_task_id": "xxx",
      "task_status": "Created/Queued/Running/...",
      "create_time": 1562763842
    }, ...
  ]
}
```

## SDK文档

### SDK介绍

#### 简介

本文档介绍门店拜访（原货架拼接）iOS、Android SDK的获取方式和系统支持，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)，咨询问题类型请选择[人工智能服务](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动
- 加入EasyDL官方QQ群（群号:1009661589）联系群管

#### 获取SDK

在EasyDL零售版的[门店拜访SDK页面](#)（[原货架拼接服务](#)）直接下载，如下图所示：

门店拜访SDK

操作文档 常见问题

功能介绍

门店拜访SDK集成了适用于快消场景的AI能力，保障快消业务代表在线下拜访门店过程中的行为规范和业务需要。

1. 人脸文字识别：支持识别门店图片中门店名称和其他文字信息，适用于门店拜访签到和门店注册验证
2. 货架拼接：支持在手机端将实时拍摄的货架图片进行拼接，最终获得将重复区域去重后的商品识别结果，适用于在货架进行商品识别的业务场景
3. 拍摄行为引导：支持拍摄过程中对光线、手机方向、照片是否模糊、照片是否被替换进行检测，保障拍摄的照片真实有效

下载SDK：[下载iOS SDK](#) [下载Android SDK](#) [SDK开发文档](#) [API开发文档](#)

体验货架拼接：[生成体验APP](#)

绑定的商品检测服务名称	关联的应用名称	版本	支持检测的SKU	状态	操作
标准服务-饮品检测	Silence-EasyDL	iOS: V5.0.0 Android: V5.0.0	-	可用	<a href="#">下载</a>
111	Silence-EasyDL	iOS: V1.0.0 Android: V1.0.0	<a href="#">查看</a>	可用	<a href="#">下载</a> <a href="#">更新</a>
silencefirst0829	Silence-EasyDL	iOS: V1.0.0 Android: V1.0.0	<a href="#">查看</a>	可用	<a href="#">下载</a> <a href="#">更新</a>
silenceapi	Silence-EasyDL	iOS: V1.0.0 Android: V1.0.0	<a href="#">查看</a>	可用	<a href="#">下载</a> <a href="#">更新</a>

< 1 > 10 条/页

## iOS系统支持

系统：iOS 9.0 以上

硬件：armv7 arm64 (Standard architectures)（暂不支持模拟器）

## Android系统支持

系统：Android Level 22以上

## iOS\_SDK

### 简介

本文档描述货架拼接iOS SDK如何使用。

### 系统支持

系统：iOS 9.0 以上

硬件：armv7 arm64（Standard architectures）（暂不支持模拟器）

### Release Notes

时间	版本	说明
2022.12.22	5.0.0	更名为门店拜访SDK，新增人脸文字识别功能、防窜拍功能
2021.12.22	4.1.0	新增手机端实时拼接模糊图像检测功能
2021.10.20	4.0.0	新增手机端实时拼接功能
2021.03.09	3.0.1	新增光线和手机方向检测功能
2020.12.30	3.0.0	新增支持拍摄图片，云端拼接功能
2020.11.12	2.0.0	新增支持排面统计占比
2019.08.30	1.0.0	支持拍摄视频，端上抽帧，云端拼接

## 集成指南 库依赖

SDK依赖以下静态库/动态库，需正确集成至项目中并配置Framework Search Paths / Header Search Paths / Library Search Paths：

- opencv2.framework：OpenCV V4.5.2，必须引入
- libmontage\_algo.a：手机端实时拼接功能库，可选引入，集成时请一并拷贝头文件目录montage\_algo至项目合适路径
- libEasyDL.a：模糊图像检测引擎库，可选引入，集成时请一并拷贝头文件目录EasyDL至项目合适路径
- libpaddle\_api\_full\_bundled.a：模糊图像检测引擎库，可选引入，集成时请一并拷贝头文件目录paddlelite至项目合适路径

libEasyDL.a 和 libpaddle\_api\_full\_bundled.a 需同时引入才可支持模糊图像检测

## 集成摄像头相关逻辑

UI部分包括摄像头代码均开源，可参考以下文件，用户拷贝相关代码至项目中即可，并修改相应文件名以避免符号冲突：

- easydl-stitch-ios/ViewController/ImagePickerController：云端拼接拍照逻辑，重合度算法开源
- easydl-stitch-ios/ViewController/MBStitchCameraViewController：手机端实时拼接拍照逻辑，拼接等相关算法依赖 libmontage\_algo.a
- easydl-stitch-ios/ViewController/VideoStitchViewController#startUIImagePicker()：云端拼接视频逻辑入口，参考该方法内对系统 UIImagePickerControllerController的使用

## 拍照拼接参数配置

```
// easydl-stitch-ios/EasyDLStitch/EasyDLStitchParams.h

##### define kThreshold 75 // 判断重合的阈值,0~100之间
##### define kStrategy "phash" // 重合算法,类型:["ahash","phash","dhash"]
##### define kThetaZ 60 // 手机倾斜Z轴角度阈值
##### define kThetaXY 20 // 手机倾斜XY轴角度阈值
##### define kOverLapControl true // 是否与遮罩重合才可以拍照
##### define kOrientationControl true // 是否手机持握方向符合要求才可以拍照
##### define kSkipFrames 3 // 跳过视频帧比对的数量,比如3为每3帧比对一次
##### define kBrightnessLow -3 // 光线强度阈值, -99~99之间
##### define kBrightnessHigh 5 // 光线强度阈值, -99~99之间
##### define kBrightnessControl false // 是否光线符合要求才可以拍照
```

参数说明：

- 获取重合度有"ahash","phash","dhash"三种算法，返回0~100之间的数值，越大表示重合度越高。不同算法返回的数值有区别，需相应调整阈值
- 手机倾斜XY轴指左右倾斜，Z轴是前后倾斜，当倾斜角度过大会影响拼接效果
- 设置只有手机倾斜角度、待拍摄图片与上一张图片重合度、环境光线亮度等条件符合要求才拍摄图片，保证拍摄效果
- 相机默认为每秒30帧，修改kSkipFrames的值调节做重合度对比的速度，避免卡顿或拍摄状态切换过快

## 视频拼接参数配置

体验APP中对视频截取帧的频率为1秒1帧，由于每个视频的帧数不能大于60，所以体验APP不能拼接长度大于60s的视频。开发者可根据实际情况调整截帧的频率，并相应限制视频长度。调整频率方法：

```
// easydl-stitch-ios/ViewController/StitchViewController.m

static int frameInterval = 1;//截帧间隔（秒）
```

并在合适的地方提示视频长度限制：

```
UIAlertController *actionSheet = [UIAlertController alertControllerWithTitle:@"选择对象(视频长度不能超过60s)" message:nil
preferredStyle:UIAlertControllerStyleActionSheet];
```

## SDK工程结构

```

EasyDL-Image-Stitching-iOS
├- LIB
├- include
│  ├── montage_algo/ // 手机端实时拼接功能库头文件
│  ├── EasyDL/ // 手机端实时拼接模糊图像检测引擎头文件
│  └- paddlelite/ // 手机端实时拼接模糊图像检测引擎头文件
├- libs
│  ├── opencv2.framework // OpenCV库
│  ├── libmontage_algo.a // 手机端实时拼接功能库
│  ├── libEasyDL.a // 手机端实时拼接模糊图像检测引擎
│  └- libpaddle_api_full_bundled.a // 手机端实时拼接模糊图像检测引擎
├- EasyDLStitch
│  ├── images/ // 资源文件
│  └- easydl-stitch-ios/ // Demo工程文件
├- RES/
│  ├── conf.json // API配置文件
│  └- fuzzy_model/ // 模糊图像检测模型

```

## SDK调用流程

### 获取鉴权

1. 进入EasyDL零售版的百度智能云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用，请点击「创建应用」按钮进行创建。创建应用后，参考[鉴权参考文档](#)，使用API Key(AK)和Secret Key(SK)获取 access\_token

下载SDK包后，填写ak、sk等信息。在RES/conf.json相应位置填入：

```

{
  "ak": "MzOzhObvEZ6InG1K3renXXXX", // API Key的值

  "sk": "188fRHYvLPmlPrNCDpBnkhL3ydXXXX", // Secret Key的值

  "apiUrl": "https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/detection/XXXX" // 定制商品检测服务API
}

```

### 云端非实时拼接调用流程

1. 创建任务：开始拼接整个流程
2. 加货架图：上传图片
3. 开始任务：启动货架拼接离线任务
4. 查询任务：查询拼接任务的状态和结果

其他：

1. 取消任务：取消正在进行或者等待的任务
2. 任务列表：查询所有状态的任务列表
3. 终止任务：终止正在进行或者等待的任务

### 调用API

云端拼接API的调用逻辑已封装在EasyDLStitchApiService文件中，以下为SDK调用货架拼接API的方法说明。货架拼接API接口的返回值及其他信息参见文档[货架拼接API调用方法](#)。

### 创建拼接任务

```
-(void)createSpliceTaskWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
api_url	是	string	商品检测服务的url	无
row_image_nums	是	array[number]	各行待拼接货架图片的数量，array长度为货架图片的行数，array[i]为第i行的货架图片数量	行数不大于3，行内图片数量不大于60
detection_threshold	否	float	商品检测服务的阈值	默认值为商品检测服务的阈值，取值范围[0, 1]
nms_iou_threshold	否	float	检测框矫正去重的阈值	默认值为0.45，取值范围[0.2,0.8]

### 上传图片

```
-(void)uploadImageWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无
row	是	number	图片对应行的index	取值从0开始，需小于创建任务参数row_image_nums的长度
column	是	number	图片在行内所在的index	取值从0开始，需小于创建任务参数row_image_nums[row]的取值
image	是	string	上传图片的base64编码	

### 启动拼接任务

```
-(void)startSpliceTaskWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

### 查询任务状态

```
-(void)queryTaskResultWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

### 查询任务列表

```
-(void)listTaskWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
task_ids	否	array[string]	只返回指定id的任务信息	无
begin_time	否	number	只返回begin_time以后创建的任务信息	时间戳
end_time	否	number	只返回end_time之前创建的任务信息	时间戳

### 终止任务

```
-(void)terminateTaskWithConfig:(NSDictionary *)config successHandler:(SuccessBlock)successHandler failHandler:
(FailureBlock)failHandler;
```

其中config为参数，后面两个回调block。参数取值及描述：

参数名称	是否必需	参数类型	描述	参数值限制
task_id	是	string	货架拼接任务id	无

### 手机端实时拼接调用流程

1. 准备目录
2. 对比图片：新图片(now)在参与实时拼接前需先与上一张参与拼接的图片(last)进行对比，如果now与last的对比特征合法则可以成功拼接，否则无法得到理想实时拼接结果
3. 实时拼接：SDK会寻找now.jpg并进行实时拼接得到新拼接结果
4. 上传云端，得到结果

其他：

1. 撤销拼接：撤销最后一次拼接结果

### SDK的调用

手机端实时拼接的调用逻辑已封装在EasyDLMBStitchApiService文件中，包括商品检测API以及拼接算法的调用，以下为SDK调用的方法说明。

### 准备目录

实时拼接过程中产生的文件需保存在本地，首先初始化准备保存的目录。单个拼接任务的保存目录不可发生变化。

```
-(void)prepare;

// 调用示例
[[EasyDLMBStitchApiService sharedService] prepare];
```

### 对比图片

```
-(void)compareImage:(UIImage *)image firstFrame:(BOOL)isFirstFrame completionHandler:(CompletionHandler)completionHandler;

// 调用示例
[[EasyDLMBStitchApiService sharedService] compareImage:image firstFrame:_firstFrameCompare completionHandler:^(id responseObject,
NSError *error) {
    if (error) {
        NSLog(@"%@", error.localizedDescription);
    } else {
        // 回调结果
        CompareResult *compareResult = responseObject;
        // do something
    }
}];
```

参数说明：

- image：当前要参与对比的图片

- isFirstFrame：是否是第一帧
  - 第一帧的定义取决于上一张参与拼接的图片（last）是否已经被对比过。假设有图片A和B，先用A与last对比，且last是初次被对比，此时isFirstFrame应为true，再用B与last对比，此时isFirstFrame应为false
- completionHandler：异步回调，在主线程

#### \*\*CompareResult\*\*

```
// montage_algo/EasyDLStitchAlgo.h

// 当前图片相对上一张参与拼接的图片的方位
@property(nonatomic) ImageDirection direction;
/**
 * 是否需要判断方向，如当拍摄完图片过近时，direction可能由于两图过于相似而不可靠，这种情况不需要判断方向，即该值=false
 * 一般direction不可靠时，该值=false
 * 对比的两张图是第一次对比时，该值=false
 * 当该值=true时，请在调用实时拼接API前确认方法是否合法，否则可能导致拼接失败
 */
@property(nonatomic, getter = needCheckDirection) BOOL checkDirection;
// 对比结果中的方位是否合法，非法的方位将无法完成拼接
@property(nonatomic) BOOL directionValid;
// 两张图片重叠部分的点位
@property(nonatomic, retain) NSArray<NSValue *> *points;
// 两张图片重叠状态
@property(nonatomic) OverlapStatus overlapStatus;
// 最后一次参与拼接的图片序号，从1开始
@property(nonatomic) int lastImgIndex;
```

#### \*\*对比结果的合法性判断参考\*\*

```
switch (compareResult.overlapStatus) {
    case OverlapStatus_Correct:
        if (compareResult.needCheckDirection && !compareResult.directionValid) {
            // 非法，当前参与对比的图片方位不正确，无法拼接
        } else {
            // 合法
        }
        break;
    case OverlapStatus_TooFar:
        // 非法，两张图重叠度过低
        break;
    case OverlapStatus_TooClose:
        // 非法，两张图重叠度过高
        break;
}
```

#### 实时拼接

- 建议调用拼接前参考【对比结果的合法性判断】，用不合法的对比结果进行实时拼接将无法获得正确输出
- 将要参与拼接的图片必须命名为“now.jpg”（也可使用EasyDLMBStitchApiService.FILENAME\_IMAGE\_NOW），并保存在 [EasyDLMBStitchApiService sharedService].currentTask.workDir指向的目录下，否则实时拼接无法正常工作。成功拼接后“now.jpg”会被SDK重新命名为{index}.jpg，其中{index}代表图片序号。
- 为获得更快的拼接效率，建议减小参与拼接的图片尺寸；为了保证拼接效果，缩放后的图片尺寸应不小于宽648和高864



```

- (void)stitchImageWithCompareResult:(CompareResult *)compareResult completionHandler:(CompletionHandler)completionHandler;

// 调用示例
[[EasyDLMBStitchApiService sharedService] stitchImageWithCompareResult:_latestCompareResult completionHandler:^(id responseObject,
NSError *error) {
    if (error) {
        NSLog(@"%@", error.localizedDescription);
    } else {
        // 回调结果
        StitchResult *stitchResult = responseObject;
        // do something
    }
}];

// 保存now.jpg示例
NSURL *workDirUrl = [EasyDLMBStitchApiService sharedService].currentTask.workDir;
/* 小图为拼接，大图为获得更好的商品检测效果 */
[EasyDLFileManager saveImage:image toUrl:[workDirUrl URLByAppendingPathComponent:FILENAME_IMAGE_NOW]
andResizeTo:CGSizeMake(648, 864)];
// SDK默认使用保存的一系列`{index}.jpg`调用商品检测API并取得结果。
// 由于建议减小该系列图片尺寸以获得更优的拼接效率，但同时更小尺寸的图片对商品检测精度有一定影响，因此为提高精度，建议同时保存最佳尺寸的图片用于上传云端。
[EasyDLFileManager saveImageForBestInfer:image toUrl:[workDirUrl URLByAppendingPathComponent:[NSString
stringWithFormat:@"%d/%d", DIR_NAME_FULL_IMAGE, FILENAME_IMAGE_NOW]];

```

参数说明：

- compareResult：对比图片回调返回的结果
- completionHandler：异步回调，在主线程。该回调在拼接任务全部完成后到达，如需更快获取缩略图和完整拼接图，可配置 EasyDLMBAPIStitchDelegate协议

**\*\*StitchResult\*\***

```

// montage_algo/EasyDLStitchAlgo.h

// 拼接错误码，0表示成功，其他为错误
@property(nonatomic) int errCode;
// 最近一张参与拼接的图片的序号
@property(nonatomic) int latestImgIndex;
// 缩略拼接图路径
@property(nonatomic, retain) NSURL *thumbnailUrl;
// 完整拼接图路径
@property(nonatomic, retain) NSURL *fullImageUrl;

```

**\*\*EasyDLMBAPIStitchDelegate\*\***

```

// easydl-montage-ios/EasyDLStitch/EasyDLMBStitchApiService.h

@protocol EasyDLMBAPIStitchDelegate <NSObject>
- (void)onStitchThumbnailGenerated:(NSURL *)thumbnailURL;
- (void)onStitchFullImageGenerated:(NSURL *)fullImageURL;
@end

// 配置示例
[EasyDLMBStitchApiService sharedService].stitchDelegate = self;

```

上传云端，得到结果

```

- (void)mergeDetectedResultsWithCompletionHandler:(CompletionHandler)completionHandler;

// 调用示例
[[EasyDLMBStitchApiService sharedService] mergeDetectedResultsWithCompletionHandler:^(id responseObject, NSError *error) {
    if (error) {
        NSLog(@"%@", error.localizedDescription);
    } else {
        // 回调结果
        MergeResult *mergeResult = responseObject;
        // do something
    }
}];

```

参数说明：

- completionHandler：异步回调，在主线程。该回调在检测任务全部完成后到达，如需感知检测任务的开始和进度更新，可配置 EasyDLMBAPIMergeDelegate 协议

#### **\*\*MergeResult\*\***

```

// montage_algo/EasyDLStitchAlgo.h

// 错误码，0表示成功，其他为错误
@property(n nonatomic) int errCode;
// 商品检测并去重合并后的结果
@property(n nonatomic, retain) NSDictionary *correctSKUDict;

```

#### **\*\*EasyDLMBAPIMergeDelegate\*\***

```

// easydl-montage-ios/EasyDLStitch/EasyDLMBStitchApiService.h

@protocol EasyDLMBAPIMergeDelegate <NSObject>
- (void)onMergeResultsStarted:(int)totalImageCount;
- (void)onMergeProgressUpdated:(int)totalImageCount completedCount:(int)completedCount;
@end

// 配置示例
[EasyDLMBStitchApiService sharedService].mergeDelegate = self;

```

#### 撤销拼接

SDK支持撤销最后一次拼接结果，如需撤销多张，请多次操作

```

- (void)undoLastTakenImageWithCompletionHandler:(CompletionHandler)completionHandler;

// 调用示例
[[EasyDLMBStitchApiService sharedService] undoLastTakenImageWithCompletionHandler:^(id responseObject, NSError *error) {
    if (error) {
        NSLog(@"%@", error.localizedDescription);
    } else {
        // 回调结果，包含撤销后，当前最后一次参与拼接的图片信息
        ImageInfo *lastImageInfo = responseObject;
        // do something
    }
}];

```

参数说明：

- completionHandler：异步回调，在主线程

#### **\*\*ImageInfo\*\***

```
// easydl-montage-ios/EasyDLStitch/EasyDLMBStitchApiService.h

// 图片序号
@property(nonatomic) int index;
// 图片列坐标
@property(nonatomic) int x;
// 图片行坐标
@property(nonatomic) int y;
```

### 模糊图像检测

手机端实时拼接已接入AI模型以支持模糊图像检测，除参考[库依赖](#)正确引入依赖库，需保证 RES/fuzzy\_model 目录下的模型文件存在。调用示例如下，也可参考 EasyDL-Stitch/easydl-stitch-ios/ViewController/MBStitchCameraViewController.m 文件中对 FuzzyModelProxy 的使用：

```
// 模型初始化
- (NSError *)modelInit;
// 推理图像判断是否模糊
- (void)inferImage:(UIImage *)image completionHandler:(void (^)(BOOL fuzzy, NSError *error))completionHandler;

// 调用示例
FuzzyModelProxy *fuzzyModelProxy = [[FuzzyModelProxy alloc] init];
[fuzzyModelProxy modelInit];
if (fuzzyModelProxy && fuzzyModelProxy.engineActive) {
    [fuzzyModelProxy inferImage:image completionHandler:^(BOOL fuzzy, NSError *error) {
        if (!error && !fuzzy) {
            // 图像非模糊
        } else {
            // 图像模糊或推理失败
        }
    }];
}
```

### 阈值设置

手机端实时拼接支持设置：

- 最小IOU置信度
- 最大IOU置信度
- NMS置信度
- 商品检测API最大重试次数

```
// easydl-montage-ios/EasyDLStitch/EasyDLMBStitchApiService.h

/**
 * 最小拼接引导置信度
 */
@property(nonatomic) CGFloat minIOUThreshold;
/**
 * 最大拼接引导置信度
 */
@property(nonatomic) CGFloat maxIOUThreshold;
/**
 * 去重置信度
 */
@property(nonatomic) CGFloat nmsIOUThreshold;
/**
 * 商品检测API重试次数
 */
@property(nonatomic) int detectRetryTimes;
/**
 * 商品检测API最大并发数
 */
@property(nonatomic) int maxQPS;
```

## 门脸文字识别调用流程

1. 初始化门店定位
2. 门脸图片上传云端

## SDK 调用

门脸文字识别流程通过 EasyDLDoorAPIService 调用，具体使用和返回参数见下

### 初始化门店定位

```
[[EasyDLDoorAPIService sharedService] startLocation];
```

### 门脸图片上传云端，获取门店检测结果

```
// easydl-montage-ios/EasyDLDoor/EasyDLDoorAPIService.h  
  
// 开始门脸文字识别  
[[EasyDLDoorAPIService sharedService] detectDoorImage:image];  
// 获取门脸识别结果  
[EasyDLDoorAPIService sharedService].blockNSDictionary = ^(NSDictionary * _Nonnull blockNSDictionary, NSError * _Nonnull error) {  
    if(blockNSDictionary != nil && error == nil) {  
        // 门脸图片识别成功  
        .....  
        // 校验门店结果  
        [EasyDLStitchAlgo checkDoorData:documentsDirectory ocrInfo:ocrJson];  
    }  
}
```

### 模糊图像检测

门脸文字识别已接入AI模型以支持模糊图像检测，除参考[库依赖](#)正确引入依赖库，需保证 RES/fuzzy\_model 目录下的模型文件存在。调用示例可参考【[手机端实时拼接调用流程-模糊图像检测](#)】，也可参考【[门脸文字识别调用流程](#)】 EasyDL-Stitch/easydl-stitch-ios/ViewController/DoorCameraViewController.mm 文件中对 FuzzyModelProxy 的使用。

### 防止图片窜拍开关参数设置

```
// easydl-montage-ios/EasyDLDoor/CheckImageConfig.h  
  
- (BOOL)getPirateImageCheck;  
  
// 调用示例  
/**  
 * 窜拍开关默认开启  
 */  
[CheckImageConfig sharedService].pirateImageCheck = true;  
// 获取开关状态  
[[CheckImageConfig sharedService] getPirateImageCheck];
```

**错误码** 以下为SDK使用的错误码，API接口错误码参见[货架拼接API错误码](#)。

错误码	说明
200002	模型配置错误, 请检查传入的配置文件是否有效
100006	API, AK/SK 换取token失败
100007	API, 请求 API 失败
100008	API, 请求商品检测API失败
200001	手机端实时拼接 - 前端引导对比图像出错
200003	手机端实时拼接 - 撤销上一次拼接结果出错
200004	手机端实时拼接 - 商品检测+去重过程中出错
200005	手机端实时拼接 - 拼接出错
300001	手机端实时拼接 - 模糊图像检测出错

## Android\_SDK

### 简介

本文介绍SDK的功能使用, 即下载包中的sdk module。

SDK为货架拼接 [云端非实时API](#)和手机端实时拼接的封装, 无任何额外功能。如果有和API文档不符的地方, 以SDK为准。

支持Android Level 22及以上编译和使用。

### Release Notes

时间	版本	说明
2022.12.22	5.0.0	更名为门店拜访SDK, 新增门脸文字识别功能、防牵拍功能
2021.12.22	4.1.0	新增手机端实时拼接模糊图像检测功能
2021.08.20	4.0.0	新增手机端实时拼接功能
2021.03.09	3.0.1	新增光线和手机方向检测功能
2020.12.30	3.0.0	新增支持拍摄图片, 云端拼接功能
2020.11.12	2.0.0	新增支持排面统计占比
2019.08.30	1.0.0	支持拍摄视频, 端上抽帧, 云端拼接

### 测试

#### 获取鉴权

1. 进入EasyDL零售版的百度智能云控制台 [应用列表页面](#), 如下图所示:



2. 如果还未创建应用, 请点击「创建应用」按钮进行创建。创建应用后, 参考[鉴权参考文档](#), 使用API Key(AK)和Secret Key(SK)获取 access\_token

```
{
  "ak": "Mz0zh0bvEZ6lnG1K3renXXXX", // API Key的值
  "sk": "188fRHYvLPmlPrNCdpBnkhL3ydXXXX", // Secret Key的值
  "apiUrl": "https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/detection/XXXX" // 定制商品检测服务API
}
```

正常情况下，启动的app及其功能和扫描二维码一致

demo的请求和结果会放在/sdcard/com.baidu.ai.easydl.montage中

### 测试云端拼接mini demo

测试app通过后，可以修改app/src/main/AndroidManifest.xml 内的启动app，修改为 "com.baidu.ai.easydl.minidemo.Minidemo" `com.baidu.ai.easydl.minidemo.Minidemo`

Minidemo中有3个task，测试时需要填入 Appkey， AppSecret， ApiUrl信息

- ApiTestAsyncTask ，测试简单流程。
- QueryAsyncTask ， 测试查询列表。
- RequestTestAsyncTask，测试assets/request下的图片输入。这个目录可以从SD卡中/sdcard/com.baidu.ai.easydl.montage/X/request复制。

### 手机端实时拼接调用流程

第1步：初始化

- 1) 【获取实例】
- 2) 【初始化API】

第2步：对比图片

新图片(now)在参与实时拼接前需先与上一张参与拼接的图片(last)进行对比，如果now与last的对比特征合法则可以成功拼接，否则无法得到理想实时拼接结果

第3步：实时拼接

SDK会寻找now.jpg并进行实时拼接得到新结果

第4步：上传云端，得到结果

### SDK 调用

手机端实时拼接各流程通过 MobileStitchAPI 调用，具体使用和返回参数见下

#### 初始化

MobileStitchAPI不支持多线程，且仅有一个实例以保证实时拼接过程中的正确文件操作。

#### 获取实例

```
/**
 * 获取实例
 *
 * @param appKey 网页上的应用的appkey
 * @param secretKey 网页上的应用的appSecret
 * @param apiUrl 商品检测服务API
 */
public static MobileStitchAPI getInstance(String appKey, String secretKey, String apiUrl);

/**
 * 获取实例
 *
 * @param appKey 网页上的应用的appkey
 * @param secretKey 网页上的应用的appSecret
 * @param apiUrl 商品检测服务API
 * @param numConcurrency 同时调用商品检测API的并发数
 */
public static MobileStitchAPI getInstance(String appKey, String secretKey, String apiUrl, int numConcurrency);
```

#### 初始化API

```
/**
 * 初始化API
 *
 * @param workDirPath 保存实时拼接过程产生的各类文件的路径
 */
public void init(String workDirPath);
```

- 异步回调：[MobileStitchAPIListener.onAPIPrepared\(int, int\[\]\)](#)

## 对比图片

```
/**
 * 对比当前图片与上一张参与拼接的图片
 *
 * @param currentImgBitmap 当前要参与对比的图片
 * @param firstFrame 是否是第一帧
 */
public void compareImages(Bitmap currentImgBitmap, boolean firstFrame);
```

- 第一帧的定义取决于上一张参与拼接的图片 (last) 是否已经被对比过。假设有图片A和B，先用A与last对比，且last是初次被对比，此时firstFrame应为true，再用B与last对比，此时firstFrame应为false。
- 异步回调：[MobileStitchAPIListener.onImagesCompared\(CompareResult\)](#)

## CompareResult

```
// 当前图片相对上一张参与拼接的图片的方位，参考MobileStitchAPI.DIRECTION_{LEFT|UP|RIGHT|DOWN|UNKNOWN}
public int getDirection();

// 对比结果中的方位是否合法，非法的方位将无法完成拼接
public boolean isDirectionValid();

/**
 * 是否需要判断方向，如当拍摄完图片过近时，direction可能由于两图过于相似而不可靠，这种情况不需要判断方向，即该值=false
 * 一般direction不可靠时，该值=false
 * 对比的两张图是第一次对比时，该值=false
 * 当该值=true时，请在调用实时拼接API前确认方法是否合法
 */
public boolean needCheckDirection();

// 两张图片重叠部分的点位
public List<PointF> getPoints();
```

## 对比结果的合法性判断参考

```
switch (compareResult.getOverlapStatus()) {
    case CompareResult.OVERLAP_CORRECT:
        if (compareResult.needCheckDirection() && !compareResult.isDirectionValid()) {
            // 非法，当前参与对比的图片方位不正确，无法拼接
        } else {
            // 合法
        }
        break;
    case CompareResult.OVERLAP_TOO_FAR:
        // 非法，两张图重叠度过低
        break;
    case CompareResult.OVERLAP_TOO_CLOSE:
        // 非法，两张图重叠度过高
        break;
}
```

## 实时拼接

```
/**
 * 拼接图片
 *
 * @param compareResult 对比图片回调返回的结果
 */
public void stitchImage(CompareResult compareResult);
```

- 建议调用拼接前参考【对比结果的合法性判断】，用不合法的对比结果进行实时拼接将无法获得正确输出
- 将要参与拼接的图片必须命名为“now.jpg”（也可使用MobileStitchAPI.IMAGE\_NAME\_NOW），并保存在初始化API时的 workDirPath 目录下，否

则实时拼接无法正常工作。成功拼接后"now.jpg"会被SDK重新命名为{index}.jpg，其中{index}代表图片序号。

- 为获得更快的拼接效率，建议减小参与拼接的图片尺寸；为了保证拼接效果，缩放后的图片尺寸应不小于宽648和高864
- 异步回调：
  - 缩略拼接图生成：[MobileStitchAPIListener.onStitchThumbnailGenerated\(String\)](#)
  - 完整拼接图生成：[MobileStitchAPIListener.onStitchFullImageGenerated\(String\)](#)
  - 拼接完成：[MobileStitchAPIListener.onStitchCompleted\(MobileStitchResult\)](#)

### MobileStitchResult

```
// 获取缩略拼接图路径
public String getThumbnailPath();

// 获取完整拼接图路径
public String getFullImgPath();

// 获取最近一张参与拼接的图片的序号
public int getLatestPhotoIndex();

// 拼接是否成功
public boolean isSuccess();
```

### 保存最佳尺寸的图片以提高商品检测精度

SDK默认使用以上保存的一系列{index}.jpg调用商品检测API并取得结果，由于建议减小该系列图片尺寸以获得更优的拼接效率，但同时更小尺寸的图片对商品检测精度有一定影响，因此为提高精度，建议同时保存最佳尺寸的图片用于上传云端。

```
// 1.原图
Bitmap bitmap = getFromSomewhere();

// 2.计算最佳缩放系数
float scaleFactor = calculateScaleFactor(bitmap);

// 3.缩放获得最佳尺寸的图片
Bitmap scaledBitmap = Bitmap.createScaledBitmap(bitmap,
    (int) (bitmap.getWidth() * scaleFactor),
    (int) (bitmap.getHeight() * scaleFactor),
    true);

// 4.保存
String fullImgFilepath = workDirPath + "/"
    + MobileStitchAPI.DIR_NAME_FULL_IMAGE + "/"
    + MobileStitchAPI.IMAGE_NAME_NOW;
ImageUtil.saveBitmap(scaledBitmap, fullImgFilepath);

/**
 * 计算最佳缩放系数
 */
private float calculateScaleFactor(Bitmap originalBitmap) {
    int longerSide;
    int shorterSide;
    if (originalBitmap.getWidth() > originalBitmap.getHeight()) {
        longerSide = originalBitmap.getWidth();
        shorterSide = originalBitmap.getHeight();
    } else {
        longerSide = originalBitmap.getHeight();
        shorterSide = originalBitmap.getWidth();
    }
    return Math.min(1333f / longerSide, 800f / shorterSide);
}
```

### 撤销拼接结果

SDK支持撤销最后一次拼接结果，请自行编码删除最后一张参与拼接的图片，再调用MobileStitchAPI.notifyLatestPhotoDeletion()通知SDK，参



考：

```
// 最后一张参与拼接的图片路径，workDirPath为初始化时工作目录路径
// latestPhotoIndex为最后一张参与拼接的图片序号，可在 MobileStitchAPIListener 以下回调时赋值
// 1. onAPIPrepared() - 参数takenPhotoSize
// 2. onStitchCompleted() - 参数result.getLatestPhotoIndex()
// 3. onDeletionConfirmed() - 参数latestPhotoIndex
String filepath = workDirPath + "/" + latestPhotoIndex + ".jpg";

// 删除图片
FileUtil.deleteFile(filepath);

// 通知SDK
// SDK确认删除后回调 MobileStitchAPIListener.onDeletionConfirmed(int, int[])
mobileStitchAPI.notifyLatestPhotoDeletion();
```

上传云端，得到结果

```
/**
 * 上传云端检测，并获得结果
 */
public void mergeDetectResults();
```

• 异步回调：

- 上传进度更新：[MobileStitchAPIListener.onDetectProgressUpdated\(int\)](#)
- 检测完成：[MobileStitchAPIListener.onDetectedResultsMerged\(MergeResult\)](#)

**MergeResult**

```
// 获取商品检测并去重后的结果
public String getCorrectedSKUJson();
```

**MobileStitchAPIListener**

手机端实时拼接通过 MobileStitchAPIListener 异步回调各函数结果，监听器可通过：

- `mobileStitchAPI.registerListener()`注册
- `mobileStitchAPI.unregisterListener()`注销

```

/**
 * API准备好时的回调
 *
 * @param takenPhotoSize 工作目录下已拍摄的图像数量
 * @param latestPhotoPos 最新拍摄图片的坐标
 */
void onAPIPrepared(int takenPhotoSize, int[] latestPhotoPos);

/**
 * 图片对比完成
 *
 * @param compareResult 对比结果
 */
void onImagesCompared(CompareResult compareResult);

/**
 * 调用拼接接口后，缩略图生成后的回调
 *
 * @param thumbnailName 缩略图在工作目录下的文件名
 */
void onStitchThumbnailGenerated(String thumbnailName);

/**
 * 调用拼接接口后，完整拼接图片生成后的回调
 *
 * @param fullImageName 完成拼接图片在工作目录下的文件名
 */
void onStitchFullImageGenerated(String fullImageName);

/**
 * 拼接完成回调
 *
 * @param mobileStitchResult 拼接结果
 */
void onStitchCompleted(MobileStitchResult mobileStitchResult);

/**
 * 删除确认回调
 *
 * @param latestPhotoIndex -1=操作失败，否则返回删除后，最新拍摄图片的下标；如删除了3.jpg，将返回2
 * @param latestPhotoPos 最新拍摄图片的坐标
 */
void onDeletionConfirmed(int latestPhotoIndex, int[] latestPhotoPos);

/**
 * 检测图片进度更新回调
 *
 * @param leftCount 剩余要处理图片的数量
 */
void onDetectProgressUpdated(int leftCount);

/**
 * 商品检测并去重处理完成的回调
 *
 * @param mergeResult 检测并去重结果
 */
void onDetectedResultsMerged(MergeResult mergeResult);

```

### 模糊图像检测

手机端实时拼接已接入AI模型以支持模糊图像检测，需引入以下依赖库及模型文件：

- libedge-infer.so：模糊图像检测引擎库
- easyedge-sdk.jar：模糊图像检测引擎库
- sdk/src/main/assets/infer/：模糊图像检测模型所在文件夹

以下为调用示例，也可参考 app/src/main/java/com/baidu/ai/easydl/montage/page/photo/mobilestitch/MobileStitchViewPresenter.java 类中

对 FuzzyModelProxy 的使用：

```
/* 初始化 */
FuzzyModelProxy fuzzyModelStateListener = new FuzzyModelProxy.ModelStateListener() {
    @Override
    public void onInitialized(Exception exception) {
        if (exception != null) {
            // 模糊模型初始化失败
        } else {
            // 模糊模型初始化成功
        }
    }

    @Override
    public void onDestroyed() {
        // 模糊模型销毁回调
    }
};
FuzzyModelProxy fuzzyModelProxy = new FuzzyModelProxy(mContext, fuzzyModelStateListener);
fuzzyModelProxy.initModel();

/* 调用示例 */
if (fuzzyModelProxy.modelEngineActivate()) {
    Bitmap bitmap = bitmapFromSomewhere();
    fuzzyModelProxy.infer(bitmap, new FuzzyModelProxy.ModelInferListener() {
        @Override
        public void onCompleted(boolean fuzzy) {
            if (!fuzzy) {
                // 图像非模糊
            } else {
                // 图像模糊
            }
        }

        @Override
        public void onException(Exception exception) {
            // 图像检测失败
        }
    }
} else {
    // 模糊图像推理引擎异常
}

/* 销毁 */
if (fuzzyModelProxy != null) {
    fuzzyModelProxy.destroyModelEngine();
    fuzzyModelStateListener = null;
}
```

## 阈值的设置

手机端实时拼接支持设置：

1. 最小IOU置信度
2. 最大IOU置信度
3. NMS置信度
4. 商品检测API最大重试次数

```
/**
 * 设置最低iou置信度，需在init()后调用
 *
 * @param threshold 在0-1范围内有效
 */
public void setMinIouThreshold(float threshold);

/**
 * 设置最高iou置信度，需在init()后调用
 *
 * @param threshold 在0-1范围内有效
 */
public void setMaxIouThreshold(float threshold);

/**
 * 设置NMS置信度，需在init()后调用
 *
 * @param threshold 在0-1范围内有效
 */
public void setNmsIouThreshold(float threshold);

/**
 * 设置商品检测API最大重试次数
 *
 * @param maxRetryTimes 最大重试次数，<=0无效
 */
public void setMaxRetryTimes(int maxRetryTimes);
```

### 云端非实时拼接调用流程

第1步，创建任务，上传图片

- 1) 【创建任务：开始拼接整个流程】
- 2) 【加货架图：上传图片】
- 3) 【开始任务：启动货架拼接离线任务】

第2步，不定时查询结果，一般10分钟后有结果参数

【查询结果：查询任务运行状态或者结果】

其它可选操作：

- 【终止任务：终止正在进行或者等待的任务】
- 【任务列表：查询所有状态的任务列表】

### SDK 调用

根据调用流程，SDK有两种调用方式：

- StitchApi，api的封装
- StitchTask，StitchApi的封装，避免taskId的传递。一个task对应一个StitchTask

返回参数以及其他信息详见文档[货架拼接API调用方法](#)。

### StitchApi

#### 初始化

```

/**
 * 初始化
 * @param appKey 网页上的应用的appkey
 * @param secretKey 网页上的应用的appSecret
 */
public StitchApi(String appKey, String secretKey) {
    super(appKey, secretKey);
}

/**
 * 初始化
 * @param appKey 网页上的应用的appkey
 * @param secretKey 网页上的应用的appSecret
 * @param connection 自定义HTTP连接
 */
public StitchApi(String appKey, String secretKey, ISdkConnection connection) {
    super(appKey, secretKey, connection);
}

```

### 创建任务

```

public CreateStitchResponse create(CreateStitchRequest request);

// CreateStitchRequest 及 CreateStitchResponse 参数同API文档

```

### 同步上传图片

```

public CommonStitchResponse upload(UploadImageRequest request);

// UploadImageRequest 及 CreateStitchResponse 参数同API文档

设置图片的话，以下2个方法2选1

public void setImageFile(String imageFile) ;

public void setImageInputStream(InputStream inputStream)

```

### 异步上传图片

```

public void uploadAsync(UploadImageRequest request,
    IApiResponseListener<CommonStitchResponse> listener)

// UploadImageRequest 参数同API文档

// IApiResponseListener<CommonStitchResponse> 接口 :
onSdkResponse(CommonStitchResponse response, String userDefinedRequestId)
// 其中userDefinedRequestId是在UploadImageRequest 里面设置的

// 使用 clearAysncQueue()可以清空未开始的任務

```

### 开始任务

```

CommonStitchResponse start(CommonStitchRequest request)

```

### 查询结果

```

public QueryStitchReponse query(CommonStitchRequest request)

```

### 任务列表

```

public ListStitchResponse list(ListStitchRequest request)

```

## StitchTask

一个任务新建一个StitchTask 调用方式同 StitchApi，参数中不必设置taskId

## AbstractApiRequest

目前Request类的基类。

```

// 设置自定义请求Id，调用异步接口的回调使用
public void setUserDefinedRequestId(String userDefinedRequestId)

// 设置是否添加debug日志
public void setEnableDebug(boolean enableDebug)

```

## CommonStitchResponse 及 AbstractApiResponse

```

// 获取任务状态
public String getTaskStatus();

// 获取logId
public String getLogId();

// 获取服务端返回的原始json
public JSONObject getOriginalJson();

// 获取请求
public AbstractApiRequest getRequest();

```

## 门脸文字识别调用流程

第1步：初始化

- 1) 【获取实例】
- 2) 【初始化API】

第2步：门脸图片上传云端，获取门脸文字识别结果

- 1) 【门脸文字识别】
- 2) 【释放资源】

## SDK 调用

门脸文字识别流程通过 DetectionDoorAPI 调用，具体使用和返回参数见下

### 初始化

DetectionDoorAPI不支持多线程，且仅有一个实例。

### 获取实例

```

/**
 * 获取实例
 * @param appKey 网页上的应用的appkey
 * @param secretKey 网页上的应用的appSecret
 */
public static DetectionDoorAPI getInstance(String appKey, String secretKey);

```

### 初始化API

```

/**
 * 初始化API
 * 建议传参getApplicationContext
 * 初始化（文件夹/定位）
 */
public void init(Context context);

```

```
// 注册门脸文字识别监听
public void registerListener(DoorAPIListener listener);

public interface DoorAPIListener {
    /**
     * 识别成功, 返回门脸文字识别结果
     * @param responseJson
     */
    void onDetectSuccess(String responseJson);
    /**
     * 识别异常
     * @param Exception e
     */
    void onException(Exception e);
}

// 销毁监听
public void unRegisterListener();
```

门脸图片上传云端，获取门店检测结果

### 门脸文字识别

```
/**
 * 开始门脸文字识别
 * @param bitmap
 */
public void detectDoorImage(Bitmap bitmap);
```

### 释放资源

```
// 释放资源
public void destroy();
```

### 模糊图像检测

门脸文字识别已接入AI模型以支持模糊图像检测，需引入以下依赖库及模型文件：

- libedge-infer.so：模糊图像检测引擎库
- easyedge-sdk.jar：模糊图像检测引擎库
- sdk/src/main/assets/infer/：模糊图像检测模型所在文件夹

调用示例可参考【手机端实时拼接调用流程-模糊图像检测】，也可参考【门脸文字识别调用流程】

app/src/main/java/com/baidu/ai/easydl/montage/page/door/IDoorViewPresenter.java类中对 FuzzyModelProxy 的使用。

### 集成指南

#### 添加NDK编译架构

SDK依赖OpenCV库，需添加NDK编译选项，支持常用的两个架构，可参考app/build.gradle配置

```
ndk {
    abiFilters "arm64-v8a", "armeabi-v7a"
}
```

#### 集成拍照逻辑

查看com.baidu.ai.easydl.montage.page.photo.take包，里面均为摄像拍照逻辑。

#### 拍照参数设置

```
package com.baidu.ai.easydl.montage.page.photo;

public interface IPhotoParam {

    /**
     * 两张图片的hash算法
     */
    String IMAGE_COMPARE_HASH_METHOD = "pHash"; // pHash,dHash,ahash

    /**
     * 两张图片的hash比较值
     */
    float IMAGE_COMPARE_HASH_CONFIDENCE_THRESHOLD = 0.75f;

    /**
     * 相机的Sensor的旋转误差值，取值为0-180，大于180表示忽略
     */
    int SENSOR_ORIENTATION_EVENT_DELTA = 20;

    /**
     * 传感器的SensorY的旋转误差值，取值为0-180，大于180表示忽略
     */
    double SENSOR_ORIENTATION_SENSOR_Y_DELTA = Math.PI / 6;

    /**
     * 拍照建议的最低亮度值
     */
    double SENSOR_LIGHT_LUMEN_MIN = 100;

    /**
     * 拍照建议的最高亮度值
     */
    double SENSOR_LIGHT_LUMEN_MAX = 500;

    /**
     * 40%图片的透明度
     */
    float IMAGE_SLIDE_TRANSPARENT_ALPHA = 0.5f;

    /**
     * 每行货架最多的照片数量，服务端支持最大60
     */
    int SLOT_MAX_PHOTO_NUM = 60;

    // 下面的参数，请不要修改
    float IMAGE_SLIDE_CROP_RATIO = 0.4f;

    boolean IMAGE_COMPARE_HASH_DEBUG_SAVE_IMAGES = false;
}
```

防止图片窜拍开关参数设置



```

package com.baidu.ai.utils;

public class CheckImageConfig {
    /**
     * 是否开启防窜拍
     * 默认开启
     */
    private volatile boolean piratelImageCheck = true;

    public void setPiratelImageCheck(boolean piratelImageCheck) {
        this.piratelImageCheck = piratelImageCheck;
    }

    public boolean getPiratelImageCheck() {
        return piratelImageCheck;
    }
}

```

## 价签识别服务

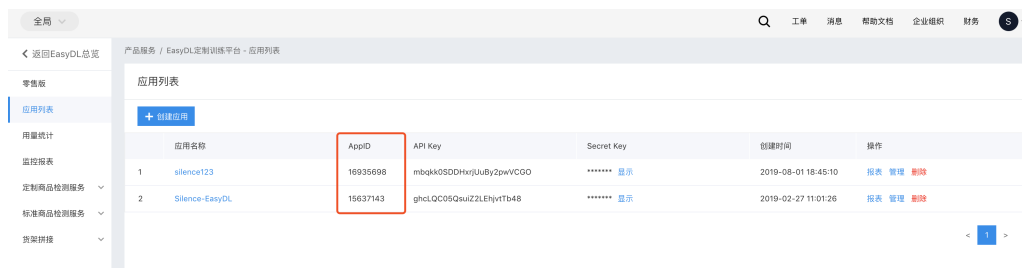
### 服务介绍

#### 简介

**价签识别**可识别货架和促销活动中的价签信息，可识别各个价签在图片中的像素位置，以及价签内商品名称和价格，可用于洞察商品在线下渠道分销的价格区间。

#### 使用须知

目前该服务处于**邀请使用**状态，请加入QQ群（群号:1009661589）联系管理员或是提交[合作咨询](#)申请使用权限，提供公司名称和在EasyDL零售版控制台[应用列表](#)创建应用的APPID，如下图：



#### 相关解决方案

- [数字化访销解决方案](#)

围绕快速消费品企业在线下渠道中的销量逻辑，提供基于AI技术的数字化访销解决方案，对访销过程的精细化管理，通过提升一线业务人员人效，最终实现销量的增长。

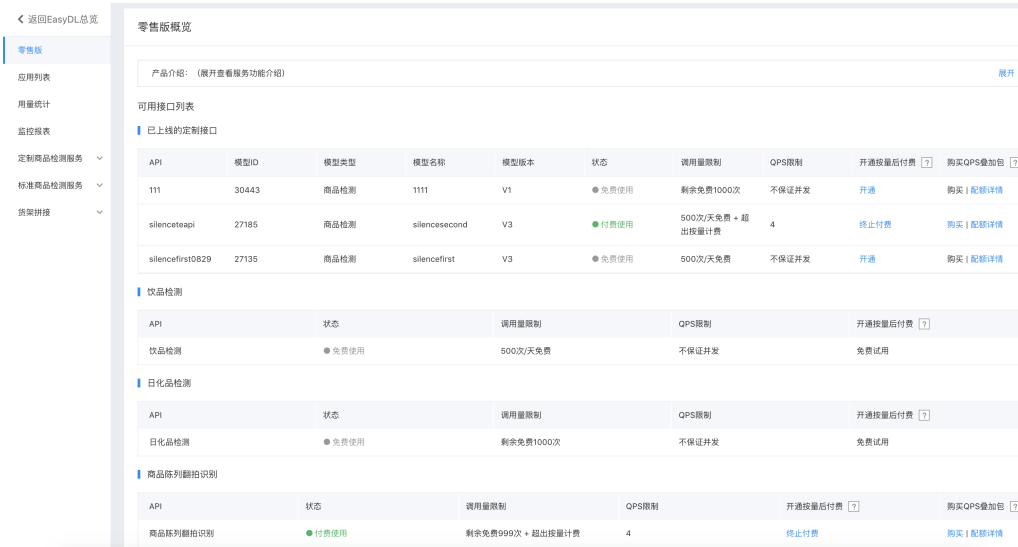
#### 使用须知

- 服务接口调用方法请见[API文档](#)

### 购买指南

#### 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。



### 🔗 计费方式

价签识别目前支持下列三种计费方式:

1. 按调用量后付费
2. 调用量次数包预付费
3. QPS叠加包预付费

### 🔗 价目表 - 按调用量后付费

#### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云EasyDL零售版控制台开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	2	服务器支持每秒处理2次查询

注：调用失败不计费

#### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
价签识别	累计1000次	1	服务器支持每秒处理1次查询

注：成功调用与失败调用均消耗免费额度

### 🔗 价目表 - 调用量次数包

如果业务上对调用次数有预估，可以选择购买单次调用价格更低的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	475 元	2	1年 (366天)
5万次	2,250 元	2	1年 (366天)
10万次	4,250 元	2	1年 (366天)
20万次	8,000 元	2	1年 (366天)
50万次	18,750 元	2	1年 (366天)
100万次	35,000 元	2	1年 (366天)
500万次	150,000 元	2	1年 (366天)

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

## 价目表 - QPS叠加包

开通付费后，免费QPS为2，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	100元/天
按月购买	2000元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## API文档

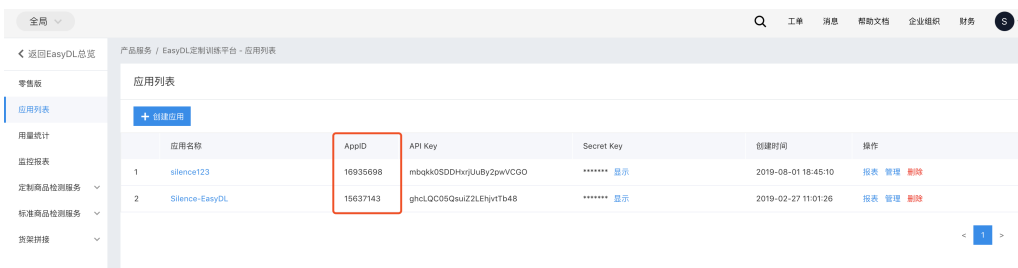
### API使用文档

#### 简介

本文档主要说明如何使用价签识别API

#### 获得使用权限

请提交[合作咨询](#)申请使用权限，提供公司名称和在EasyDL零售版控制台[应用列表](#)创建应用的APPID，如下图：



#### 接口鉴权

1. 进入EasyDL零售版的百度云控制台[应用列表页面](#)，如下图所示：



2. 如果还未创建应用, 请点击「创建应用」按钮进行创建。创建应用后, 参考[鉴权参考文档](#), 使用API Key(AK)和Secret Key(SK)获取 access\_token

请求说明

请求示例

价签识别请求URL : [https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/price\\_tag](https://aip.baidubce.com/rpc/2.0/easydl/v1/retail/price_tag)

HTTP 方法 : POST

URL参数 :

参数	值
access_token	通过API Key和Secret Key获取的access_token, 参考 <a href="#">鉴权认证机制文档</a>

Header如下 :

参数	值
Content-Type	application/json

提示 : 如果出现336001或336002的错误码很可能是因为请求方式错误, 请以json方式请求。

Body请求示例 :

```
{
  "image": "<base64数据>"
}
```

Body中放置请求参数, 参数详情如下 :

请求参数

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据, base64编码, 要求base64编码后大小不超过4M, 最短边至少15px, 最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部

提示 : image参数中“去掉头部”指的是图片经base64编码后的头部信息「data:image/jpeg;base64,」, 如下图所示 :



## 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	number	唯一的log id, 用于问题定位
words_result_num	否	number	检测到的价签数量
words_result	否	array(object)	识别的价签信息
+name	否	string	价签中的商品名称
+price	否	string	价签中的价格
+brief	否	string	价签中的其它文字信息
+location	否		价签的像素位置
++left	否	number	检测到的目标主体区域到图片左边界的距离
++top	否	number	检测到的目标主体区域到图片上边界的距离
++width	否	number	检测到的目标主体区域的宽度
++height	否	number	检测到的目标主体区域的高度

### 错误码

若请求错误, 服务器将返回的JSON文本包含以下参数:

- **error\_code**: 错误码。
- **error\_msg**: 错误描述信息, 帮助理解和解决发生的错误。

例如Access Token失效返回:

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（群号:1009661589）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336000	Internal error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336002	JSON不合法	入参格式或调用方式有误，比如缺少必要参数代码格式是否有误。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336003	Base64解码失败	图片格式有误或base64编码有误，请根据接口文档检查格式，base64编码请求时注意要去掉头部。有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336004	输入文件大小不合法	图片或音频、文本格式有误，请根据接口文档检查入参格式，有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队
336005	解码输入失败/分词错误	图片编码错误（非jpg,bmp,png等常见图片格式），请检查并修改图片格式
336006	缺失必要参数	image字段缺失（未上传图片）
336100	model temporarily unavailable	遇到该错误码请等待1分钟后再次请求，可恢复正常，若反复重试依然报错或有疑问请通过QQ群（群号:1009661589）或工单联系技术支持团队

## 牵拍识别服务

### 服务介绍

#### 简介

牵拍识别服务可对用户上传的数据进行疑似牵拍图（相似图）分组，用户可按照「人」、「门店」、「时间」定义需要识别的范围，服务按定义的范围返回有相似图存在的图片组。服务支持对上传的图片进行批量识别，牵拍识别支持两类任务：

单次任务：对本地上传的数据进行一次性的牵拍识别

周期任务：定期对调用定制模型云服务API的图片数据进行抓拍识别

## 与门店关联

当抓拍识别任务中定义的图片门店ID与门店库中的业务门店ID一致时，可以支持在门店管理的门店库中查看该门店最近一次抓拍任务的识别情况。

## API文档

### API调用方法

#### 接口总览

##### 1. 图片管理接口

接口名称	API URL	API描述
创建图集API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_dataset">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_dataset</a>	创建图集，获取图集ID
上传图片API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/upload">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/upload</a>	指定图集ID，将需要识别的图片上传至该图集
图集列表API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/dataset_list">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/dataset_list</a>	查询已创建的图集和状态

##### 图集状态 dataset\_status

状态	描述
Idle	空闲状态，可被使用的图集
Processing	正在运行或是排队中的任务涉及的图集

##### 2. 任务管理接口

接口名称	API URL	API描述
创建任务API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_task">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_task</a>	创建抓拍识别任务，获取任务ID
查询结果API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/query">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/query</a>	指定任务ID，查询任务状态和结果
任务列表API	<a href="https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/task_list">https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/task_list</a>	查询已创建的任务和状态

##### 任务状态 task\_status

状态	描述
Queued	排队中的任务
Running	正在运行的任务
Success	运行成功的任务
Failure	运行失败的任务

#### 接口调用流程

##### 1. 创建图集，获得图集ID

创建图集API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_dataset](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_dataset)

##### 2. 指定图集ID，把要识别牵拍的图片上传到该图集

上传图片API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/upload](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/upload)

##### 3. 创建任务，指定图集ID，对该图集做抓拍识别，获取任务ID

创建任务API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_task](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_task)

##### 4. 指定任务ID，轮询任务结果，可以每隔一定时间调用API查询任务结果，比如10s查询一次，任务完成后可以查到结果

查询结果API：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/query](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/query)

另外，可通过图集列表API和任务列表API查询所有图集和任务。

## 接口调用方法

### 创建图集API

#### 1.接口描述

创建图集，用于存放需要识别牵拍的图片

#### 2.请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_dataset](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_dataset)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

无需上传参数

#### 3.返回说明

#### 返回参数

参数名称	是否必须	数据类型	含义
log_id	是	int	唯一的log id，用于问题定位
dataset_id	是	int	图集ID
dataset_status	是	string	图集状态

### 上传图片API

#### 1.接口描述

为指定图集上传需要识别牵拍的图片。仅有Idle状态的图集，可以上传图片，对Running状态的图集的上传图片报错。

#### 2.请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/upload](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/upload)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数



参数名称	是否必须	数据类型	取值范围	含义
dataset_id	是	int	-	指定上传图片的数据集id
image	是	string	-	图像数据, base64编码, 要求base64编码后大小不超过4M, 最短边至少15px, 最长边最大4096px,支持jpg/png/bmp格式 注意请去掉头部
image_name	是	string	-	图片名称或编码, 请保证该字段在创建任务时定义的查牵拍范围内的唯一性
image_people	是	string	-	该图片的人员信息
image_store	是	string	-	该图片的网点信息, 如ID
image_time	是	string	-	该图片的拍摄时间, 格式为"yyyy-MM-dd hh:mm:ss"

提示：当一张图片的image\_name、image\_people、image\_store、image\_time全都一致时，判为重复图片，后上传的直接忽略。

### 3. 返回说明

#### 返回参数

参数名称	是否必须	数据类型	含义
log_id	是	int	唯一的log id, 用于问题定位
dataset_status	是	string	任务状态

#### 图集列表API

##### 1. 接口描述

按条件查询创建的图集信息

##### 2. 请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/dataset\\_list](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/dataset_list)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

#### 请求参数

参数名称	是否必需	参数类型	描述
dataset_ids	否	array[string]	只返回指定id的任务信息，最多返回前200个dataset_id的结果
begin_time	否	number	时间戳，只返回begin_time以后创建的任务信息
end_time	否	number	时间戳，只返回end_time之前创建的任务信息

### 3. 返回说明

#### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	int	唯一的log id, 用于问题定位
dataset_info	否	array[dict]	任务列表, 最多返回前200个dataset_id的结果
+dataset_id	否	string	任务id
+dataset_status	否	string	任务状态
+create_time	否	int	时间戳, 图集创建时间
+image_num	否	int	图集中含有的图片数量

## 创建任务API

### 1.接口描述

创建抓拍识别任务。

### 2.请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/create\\_task](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/create_task)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必需	数据类型	取值范围	含义
dataset_id	是	int	-	指定需要进行识别抓拍的图集ID
threshold	否	array[float]	每个元素取值范围为0.60~1.00	例如：[0.8,0.95]。可精确到小数点后两位。阈值为界定两张图视为抓拍的score的阈值，当score大于threshold时，则判定为疑似抓拍结果返回。不填，默认为0.8
task_people	否	boolean	“True”、“False”	区分不同人员，在同一人员拍摄的图片范围内，识别抓拍。True为区分，False为不区分；不填，默认为False
task_store	否	string	“undefined”、“within”	“undefined”表示不区分网点；within表示区分不同网点，在同一网点的图片范围内，识别抓拍；不填，默认为“undefined”
task_time	否	string	“undefined”、“year”、“month”、“day”	“undefined”表示不区分年份；“year”表示区分不同年份，在同一年份内的图片范围内，识别抓拍；“month”表示区分不同月份（不同年同月，算作不同月），在同一个月份内的图片范围内，识别抓拍；“day”表示区分不同日子（不同年或不同月的同一日，算作不同日），在同一天内的图片范围内，识别抓拍。不填，默认为“undefined”

### 3.返回说明

#### 返回参数

参数名称	是否必须	数据类型	含义
log_id	是	int	唯一的log id，用于问题定位
task_id	否	string	新建任务对应的id
task_status	否	string	任务状态

## 查询结果API

### 1.接口描述

查询指定任务的结果。

### 2.请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/query](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/query)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必须	数据类型	取值范围	含义
task_id	是	string	-	任务对应的id

### 3.返回说明

#### 返回参数

参数名称	是否必须	数据类型	含义
log_id	是	int	唯一的log id，用于问题定位
task_status	否	string	任务状态
task_results_url	否	dict	dict的key为阈值，value为结果文件URL，文件格式为json。url每次查询随机生成，有效期7天

#### 文件内的参数

参数名称	是否必须	数据类型	含义
task_results	否	array[dict]	任务结果
+re_people	否	string	当创建任务「task_people」为true时，返回人员信息，与用户上传时传上的「img_people」一致；当「task_people」为false时，返回-1
+re_store	否	string	当创建任务「task_store」为"within"时，返回人员信息，网点信息，与用户上传时传上的「image_store」一致；当「task_store」为"undefined"时，返回-1
+re_year	否	int	当创建任务「task_time」不为"undefined"时，返回年份，与用户上传时传上的「image_time」中的"YY"一致；当「task_time」均为"undefined"时，返回-1
+re_month	否	int	当创建任务「task_time」为"month"或"day"时，返回月份，与用户上传时传上的「image_time」中的"MM"一致；当「task_time」为"undefined"或"year"时，返回-1
+re_day	否	int	当创建任务「task_time」为"day"时，返回日信息，与用户上传时传上的「image_time」中的"DD"一致；当「task_time」不为"day"时，返回-1
+img_group	否	array[dict]	包含的疑似相似图的信息
++group_id	否	int	相似图组的组ID
++img_name	否	array[string]	相似图组内的包含的所有图片名称，与用户上传时传上的「image_name」一致

## 任务列表API

### 1.接口描述

按条件查询范围内的任务状态。

### 2.请求说明

HTTP 方法：POST

接口URL：[https://aip.baidubce.com/rpc/2.0/ai\\_custom\\_retail/v1/tasks/image\\_similar/task\\_list](https://aip.baidubce.com/rpc/2.0/ai_custom_retail/v1/tasks/image_similar/task_list)

URL参数：

参数	值
access_token	通过API Key和Secret Key获取的access_token，参考 <a href="#">鉴权认证机制文档</a>

Header如下：

参数	值
Content-Type	application/json

Body中放置请求参数，参数详情如下：

### 请求参数

参数名称	是否必需	参数类型	描述
task_ids	否	array[string]	只返回指定id的任务信息，最多返回前200个task_id的结果
begin_time	否	number	时间戳，只返回begin_time以后创建的任务信息
end_time	否	number	时间戳，只返回end_time之前创建的任务信息

### 3.返回说明

#### 返回参数

参数名称	是否必需	参数类型	描述
log_id	是	number	唯一的log id，用于问题定位
tasks_info	否	array[dict]	任务列表，最多返回前200个task_id的结果
+task_id	否	string	任务id
+task_status	否	string	任务状态
+create_time	否	number	时间戳，任务创建时间

## 错误码

### 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- **error\_code**：错误码。
- **error\_msg**：错误描述信息，帮助理解和解决发生的错误。

例如Access Token失效时，接口返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码列表如下：

error_code 错误码	error_msg 错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（1009661589）或工单联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超限额
6	No permission to access data	无权限访问该用户数据
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM鉴权失败
15	app not exists or create failed	应用不存在或者创建失败
17	Open api daily request limit reached	每天请求量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（1009661589）联系群管手动提额
18	Open api qps request limit reached	QPS超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
19	Open api total request limit reached	请求总量超限额，已上线计费的接口，请直接在控制台开通计费，调用量不受限制，按调用量阶梯计费；未上线计费的接口，请通过QQ群（649285136）联系群管手动提额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
336200	internal error	内部错误，如果持续出现此类错误，请通过QQ群（群号:1009661589）或工单联系技术支持团队。
336201	unknown task id	未知的任务id
336202	invalid param: 'param_name'	请求参数'param_name'的参数值不合法
336203	missing param: 'param_name'	请求参数'param_name'缺失
336206	invalid base64	加图操作：错误的base64图片编码
336207	failed loading image	加图操作：加载图片失败
336208	invalid image format	加图操作：不支持的图片格式，支持格式: bmp、jpg、jpeg、png
336209	invalid image shape	加图操作：不支持的图片形状，图片长宽需在[15, 4096]之间
336210	invalid image size	加图操作：不支持的图片大小，图片大小不超过4M
336212	invalid json	请求数据格式不正确
336215	unknown dataset id	未知的dataset id
336216	dataset is busy	dataset正在进行抓拍任务，暂时无法上传图片
336217	dataset is empty	dataset为空，需上传图片后才可触发抓拍任务

## 零售版常见问题

### 🔗 训练相关问题

为什么建议每个SKU至少出现在20张实景图图中？

上传的实景图，只有标注过的图片会被训练，所有训练的图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，如果标注的训练数据不足，可能会导致某个SKU的精确度远低于其它SKU，或是训练结果出现mAP、精确率、召回率全都为0的情况。

### 模型的训练结果是如何得到的？

上传的实景图，只有标注过的图片会被训练，所有训练图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，训练集训练出的模型去对测试集进行检测，检测得到的结果跟人为标注的结果进行对比，得到页面显示的mAP，精确率和召回率。

### 为什么同样的数据集每次训练出来的结果会不一样？

上传的实景图，只有标注过的图片会被训练，所有训练的图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，由于训练数据和测试数据每次都是随机抽取的，所以同样的数据集每次训练出来的结果会不一样。

### SKU单品图是用来做什么的？

SKU单品图用来降低实景图即训练数据采集和标注成本的。为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且单品图无需进行任何标注，即可让模型学习到这个SKU各个角度的样子。由百度提供的SKU预置了50张左右的单品图，绝大多数情况下无需再自行上传单品图。

### SKU单品图需不需要标注？

SKU单品图不要标注，只需要参考「SKU单品图数据要求」文档采集并上传至相应的SKU即可。

### 模型训练失败怎么办？

如果遇到模型训练失败的情况，请直接加入官方QQ群（群号:1009661589）咨询解决。

## 🔗 模型校验相关问题

### 如何校验模型的效果？

校验模型的目的是验证模型效果是否达到业务需求和找到模型效果最优的阈值，步骤如下两点：

1. 初步校验：模型训练好后，可以使用「[校验模型](#)」功能，在页面上提交几张没有被用于训练且从实际业务场景中采集到的图片，调整阈值查看结果，找到校验结果最优的阈值范围。如果无论什么阈值都无法满足您的业务需求，则需要继续优化模型，可以针对校验中发现问题参考[模型效果优化相关问题](#)对模型进行调优。
2. 批量校验：在初步校验后，得到校验结果最优的阈值范围，申请发布模型，发布成功后调用服务接口进行批量校验，找到校验结果最优的阈值。调用接口的时候可以通过threshold这个参数设置阈值，threshold可以精确度小数点后2位。

## 🔗 模型上线相关问题

### 希望加急上线怎么处理？

加入官方QQ群（群号:1009661589）咨询群管高优审核。

### 每个账号可以上线几个模型？是否可以删除已上线的模型？

每个账号最多申请发布十个模型，已上线模型无法删除。

## 🔗 SKU相关问题

### 每个账号允许创建多少个SKU

每个账号默认允许创建的SKU数量为50个，如果需要增加SKU数量，请加入官方QQ群（群号:1009661589）咨询解决。

## 🔗 收费相关问题

### 接口上线后是否收费？调用量不够怎么办？

目前接口是限量免费使用的原则，上线模型后可以免费获得1000次/天，qps=2的调用限额（QPS为每秒请求数）。如需调用更多次数，请在控制台中开通付费，开通付费后无调用次数限制，按调用成功次数收费，QPS免费提高至4，费用请参考[服务价格文档](#)。

## 🔗 模型效果优化相关问题

### 如何正确标注

- 单独框选要识别的SKU，不可同时框选多个目标
- 完整并仅仅框选要识别的SKU

- 标注框不要框选到其它SKU或是价目标签等非要识别的SKU的干扰信息
- 在实景图中出现的所有要识别的SKU必须全部标注

### 部分SKU识别效果太差

上传SKU单品图能够有效提高识别效果，上传要求如下：

- 图片像素足够高，不能模糊不清
- 单品图背景颜色必须为纯色且与SKU主体颜色不相似
- 角度、光线覆盖到实际检测场景中SKU所有可能出现的情况
- 要识别的SKU建议至少出现在20张实景图中，并且正确标注

### 错误示例中检查出漏标的情况

- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调低阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值
- 在实景图出现的该SKU必须全部标注，不能存在漏标注的情况
- 要识别的SKU建议至少出现在20张实景图中，并且正确标注

### 某些SKU在特定的一些角度下识别度很低

- 添加出现这些角度的SKU的实景图到训练的实景图集中，并正确标注
- 为这些SKU添加这些角度的单品图

### 错误示例中出现一个大框框选住多个SKU的情况

- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调高阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值

### 训练结果精确度很高，但是校验或是在实际场景调用时结果不好

- 要识别的SKU建议至少出现在20张实景图中，并且正确标注
- 参与训练的实景图集一定要包含和实际商品检测场景环境一致的图片，每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调整阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值
- 用于训练、校验、实际检测的图片，像素都要足够高，不能出现模糊不清的情况

### mAP、精确率、召回率全都为0

- 要识别的各个SKU推荐至少出现在20张以上实景图中，并且正确标注

### 误识别到相似度极高的非目标SKU

- 提交工单或是加入官方QQ群（群号:1009661589），将这些相似度极高的SKU基本信息（SKU名称、品牌、规格、包装）反馈给我们

## 🔗 其他问题

### 模型能否支持私有化部署？

目前定制商品检测服务提供在线调用API，如需私有化部署，可以提交工单咨询或是加入官方QQ群（群号:1009661589）联系管理员反馈

### 申请发布模型审核不通过都是什么原因？

可能原因如下：

1. 经过电话沟通当前模型存在问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝
2. 电话未接通且模型效果较差，会直接拒绝。如果需要申诉，加入官方QQ群（群号:1009661589）咨询群管解决

## 零售版服务介绍



## 简介

EasyDL是百度大脑中的一个定制化训练和服务平台，EasyDL零售版是EasyDL针对零售场景推出的行业版。EasyDL零售版提供两种服务，分别为定制商品检测服务和货架拼接服务。

**定制模型服务**是EasyDL零售版的一项服务，专门用于训练货架合规性检查、自助结算台、无人零售货柜等场景下的定制化AI模型，训练出的模型将以API的形式为客户提供服务。该服务包含以下2种定制模型：

### 1. 商品检测模型

- **适用场景：**适用于适用于货架、端架、挂架等场景的商品陈列规范核查，支持识别商品基本信息，陈列顺序、层数、场景，统计排面数量和占比
- **服务功能：**
  - **商品基本信息识别：**商品的名称、品牌、规格、编号；商品在图片中的坐标位置；商品识别的置信度
  - **商品陈列层数识别：**商品陈列所在货架层数和货架总层数；商品陈列顺序；货架是否拍摄完整判断
  - **商品陈列场景识别：**场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台
  - **商品排面占比统计：**商品的排面数及排面占比；每层货架可识别商品数量及未知商品数量；货架的总空位数、每层货架空位数及货架利用率
  - **商品陈列翻拍识别：**识别商品陈列照片是对手机屏幕翻拍的可能性

### 2. 地堆检测模型

- + **适用场景：**适用于堆箱、堆头、地龙等场景的商品陈列规范核查，支持识别商品基本信息，可视商品计数，纵深商品计数和占地面积
- + **服务功能：**
  - + **商品基本信息识别：**商品的名称、品牌、规格、编号；商品在图片中的坐标位置；商品识别的置信度；陈列顺序；可视商品计数，纵深商品计数和占地面积
  - + **商品陈列场景识别：**场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、卧式冰柜、冷风柜、地堆、割箱、地龙、小端架、吧台
  - + **商品陈列翻拍识别：**识别商品陈列照片是对手机屏幕翻拍的可能性

**货架拼接服务**基于百度EasyDL深度学习算法，支持将多个货架局部图片或视频，组合为完整货架图片。同时支持输出在完整货架图中的商品检测结果，包含SKU的名称和数量，适用于需要在长货架进行商品检测的业务场景。

**翻拍识别服务**能够识别出通过手机翻拍出的商品陈列照片，比如商品货架陈列图片和地堆商品陈列图片，可降低人工审核人力，高效审核零售业务中通过翻拍原有图片来造假图片。

**价签识别服务**能够识别货架和促销活动中的价签信息，可识别各个价签在图片中的像素位置，以及价签内商品名称和价格，可用于洞察商品在线下渠道分销的价格区间。

## 功能介绍

### 定制商品检测服务

#### AI模型训练平台

专门用于定制货架合规性检查、自助结算台、无人零售货柜等零售场景下识别商品的高精度AI模型。

#### 全可视化操作

所有模型训练相关的操作都可以在网页上进行，无需编程，仅需五步即可部署定制化AI模型。

#### 预置的商品库

预置近千种商品单品图可供客户在创建SKU时选择，用于合成训练数据，极大降低了训练数据采集和标注成本。

#### 可自定义商品

客户可根据业务需求创建属于自己的商品，商品信息支持完全自定义，充分满足客户定制化需求。

#### 全面的商品信息

商品基本信息识别

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

商品陈列层数识别

接口支持识别商品陈列所在货架层数，货架类型支持：货架、端架和立式冰柜内货架

商品陈列场景识别

接口支持识别商品陈列的场景，场景类型支持：普通货架、挂钩货架、斜口篮货架、端架、立式冰柜、地堆、割箱、地龙、小端架、吧台

商品排面占比统计

接口支持统计商品排面数/占比、未识别商品数、总空位数、每货架层的空位数及货架利用率

商品陈列翻拍识别

识别商品陈列照片是对手机屏幕翻拍的可能性

## 🔗 货架拼接服务

### • 拼接和商品检测相结合

支持将多个货架局部图片或视频，组合为完整货架图片，并支持输出在完整货架图中的商品检测结果，包含SKU的名称和数量。

### • 丰富的服务方式

支持三种服务方式：云服务API、完全开源的SDK以及可以直接体验的手机APP。

## 🔗 适用场景

### • 货架合规性检查

精准识别出货架、冰柜和端架上陈列商品的数量和种类，为品牌商分析陈列排面占比，重点SKU分销率、缺货率、合格率提供数据支撑。

### • 互动营销

训练定制的商品识别，实现对C端用户提交的商品图片进行识别，配合游戏规则完成闯关/抽奖式的互动营销。

## 🔗 技术优势

### • 免费训练与测试

平台提供大量免费的GPU训练资源，及每天500次免费调用量，用于模型迭代和效果验证，有效降低项目开发和测试成本。

### • 高可用模型效果

针对零售场景专项算法调优，结合图像合成与增强技术提升模型泛化能力，模型准确率可达97%+，保证模型在生产环境中具有高可用性。

### • 预置模型和数据

平台提供直接可用的商品检测API，覆盖常见商品品类；提供大量预置单品图数据，可用于训练定制模型，有效提升项目落地效率。

### • 企业级安全保障

数据加密与隔离，完善的服务调用鉴权，为客户的数据和模型提供企业级安全保障。

### • 功能完善且丰富

全面覆盖各类零售场景的商品识别需求，应对不同场景的业务需求提供多种可选服务类型。

## 🔗 与EasyDL物体检测的相同点和不同点

EasyDL零售版是EasyDL专门针对零售场景下识别商品推出的版本，相比于物体检测模型，零售版更贴合快消零售场景下的业务需求，专门用于训练货架合规性检查、自助结算台、无人零售货架等场景下的定制化商品检测AI模型，训练出的模型可发布成云服务API，服务支持四种功能：商品基本信息识别、商品陈列层数识别、商品陈列场景识别和商品排面占比统计，适用于识别货架中的商品信息，商品计数和陈列顺序等，辅助货架商品陈列合规检查，如缺货率、陈列情况等。

## 相同点

同为检测模型，接口支持返回目标物体的名称和物体在图片上的位置。

## 不同点

- 模型算法不同：零售版的模型算法专门根据零售行业的场景和业务需求做了专项优化，基于百度大脑大规模零售数据预训练，并利用商品增强合成技术将SKU单品图合成实景货架图，有针对性的提高了训练商品检测模型的精确度。
- 训练数据不同：零售版的数据除了需要标注的实景业务图片外，支持为每个SKU标签上传单品图。SKU单品图用来降低实景图即训练数据采集和标注成本的。为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分的成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且**单品图无需进行任何标注**，即可让模型学习到这个SKU各个角度的样子。**合成图片过程在训练阶段自动完成，无需操作操作和进行标注**。EasyDL零售版平台预置了近千种商品，每个商品预置了50张左右的单品图，绝大多数情况下无需再自行上传单品图。
- 云服务API功能不同：零售版云服务API支持四种功能：商品基本信息识别、商品陈列层数识别、商品陈列场景识别和商品排面占比统计；物体检测云服务API仅支持商品基本信息识别。

## 购买指南

### ☞ 开通付费及购买服务

EasyDL零售版的各项服务的「开通付费」、「购买QPS叠加包」、「购买调用量次数包」、「关闭付费」等操作均在[EasyDL零售版控制台](#)进行，只需在相应需要付费使用的接口位置，跟随页面提示完成后续充值及付费，即可完成。

API	模型ID	模型类型	模型名称	模型版本	状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
已上线的定制接口									
111	30443	商品检测	1111	V1	● 免费使用	剩余免费1000次	不保证并发	开通	购买   配账详情
silenceapi	27185	商品检测	silencesecond	V3	● 付费使用	500次/天免费 + 超出按量计费	4	终止付费	购买   配账详情
silencefirst0829	27135	商品检测	silencefirst	V3	● 免费使用	500次/天免费	不保证并发	开通	购买   配账详情
饮品检测									
API					状态	调用量限制	QPS限制	开通按量后付费	
饮品检测					● 免费使用	500次/天免费	不保证并发	免费试用	
日化品检测									
API					状态	调用量限制	QPS限制	开通按量后付费	
日化品检测					● 免费使用	剩余免费1000次	不保证并发	免费试用	
商品陈列翻拍识别									
API					状态	调用量限制	QPS限制	开通按量后付费	购买QPS叠加包
商品陈列翻拍识别					● 付费使用	剩余免费999次 + 超出按量计费	4	终止付费	购买   配账详情

### ☞ 定制商品检测服务

### ☞ 价目表 - 按调用量后付费

定制商品检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

#### 1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

#### 2. 商品陈列层数识别（可选）

接口支持识别商品陈列所在货架层数，货架类型支持：货架、端架和立式冰柜内货架

#### 3. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

#### 4. 商品排面占比统计（可选）

接口支持统计商品排面数/占比、未识别商品数、空位数及货架利用率

#### 5. 商品陈列翻拍识别（可选）

识别商品陈列照片是对手机屏幕翻拍的可能性

## 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用三项服务，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列层数识别和商品陈列场景识别两项服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见[服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

月调用量（万次）	单次调用价格（元）	QPS限制	说明
0<月调用量<=15	0.009	4	服务器支持每秒处理4次查询
15<月调用量<=150	0.008	4	服务器支持每秒处理4次查询
150<月调用量	0.007	4	服务器支持每秒处理4次查询

- 商品陈列层数识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.04	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品排面占比统计（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.02	4	服务器支持每秒处理4次查询

- 商品陈列翻拍识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.05	4	服务器支持每秒处理4次查询

注：调用失败不计费

## 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制商品检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制商品检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

## 费用举例

从2019-3-1至2019-3-31，定制商品检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

- 商品基本信息识别的费用为43,650元，明细如下：

前15万次落入0~15w阶梯，单次调用0.009元/次，费用为1,350元；

中间15万~150万次落入15~150w阶梯，单次调用0.008元/次，费用为10,800元；

最后150万~600万次落入大于150w阶梯，单次调用0.007元/次，费用为31,500元；

共计43,650元

2. 商品陈列层数识别的费用为360,000元，明细如下：

月调用量为600万次，单次调用0.04元/次，费用为240,000元

3. 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计319,650元。

#### 🔗 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1050元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制商品检测API的所有服务功能均有效

#### 🔗 定制地堆检测服务

#### 🔗 价目表 - 按调用量后付费

定制地堆检测模型云服务API支持以下三项功能，启停服务功能的方法请见 [服务功能文档](#)。

1. 商品基本信息识别（必选）

接口支持识别商品信息（商品名称、品牌、规格）、编号和置信度

2. 商品陈列场景识别（可选）

货架场景：货架、端架（小方货架）；冰柜场景：立式冰柜；地堆场景：堆箱、割箱、地龙

3. 商品陈列翻拍识别（可选）

识别商品陈列照片是对手机屏幕翻拍的可能性

#### 付费调用

每个模型发布的云服务API享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费，开通后即可开始计费使用。

注：为保证老用户可正常使用商品陈列场景识别服务功能，对于服务功能上线前已经开通付费的定制模型API，默认对各项服务功能开通付费。商品陈列场景识别服务功能，需在EasyDL零售版模型训练后台启动后方可使用，启动方法请见 [服务功能文档](#)，启动后发生调用才会按实际调用次数进行收费。

选择开启的功能不同，单次接口调用价格不同，详情如下：

- 商品基本信息识别（必选），按月调用量阶梯计费，调用单价按照自然月累积调用量所落阶梯区间而变化。月初，上月累积的调用量清零，重新开始累积本月调用量。

单次调用价格（元）	QPS限制	说明
0.016	4	服务器支持每秒处理4次查询

- 商品陈列场景识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.006	4	服务器支持每秒处理4次查询

- 商品陈列翻拍识别（可选），单次调用额外收取费用

单次调用价格（元）	QPS限制	说明
0.05	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

EasyDL零售版各项服务都具有免费调用额度，开通付费后，免费调用额度仍保留，EasyDL零售版的定制地堆检测服务免费额度如下：

服务名称	免费调用额度	QPS限制
定制地堆检测服务	累计1000次	1~2，服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 费用举例

从2019-3-1至2019-3-31，定制地堆检测API在三种服务功能都开启的情况下，月调用量为600万次（已除去免费额度），那么费用如下：

- 商品基本信息识别的费用为96,000元，明细如下：

月调用量为600万次，单次调用0.016元/次，费用为96,000元

- 商品陈列场景识别的费用为，明细如下：

月调用量为600万次，单次调用0.006元/次，费用为36,000元

综上，三月费用合计132,000元。

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	60元/天
按月购买	1200元/月

购买QPS叠加包前需保证已开通按量后付费或购买次数包

购买的QPS叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

购买QPS叠加包提升的额度对定制地堆检测API的所有服务功能均有效

### 翻拍识别服务

#### 价目表 - 按调用量后付费

#### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	4	服务器支持每秒处理4次查询

注：调用失败不计费

### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
商品陈列翻拍识别	累计1000次	1~2	服务器支持每秒处理1~2次查询

注：成功调用与失败调用均消耗免费额度

### 价目表 - 调用量次数包

如果对调用次数有预估，可以选择购买**单次调用价格更低**的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	490 元	4	1年
10万次	4,800 元	4	1年
100万次	45,000 元	4	1年
500万次	212,500 元	4	1年
1000万次	420,000 元	4	1年
2000万次	800,000 元	4	1年

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

### 价目表 - QPS叠加包

开通付费后，免费QPS为4，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	50元/天
按月购买	1200元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

### 价签识别服务

#### 价目表 - 按调用量后付费

#### 付费调用

每个账户享有累计1000次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

调用量	价格	QPS限制	说明
每次调用	0.05元	2	服务器支持每秒处理2次查询

注：调用失败不计费

### 免费额度

每个账号享有一定量免费调用额度，如下表：

云服务API	免费调用额度	QPS限制	说明
价签识别	累计1000次	1	服务器支持每秒处理1次查询

注：成功调用与失败调用均消耗免费额度

### 价目表 - 调用量次数包

如果业务上对调用次数有预估，可以选择购买**单次调用价格更低**的次数包，价格如下：

规格	价格	QPS限制	有效期
1万次	475 元	2	1年 (366天)
5万次	2,250 元	2	1年 (366天)
10万次	4,250 元	2	1年 (366天)
20万次	8,000 元	2	1年 (366天)
50万次	18,750 元	2	1年 (366天)
100万次	35,000 元	2	1年 (366天)
500万次	150,000 元	2	1年 (366天)

购买后不可退款，次数包使用完后，开始按调用量每次0.05元收取费用

### 价目表 - QPS叠加包

开通付费后，免费QPS为2，如果有更多的并发请求需要，可以根据业务需求按天或按月购买QPS叠加包，价格如下：

购买方式	每QPS价格
按天购买	100元/天
按月购买	2000元/月

购买 QPS 叠加包需保证已开通按量后付费或购买次数包

购买的 QPS 叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

### 货架拼接服务

货架拼接服务支持按任务数后付费、任务次数包预付费和并发任务叠加包预付费三种计费方式。

### 价目表 - 按任务数后付费

#### 付费调用

每个账户享有累计200次免费调用额度，免费额度用尽后，请在百度云[EasyDL零售版控制台](#)开通计费后继续使用，开通后即可开始计费使用，价格如下：

任务数	价格（元）	并发任务数限制	说明
每次拼接任务	0.2	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务启动后失败和运行前终止不计费，任务成功和运行后终止会计费用



## 免费额度

每个账号享有一定量免费调用额度，如下表：

服务	免费任务额度	并发任务数限制	说明
货架拼接	累计200次	1	只允许一个拼接任务在运行，超出限制的任务排队等待

注：任务成功与失败调用均消耗免费额度

## 价目表 - 任务次数包

如果对拼接任务次数有预估，可以选择购买**单次任务价格更低**的次数包，价格如下：

规格	价格	并发任务数限制	有效期
1千次	200 元	1	1年
1万次	1,900 元	1	1年
10万次	18,000 元	1	1年
100万次	150,000 元	1	1年
500万次	600,000 元	1	1年

**购买后不可退款**，任务次数包使用完后，开始按调用量每个任务0.2元收取费用

## 价目表 - 并发任务叠加包

开通付费后，并发任务数限制为1，如果有更多的并发请求需要，可以根据业务需求按天或按月购买并发任务叠加包，价格如下：

购买方式	每并发任务价格
按天购买	2元/天
按月购买	40元/月

购买 并发任务叠加包需保证已开通按量后付费或购买任务次数包

购买的并发任务叠加包自起始日0点生效，至停止日24点失效。若起始日为本日，则是从下单成功时当即生效

## 余额不足提醒与欠费处理

### 余额不足提醒

根据您的历史的账单金额，判断您的账户余额（含可用代金券）是否足够支付未来的费用，若不足以支付，系统将在欠费前三天、两天、一天发送续费提醒短信，请您收到短信后及时前往控制台财务中心[充值](#)。

### 欠费处理

- 北京时间整点检查您的账户余额是否足以支付本次账单的费用（如北京时间11点整检查账户余额是否足以支付10点至11点的账单费用），若不足以支付，即为欠费，欠费时系统会发送欠费通知。
- 欠费后您开通付费的产品将进入欠费状态，只能使用每日的免费额度，超过额度的请求系统将不再响应，且不再保证并发处理。

## 数据看板

新增门店数据

## 零售版常见问题

### 训练相关问题

### 为什么建议每个SKU至少出现在20张实景图图中？

上传的实景图，只有标注过的图片会被训练，所有训练的图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，如果标注的训练数据不足，可能会导致某个SKU的精确度远低于其它SKU，或是训练结果出现mAP、精确率、召回率全都为0的情况。

### 模型的训练结果是如何得到的？

上传的实景图，只有标注过的图片会被训练，所有训练图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，训练集训练出的模型去对测试集进行检测，检测得到的结果跟人为标注的结果进行比对，得到页面显示的mAP，精确率和召回率。

### 为什么同样的数据集每次训练出来的结果会不一样？

上传的实景图，只有标注过的图片会被训练，所有训练的图片中，系统会随机抽取70%作为训练集，剩余的30%作为测试集，由于训练数据和测试数据每次都是随机抽取的，所以同样的数据集每次训练出来的结果会不一样。

### SKU单品图是用来做什么的？

SKU单品图用来降低实景图即训练数据采集和标注成本的。为了让模型能够完整地识别一个SKU，需要训练的图片中出现这个SKU的各个角度的样子，这意味着需要从实际业务场景中采集大量的图片，并且进行大量的标注工作。为了降低这部分的成本，我们通过数据合成和增强技术，只需为SKU上传各个角度的单品图，且单品图无需进行任何标注，即可让模型学习到这个SKU各个角度的样子。由百度提供的SKU预置了50张左右的单品图，绝大多数情况下无需再自行上传单品图。

### SKU单品图需不需要标注？

SKU单品图不要标注，只需要参考「SKU单品图数据要求」文档采集并上传至相应的SKU即可。

### 模型训练失败怎么办？

如果遇到模型训练失败的情况，请直接加入官方QQ群（群号:1009661589）咨询解决。

## 🔗 模型校验相关问题

### 如何校验模型的效果？

校验模型的目的是验证模型效果是否达到业务需求和找到模型效果最优的阈值，步骤如下两点：

1. 初步校验：模型训练好后，可以使用「[校验模型](#)」功能，在页面上提交几张没有被用于训练且从实际业务场景中采集到的图片，调整阈值查看结果，找到校验结果最优的阈值范围。如果无论什么阈值都无法满足您的业务需求，则需要继续优化模型，可以针对校验中发现问题参考[模型效果优化相关问题](#)对模型进行调优。
2. 批量校验：在初步校验后，得到校验结果最优的阈值范围，申请发布模型，发布成功后调用服务接口进行批量校验，找到校验结果最优的阈值。调用接口的时候可以通过threshold这个参数设置阈值，threshold可以精确度小数点后2位。

## 🔗 模型上线相关问题

### 希望加急上线怎么处理？

加入官方QQ群（群号:1009661589）咨询群管高优审核。

### 每个账号可以上线几个模型？是否可以删除已上线的模型？

每个账号最多申请发布十个模型，已上线模型无法删除。

## 🔗 SKU相关问题

### 每个账号允许创建多少个SKU

每个账号默认允许创建的SKU数量为50个，如果需要增加SKU数量，请加入官方QQ群（群号:1009661589）咨询解决。

## 🔗 收费相关问题

### 接口上线后是否收费？调用量不够怎么办？

目前接口是限量免费使用的原则，上线模型后可以免费获得1000次/天，qps=2的调用限额（QPS为每秒请求数）。如需调用更多次数，请在控制台中开通付费，开通付费后无调用次数限制，按调用成功次数收费，QPS免费提高至4，费用请参考[服务价格文档](#)。

## 🔗 模型效果优化相关问题

### 如何正确标注

- 单独框选要识别的SKU，不可同时框选多个目标
- 完整并仅仅框选要识别的SKU
- 标注框不要框选到其它SKU或是价目标签等非要识别的SKU的干扰信息
- 在实景图中出现的所有要识别的SKU必须全部标注

#### 部分SKU识别效果太差

上传SKU单品图能够有效提高识别效果，上传要求如下：

- 图片像素足够高，不能模糊不清
- 单品图背景颜色必须为纯色且与SKU主体颜色不相似
- 角度、光线覆盖到实际检测场景中SKU所有可能出现的情况
- 要识别的SKU建议至少出现在20张实景图中，并且正确标注

#### 错误示例中检查出漏标的情况

- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调低阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值
- 在实景图出现的该SKU必须全部标注，不能存在漏标注的情况
- 要识别的SKU建议至少出现在20张实景图中，并且正确标注

#### 某些SKU在特定的一些角度下识别度很低

- 添加出现这些角度的SKU的实景图到训练的实景图集中，并正确标注
- 为这些SKU添加这些角度的单品图

#### 错误示例中出现一个大框框选住多个SKU的情况

- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调高阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值

#### 训练结果精确度很高，但是校验或是在实际场景调用时结果不好

- 要识别的SKU建议至少出现在20张实景图中，并且正确标注
- 参与训练的实景图集一定要包含和实际商品检测场景环境一致的图片，每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 参考模型列表中模型效果下的「完整评估结果」内的推荐阈值，校验时适当调整阈值后测试效果，找到合适的阈值，在调用API接口时用请求参数threshold设定合适的阈值
- 用于训练、校验、实际检测的图片，像素都要足够高，不能出现模糊不清的情况

#### mAP、精确率、召回率全都为0

- 要识别的各个SKU推荐至少出现在20张以上实景图中，并且正确标注

#### 误识别到相识度极高的非目标SKU

- 提交工单或是加入官方QQ群（群号:1009661589），将这些相似度极高的SKU基本信息（SKU名称、品牌、规格、包装）反馈给我们

#### 🔗 其他问题

##### 模型能否支持私有化部署？

目前定制商品检测服务提供在线调用API，如需私有化部署，可以提交工单咨询或是加入官方QQ群（群号:1009661589）联系管理员反馈

##### 申请发布模型审核不通过都是什么原因？

可能原因如下：

1. 经过电话沟通当前模型存在问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝

2. 电话未接通且模型效果较差，会直接拒绝。如果需要申诉，加入官方QQ群（群号:1009661589）咨询群管解决

# EasyDL桌面版使用说明（已下线）

## 产品简介

### 产品介绍

飞桨EasyDL桌面版是百度针对客户端开发的零门槛AI开发平台，可在离线状态通过本地资源完成包括数据管理与数据标注、模型训练、模型部署的一站式AI开发流程。

无需机器学习专业知识，通过 模型创建→数据上传→模型训练→模型发布 全流程可视化便捷操作，最快15分钟即可获得一个高精度模型。

针对有一定AI模型开发基础的开发者，飞桨EasyDL同时还提供了预置模型调参、Notebook建模两种建模方式，开发者可根据自身经验进行调整，以获得更适合特定场景的模型。

目前已支持训练图像分类、物体检查、实例分割、语义分割4种不同应用场景的模型。

### 应用场景

#### 图像分类：

- 图片内容检索：定制训练需要识别的各种物体，并结合业务信息展现更丰富识别结果
- 图片审核：定制图像审核规则，如训练直播场景中抽烟等违规现象
- 制造业分拣或质检：定制生产线上各种产品识别，进而实现自动分拣或者质检
- 医疗诊断：定制识别医疗图像，辅助医生肉眼诊断

#### 物体检测：

- 视频监控：如检测是否有违规物体、行为出现
- 工业质检：如检测图片里微小瑕疵的数量和位置
- 医疗诊断：如医疗细胞计数、中草药识别等

#### 实例分割：

- 专业检测：应用于专业场景的图像分析，比如在卫星图像中识别建筑、道路、森林，或在医学图像中定位病灶、测量面积等
- 智能交通：识别道路信息，包括车道标记、交通标志等

#### 语义分割：

- 自动驾驶：识别道路障碍、分割线等，辅助驾驶决策
- 智能分拣：包括工业零部件分拣、垃圾分拣等

## 功能介绍

飞桨EasyDL桌面版提供数据处理、模型训练、模型部署全流程的模型生产能力。

#### 数据处理：

提供针对图像的成熟标注模板及工具，便捷的为AI开发准备高质量训练数据，提供可视化管理能力，支持不同数据格式的导入、导出、查看

#### AotuDL建模：

为零AI开发基础的用户提供的建模方式，内置基于百度文心大模型的成熟预训练模型，可针对用户数据进行算法自动优化，助用户使用少量数据也能获得具备出色效果与性能模型

#### 预置模型调参建模：

为有一定AI开发基础的开发者提供预置模型调参建模方式，涵盖ResNet、YOLO、PicoDet、MaskRCNN等近30种网络类型，适配大部分场景，开发者只需选择合适的预训练模型以及网络，根据自身经验进行调整，以获得更适合特定场景的模型

#### Notebook建模：

集成了包括PaddleX、PaddleDetection、PaddleSeg、PaddleClas等端到端开发套件的轻量级IDE，用户可在该模块内进行代码编辑、调试等开发工作，快速高效的完成各类任务的实现，可对预置模型调参中的模型进行代码级优化

**模型部署：**

训练完成的模型可发布为在服务器、小型设备、专项适配硬件上直接部署的SDK，覆盖主流芯片与操作系统，充分满足不同业务场景对模型部署的要求

不同版本功能对比

产品分为标准版与高级版，标准版提供图像分类、物体检测、实例分割、语义分割场景下完整的模型生产能力，高级版在标准版的基础上在各环节进一步提供了便捷化的应用工具。

每位新用户下载并成功激活飞桨EasyDL后，将专享30天高级版免费试用权益

模块	功能点	功能明细	标准版	高级版
数据	数据集创建	创建图像分类、物体检测、实例分割、语义分割数据集	✓	✓
	数据导入	导入图像分类、物体检测、实例分割、语义分割已标注/未标注数据集	✓	✓
	数据标注	图像分类提供常规标注能力、批量标注能力	✓	✓
		物体检测提供常规标注能力	✓	✓
开发	AutoDL模式	实例分割、语义分割提供常规标注能力，自动识别轮廓标注能力	✓	✓
		发起图像分类、物体检测、实例分割、语义分割任务训练	✓	✓
		选择训练算法	✓	✓
		根据部署环境提供针对性训练算法	✓	✓
	预置模型调参模式	手动/自动设置数据增强策略		✓
		发起图像分类、物体检测、实例分割、语义分割任务训练		✓
		选择网络、预训练模型		✓
		手动/自动设置模型训练参数		✓
	Notebook模式	手动/自动设置数据增强策略		✓
		预置PaddleX、PaddleDetection、PaddleSeg、PaddleClas等端到端开发套件	✓	✓
		零代码开发模型转Notebook优化	✓	✓
		支持代码高亮，自动补全	✓	✓
可管理代码、数据、模型等类型的文件		✓	✓	
实时可视化CPU、内存、显卡、硬盘的运行数据		✓	✓	
评估报告	训练完成后自动生成模型评估报告	✓	✓	
任务校验	训练完成后支持发起模型校验	✓	✓	
模型	模型部署	导出模型源文件	✓	✓
		导出可直接用于服务器、小型设备、专项适配硬件上直接部署的SDK，覆盖主流芯片与操作系统		✓

**系统支持**

客户端安装系统要求

- **Windows**：Windows 10及以上版本（64 bit）
- **Mac**：macOS 10.11及以上版本（IntelCPU）
- **Linux**：Ubuntu 18.04及以上版本（64 bit）

CPU训练环境要求

请确保CPU芯片支持AVX指令集

GPU训练环境要求

如果您的计算机有NVIDIA® GPU，且需要使用GPU环境进行训练，请确保满足以下条件：

- **Windows 10/11**：驱动版本需527.41及以上，需安装 CUDA 12.0 与 cuDNN v8.9.0
- **Ubuntu 18.04/20.04**：驱动版本需525.60.13及以上，需安装 CUDA 12.0 与 cuDNN v8.9.0

#### 🔗 CUDA、cuDNN安装指南

您可参考NVIDIA官方文档了解CUDA和CUDNN的安装流程和配置方法，详见：

- **CUDA 安装指南**：<https://docs.nvidia.com/cuda/>
- **cuDNN安装指南**：<https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html>

## 下载与激活

#### 🔗 下载

访问飞桨EasyDL桌面版官网：<https://ai.baidu.com/easydl/paddle>，根据您的系统，下载对应的客户端。

#### 🔗 激活

每位新用户下载并成功激活飞桨EasyDL后，将专享30天高级版免费试用权益。高级版体验到期后，若仍需要使用高级版特有功能，可点击进入「[购买高级版](#)」。

完成高级版购买后，在客户端内输入获得的高级版序列号并联网激活，即可继续使用高级版功能。

#### 购买入口1



# 飞桨EasyDL

零门槛AI开发平台

购买入口2

使用基础版

激活高级版

1.首次激活基础版/高级版时请确保您的设备处于联网状态

2.基础版提供数据标注、模型训练与源文件导出等一站式AI开发能力,高级版在此基础上提供适配多种部署场景的SDK [了解基础版/高级版](#)

提示:您的高级版试用权益已到期,您可以继续使用基础版或激活高级版,高级版序列号请 [购买高级版](#) 获取



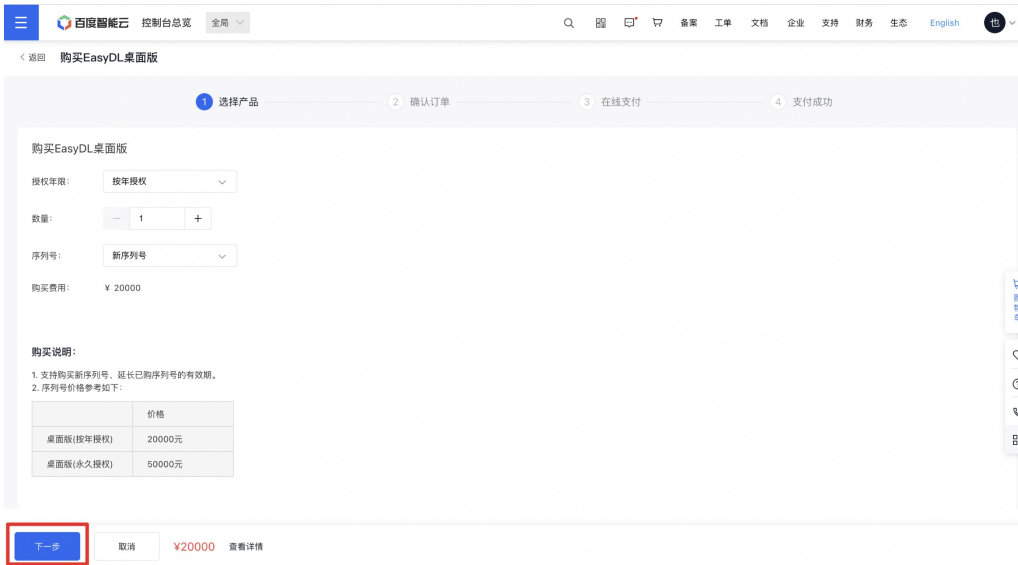
购买入口3



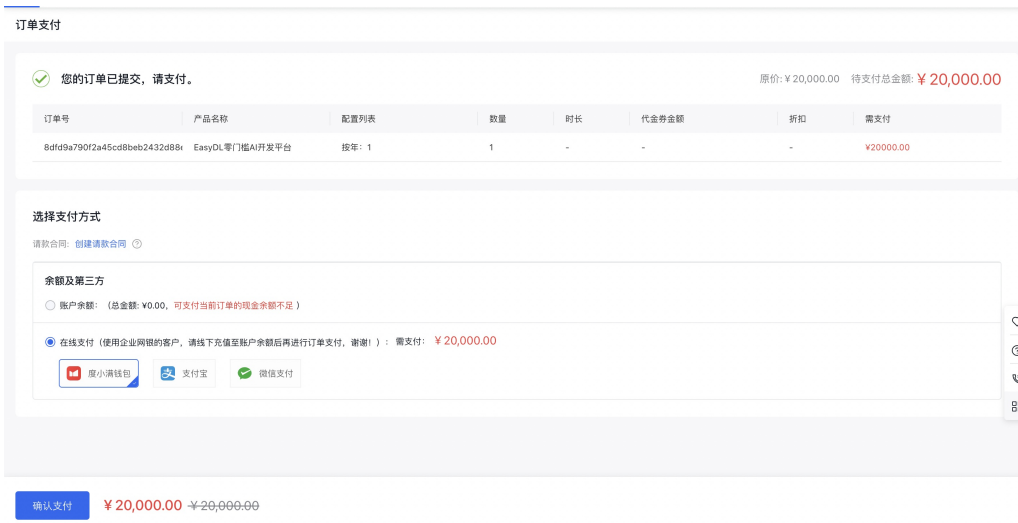
点击高级版购买入口跳转到云端登录个人账户



根据需求选购并点击下一步继续购买操作

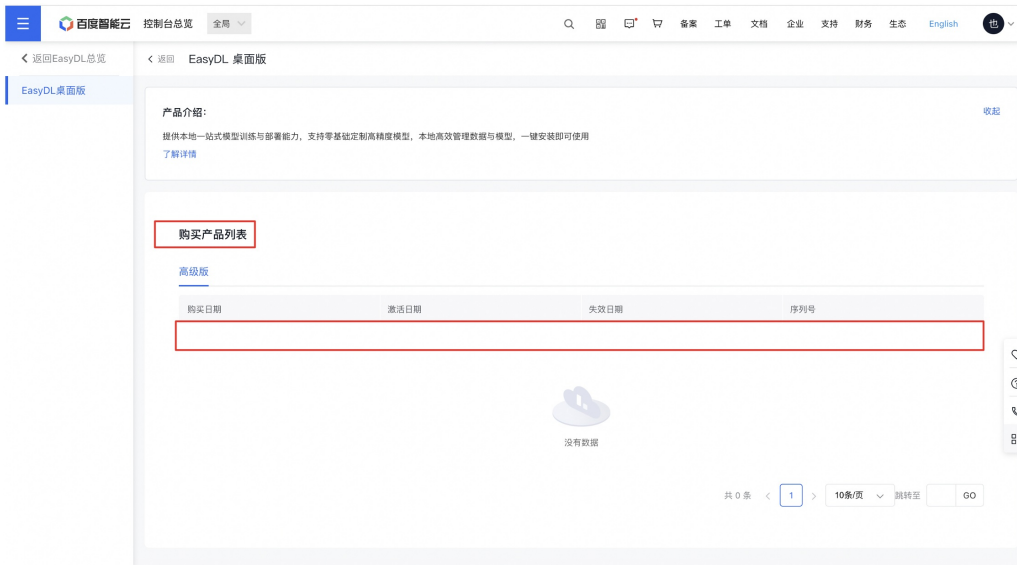


订单支付

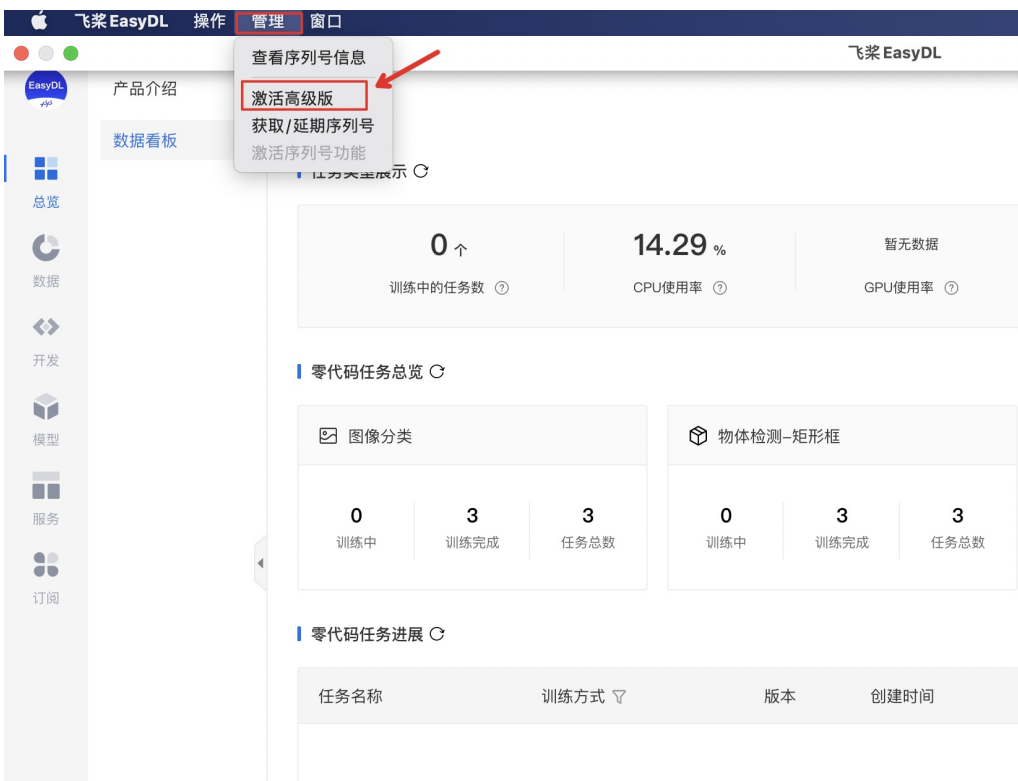


支付成功后会在购买记录里显示已购买产品信息和序列号信息





返回EasyDL桌面版客户端输入序列号完成高级版激活





# 飞桨EasyDL

零门槛AI开发平台

请输入16位序列号

完成绑定

1. 绑定时请保证您的设备处于联网状态
2. 该序列号将作为唯一标识用于您后续在客户端进行产品升级/延期，请勿共享
3. 如有序列号可直接输入完成绑定使用，如无序列号请先 [购买高级版](#)

欢迎订阅飞桨EasyDL高级版 收起

免费试用：有效期剩余 14654天 （如需延期，请点击右侧购买高级版） 购买高级版

<p><b>高级版激活完成</b></p>  <p><b>灵活的高阶调参训练方式</b></p> <p>支持手动、自动参数设置，系统自动设置参数训练完成后可视化展示</p>	 <p><b>高效的数据增强策略</b></p> <p>有效提高数据利用率，助力训练出精度更高、泛化能力更强的模型</p>	 <p><b>便捷的模型部署服务</b></p> <p>模型一键发布为离线SDK，适配广，最快5分钟即可完成业务集成</p>
---	--	---

## AI开发基础知识

### AI概念及基本原理

人工智能（Artificial Intelligence，英文缩写为AI。它是研究、开发用于模拟、延伸和扩展人的智能的理论、方法、技术及应用系统的一门新的技术科学。人工智能企图生产出一种新的能以人类智能相似的方式做出反应的智能机器，该领域的研究包括机器人、语言识别、图像识别、自然语言处理等。

在EasyDL平台背后主要使用了深度学习的技术，深度学习是机器学习(ML, Machine Learning)领域中一个新的研究方向。通过学习样本数据的内在规律和表示层次，最终目标是让机器能够像人一样具有分析学习能力，能够识别文字、图像和声音等数据。

## AI模型开发的基本流程介绍



### \*\*step1：分析业务需求\*\*

在正式启动训练模型之前，需要有效分析和拆解业务需求，明确模型类型如何选择。这里我们可以举一些实际业务场景进行分析。

**举例：**原始业务需求—某企业希望为某个高端小区物业做一套智能监控系统，希望对多种现象智能监控并及时预警，包括保安是否在岗、小区内是否有异常噪音、小区内各个区域的垃圾桶是否已满等多个业务功能。针对这个原始业务需求，我们可以分析出不同的监控对象所在的位置不同、监控的数据类型不同（有的针对图片进行识别、有的针对声音进行判断），需要多个模型综合应用。**监控保安是否在岗**——通过图像分类模型进行判断 **监控小区内是否有异常噪音**——定时收集声音片段通过声音分类模型进行判断 **监控小区内各个区域垃圾桶是否已满**——由于监控区域采集的画面可能会存在多个垃圾桶，此处需要通过物体检测模型进行判断。

### \*\*step2：采集/收集数据\*\*

在通过上述第一步分析出基本的模型类型，需要进行相应的数据收集工作。数据的主要原则为尽可能采集真实业务场景一致的数据，并覆盖可能有的各种情况

### \*\*step3：标注数据\*\*

采集数据后，可以通过EasyDL在线标注工具或线下其他标注工具对已有的数据进行标注。如上述保安是否在岗的图像分类模型，需要将监控视频抽帧后的图片按照【在岗】及【未在岗】两类进行整理；小区内各个区域垃圾桶是否已满，需要将监控视频抽帧后的图片标注其中每个垃圾桶的【空】【满】两种状态进行标注。

### \*\*step4：训练模型\*\*

训练模型阶段可以将已有标注好的数据基于已经确定的初步模型类型，选择算法进行训练。通过使用EasyDL平台，可以可视化在线操作训练任务的启停、训练任务的配置。可以大幅减少线下搭建训练环境、自主编写算法代码的相关成本。

### \*\*step5：评估模型效果\*\*

训练后的模型在正式集成之前，需要评估模型效果是否可用。在这个环节上EasyDL提供了详细的模型评估报告，以及在线可视化上传数据测试模型效果的功能。

### \*\*step6：部署模型\*\*

当确认模型效果可用后，可以将模型部署至生产环境中。传统的方式需要将训练出的模型文件加入工程化相关处理，通过使用EasyDL，可以便捷地将模型部署在公有云服务器或本地设备上，通过API或SDK集成应用，或直接购买软硬一体产品，有效应对各种业务场景所需，提供效果与性能兼具的服务。

## 快速开始

### 用零代码开发实现图像分类

#### 示例说明

图像分类模型主要用于识别一张图中是否是某类物体/状态/场景，适合图片中主体或状态单一的场景。本文以害虫识别模型在macOS客户端中的使用为例演示图像分类模型训练全过程。

#### 实现步骤

只需八步即可完成自定义AI模型的训练及发布的全过程。

##### Step1：提前准备训练数据

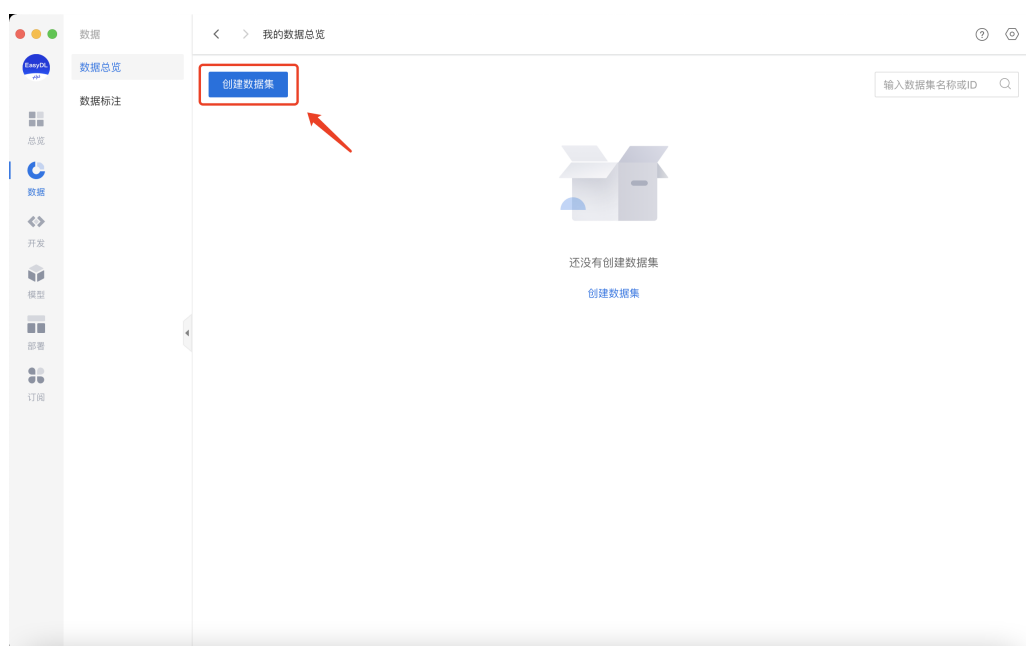
图像分类需要提供包含不同类别的图片并标注，完成后即可训练对应图像分类模型，自动识别图中是否包含某类物体/状态/场景，下面我们来看看这次训练所需的昆虫图片示例：

图片数量越多理论上训练效果越好，图像分类的图片数量建议每个类别不低于20张图片



### Step2 : 创建数据集

在数据总览界面点击【创建数据集】



在数据集创建界面输入数据集名称、选择标注类型后点击【完成】



### Step3 : 导入数据

数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面

数据ID	数据量	标注类型	标注状态	操作
20	0	图像分类	0% (0/0)	<input type="button" value="导入"/> <input type="button" value="导出"/>

数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息图片的导入为示例，其余各类型导入方式可参考 [导入图像数据选择数据标注状态与文件路径](#)

### 1 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

- i** 提示：1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用  
 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例  
[图像分类训练数据集\(coco格式\)](#)

上传图片时，请注意格式要求！

### 3、导入格式要求

#### 图片格式要求

目前支持图片类型为jpg, png, bmp, jpeg，图片大小限制在14M以内。

图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px。

#### 导入路径要求

无标注信息：导入请确保将全部图片保存至同一层文件目录。

有标注信息：导入请确保将全部图片与对应标注信息保存至同一层文件目录。该目录下子文件目录及非相关内容（例如压缩包）不导入。

完成后，点击【确认并返回】跳转至数据总览页

### 1 导入数据

数据标注状态  无标注信息  有标注信息

导入方式  上次导入路径：/Users/heyun02/Desktop/数据...

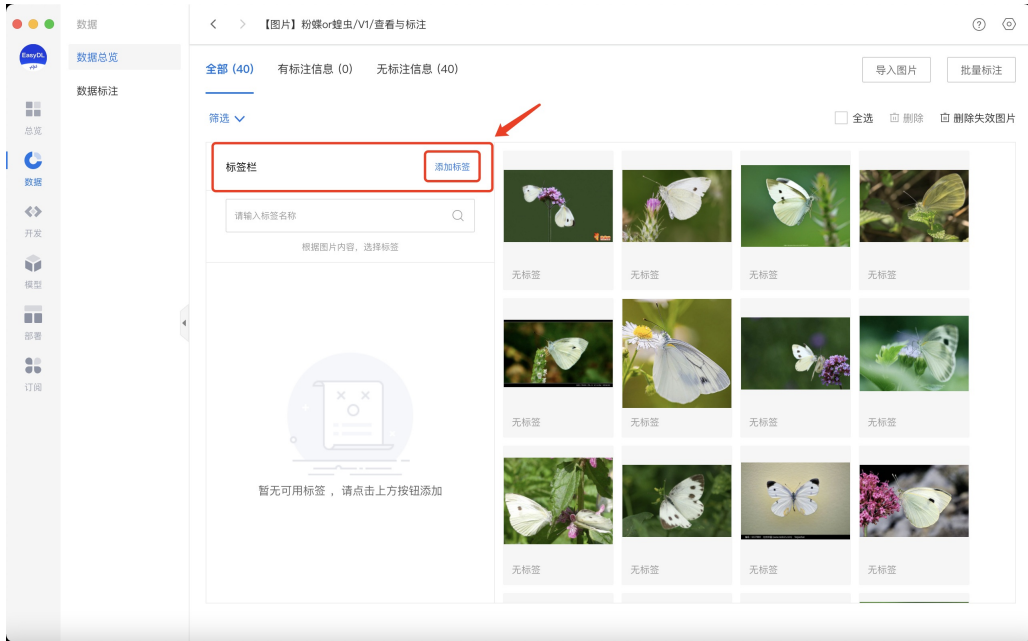
- i** 提示：1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用  
 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例  
[图像分类训练数据集\(coco格式\)](#)

### Step4：标注数据

在数据总览页找到需要标注的数据集，点击【查看与标注】，跳转至标注页面



在左侧标签栏下，点击【添加标签】创建数据集标签

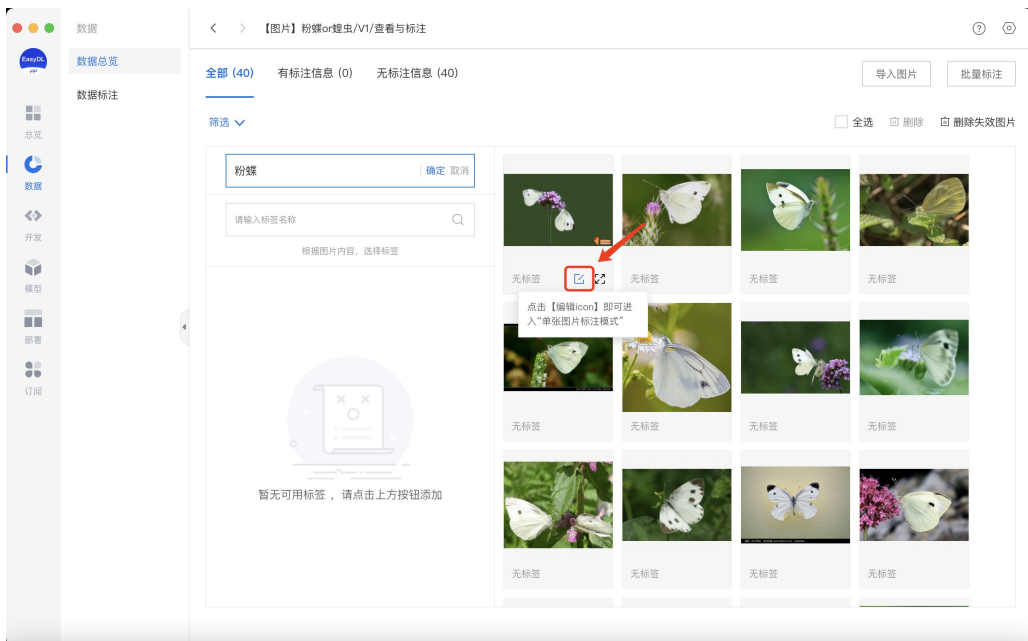


分别输入粉蝶、蝗虫并点击【确认】添加数据标签

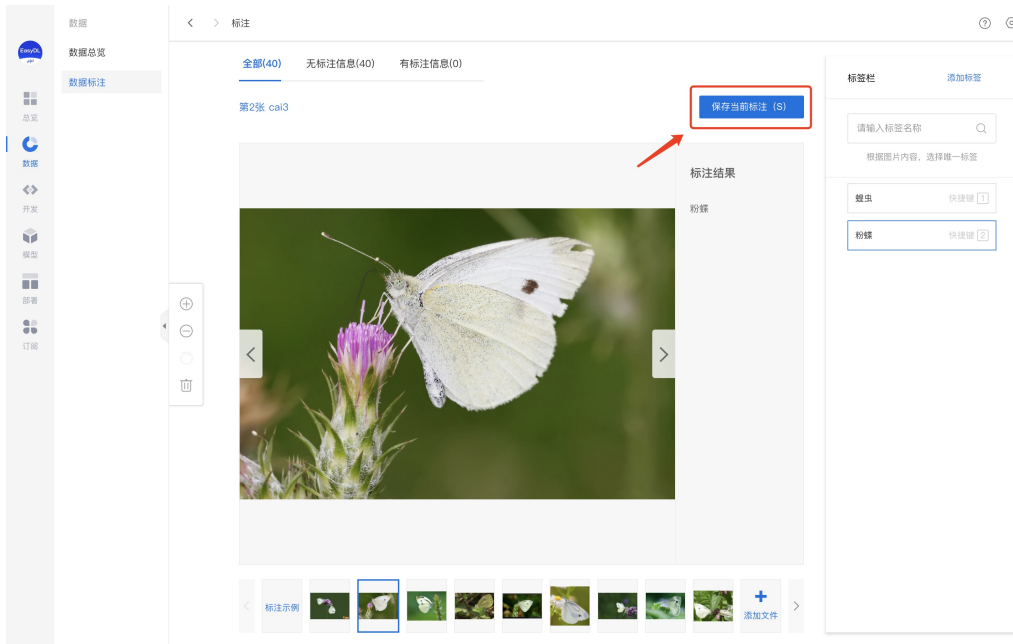
筛选



点击图片右下角红框内图标进入到数据标注界面

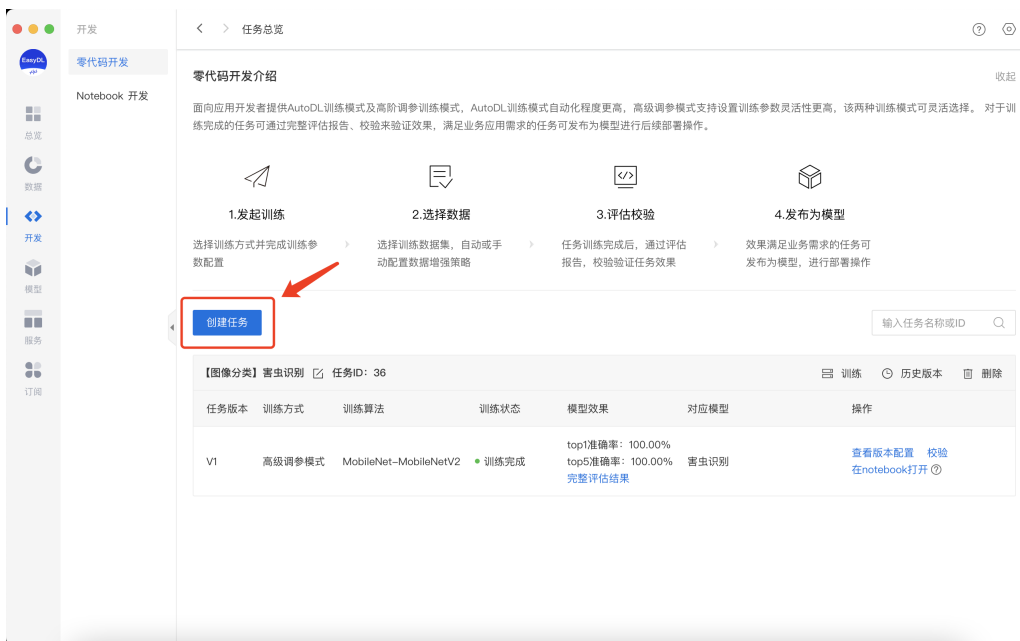


在当前图片下选择右侧标签栏内的某一类别，代表为图片打上相应的标签，点击【保存当前图片】或直接点击下一张图标，在保存标注结果后自动跳转至下一张。标注完所有图片后，该数据集便可用于后续训练任务



**Step5：创建训练任务**

在任务总览界面点击【创建任务】



在训练任务创建界面输入任务名称、选择任务类型后点击【创建任务】

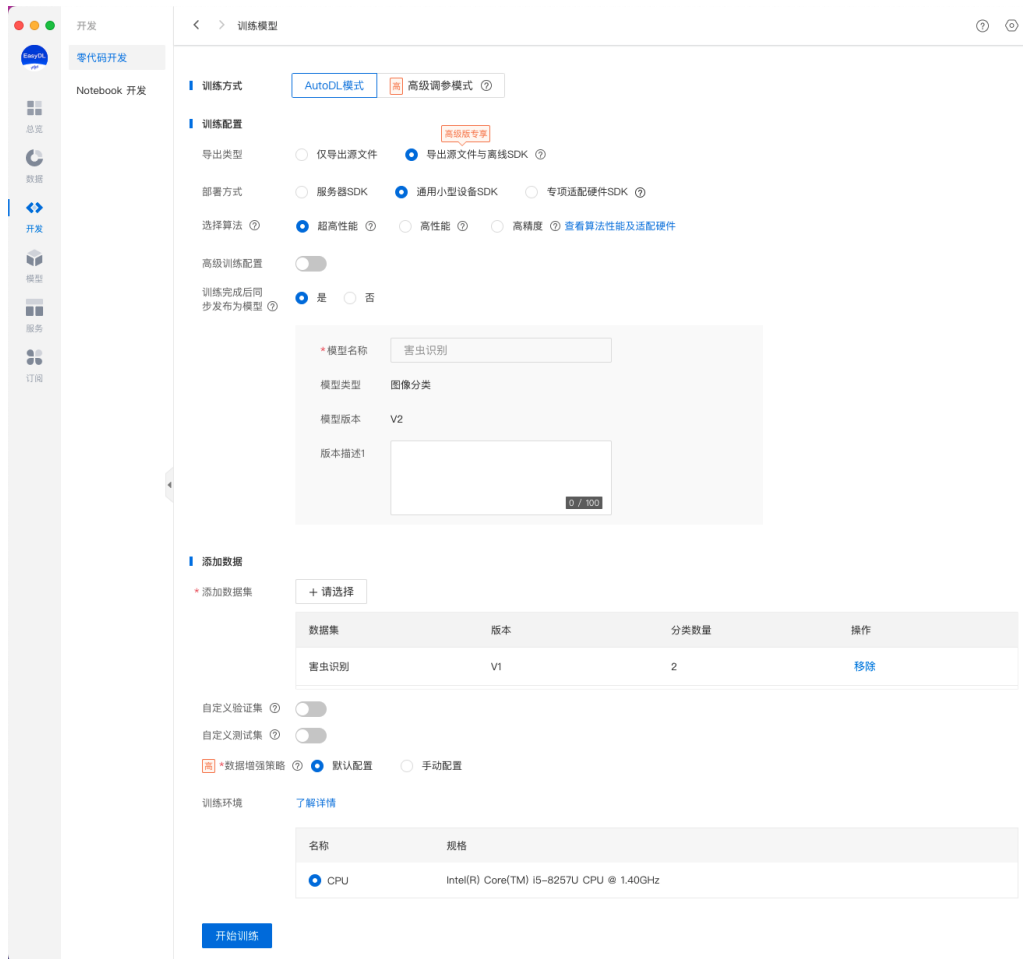




任务创建完成后，点击【训练】，进入训练配置阶段



根据需求选择各项训练配置后，添加训练数据集，点击【开始训练】



在任务总览页任务列表下，可以看到处于训练状态的训练任务，将鼠标放置感叹号图标处，即可查看训练进度



### Step6：模型校验

训练完成后，可在任务列表下，点击【校验】



< > 任务总览 ? @

---

零代码开发介绍 展开

面向应用开发者提供AutoDL训练模式及高阶调参训练模式，AutoDL训练模式自动化程度更高，高阶调参训练模式训练参数设置灵活。

[创建任务](#) 输入任务名称或ID 🔍

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	仅导出源文件-自行部署-高性能	训练完成	top1准确率: 100.00% top5准确率: 100.00% <a href="#">完整评估结果</a>		<a href="#">查看版本配置</a> <a href="#">删除</a> <a href="#">发布为模型</a> <a href="#">校验</a>

### 点击【添加图片】，进行模型校验

< > 校验模型 ? @

选择任务 害虫识别 选择版本 V1 训练算法 仅导出源文件-自行部署-高性能

当前模型准确率 100.00% [评估报告](#) 调整阈值  0.03

识别结果: 粉蝶

[点击图片](#)

预测分类 置信度 > 3.00% ▾

粉蝶	99.98%
----	--------

### Step7：发布为模型

确认模型效果满意后，可在任务列表下，点击【发布为模型】

< > 任务总览 ? @

---

零代码开发介绍 展开

面向应用开发者提供AutoDL训练模式及高阶调参训练模式，AutoDL训练模式自动化程度更高，高阶调参训练模式训练参数设置灵活。

[创建任务](#) 输入任务名称或ID 🔍

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	仅导出源文件-自行部署-高性能	训练完成	top1准确率: 100.00% top5准确率: 100.00% <a href="#">完整评估结果</a>		<a href="#">查看版本配置</a> <a href="#">删除</a> <a href="#">发布为模型</a> <a href="#">校验</a>

输入模型名称、版本描述后点击【确认】正式发布未模型

## 发布至模型仓库

✕

\* 模型名称

害虫识别

模型类型 图像分类

模型版本 V1

版本描述

0 / 100

取消

确定

## Step8：导出模型文件或部署SDK

根据您在训练时选择的部署方式，可在模型总览版本列表中，点击【导出模型文件】或【部署】

版本	对应任务	训练方式	描述	导入时间	操作
V2	10-V2	AutoDL模式		2021-11-22 08:31:21	导出模型文件 部署
V1	9-V1	AutoDL模式		2021-11-22 07:42:03	导出模型文件

点击部署进入服务发布界面，选择模型部署设备的芯片型号，点击【发布】，跳转至服务列表页

&lt; &gt; 模型部署

选择模型

模型版本

训练方式 零代码开发-AutoDL模式

训练算法 服务器-高性能

选择部署环境

选择系统和芯片  本地发布

Linux

Windows

发布

在服务列表中选择需要导出的模型SDK，点击【导出SDK】，选择存储位置后即可完成模型SDK的导出

部署 服务列表

飞桨EasyDL推出智能边缘控制台，助力本地部署场景用户高效管理端侧预测设备与服务，极速完成本地数据与模型串联，提高模型落地效率。了解详情

服务器 通用小型设备 专项适配硬件

模型名称	发布版本	应用平台	发布状态	发布方式	发布时间	操作
害虫识别	5-V2	通用X86 CPU-Linux	已发布	本地部署	2021-11-22 20:51	导出SDK
猫狗分类	1-V1	英伟达GPU-Windows	已发布	本地部署	2021-09-27 17:34	导出SDK

## 用零代码开发实现物体检测

### 示例说明

物体检测模型主要用于检测图中每个物体的位置、类型。适合图中有多个主体要识别、或要识别主体位置及数量的场景。本文以螺丝螺母识别模型在macOS客户端中的使用为示例演示物体检测模型训练全过程。

### 实现步骤

只需八步即可完成自定义AI模型的训练及发布的全过程。

#### Step1：提前准备训练数据

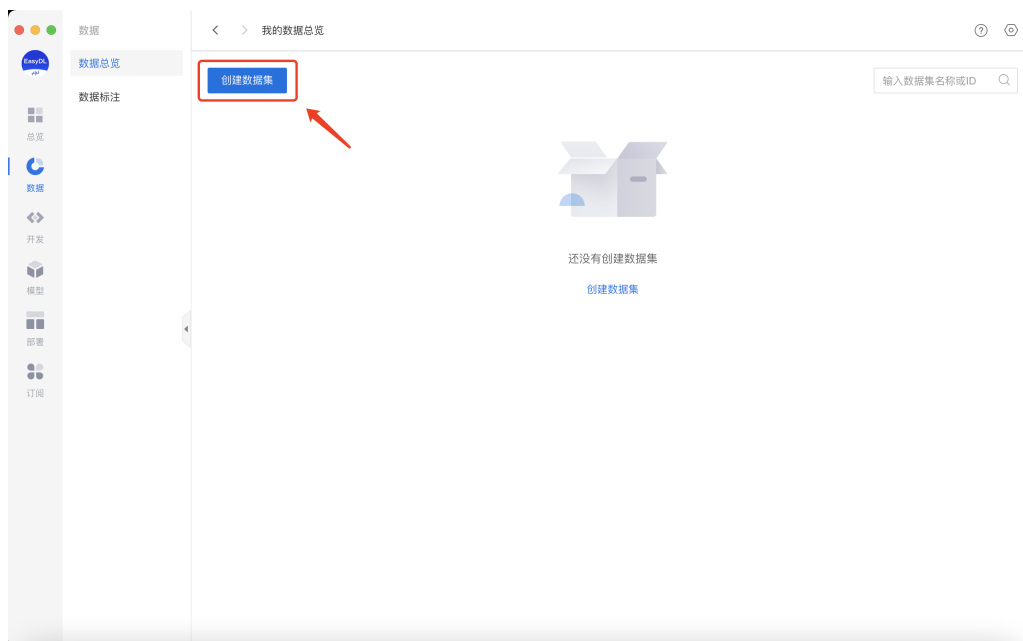
物体检测需要提供包含目标物体的图片并标注物体即可训练物体检测模型，自动识别图中所有目标物体的位置、名称，下面我们来看看这次需要计数的包含螺丝螺母的图片示例：

图片数量越多理论上训练效果越好，物体检测的图片数量建议每个类别不低于20张图片 注意图片需要为业务生产的真实环境所采集的图片，与真实场景越贴近，训练模型效果越佳



### Step2 : 创建数据集

在数据总览界面点击【创建数据集】



在数据集创建界面输入数据集名称、选择标注类型后点击【完成】



### Step3 : 导入数据

数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面

数据集ID	数据量	标注类型	标注状态	操作
22	0	物体检测	0% (0/0)	<a href="#">导入</a> <a href="#">导出</a>

数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息图片的导入为示例，其余各类型导入方式可参考 [导入图像数据选择数据标注状态与文件路劲](#)

### 1 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

- 提示：** 1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用
- 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例 [图像分类训练数据集\(coco格式\)](#)

上传图片时，请注意格式要求！

### 3、导入格式要求

#### 图片格式要求

目前支持图片类型为jpg, png, bmp, jpeg，图片大小限制在14M以内。

图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px。

#### 导入路径要求

无标注信息：导入请确保将全部图片保存至同一层文件目录。

有标注信息：导入请确保将全部图片与对应标注信息保存至同一层文件目录。该目录下子文件目录及非相关内容（例如压缩包）不导入。

完成后，点击【确认并返回】跳转至数据总览页

### 1 导入数据

数据标注状态  无标注信息  有标注信息

导入方式  上次导入路径：/Users/heyun02/Desktop/数据...

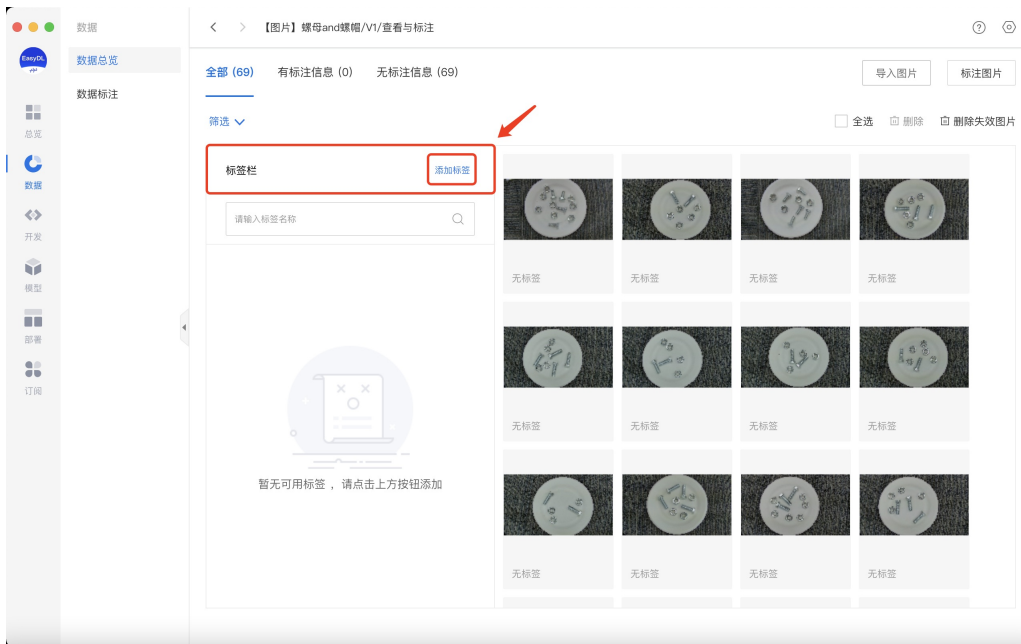
- 提示：** 1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用
- 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例 [图像分类训练数据集\(coco格式\)](#)

### Step4：标注数据

在数据总览页找到需要标注的数据集，点击【查看与标注】，跳转至标注页面

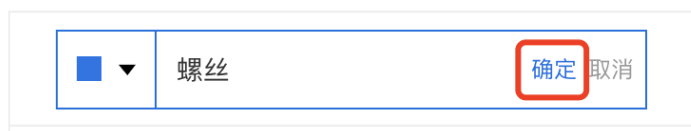
数据集ID	数据量	标注类型	标注状态	操作
22	69	物体检测	0% (0/69)	<a href="#">查看与标注</a> <a href="#">导入</a> <a href="#">导出</a>

在左侧标签栏下，点击【添加标签】创建数据集标签

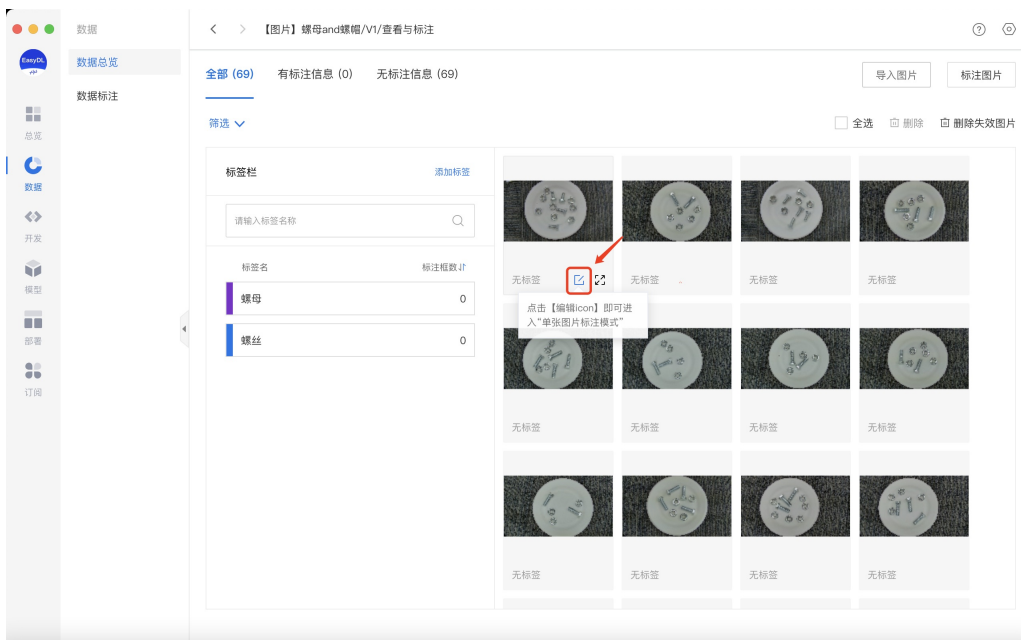


分别输入螺丝、螺母并点击【确认】添加数据标签

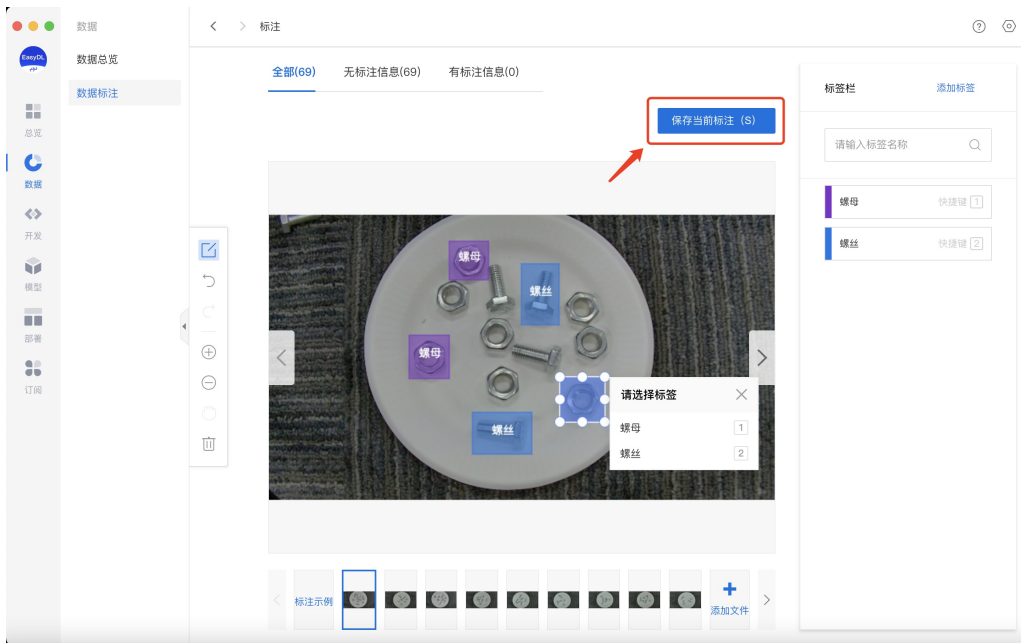
筛选 ▾



点击图片右下角红框内图标进入到数据标注界面

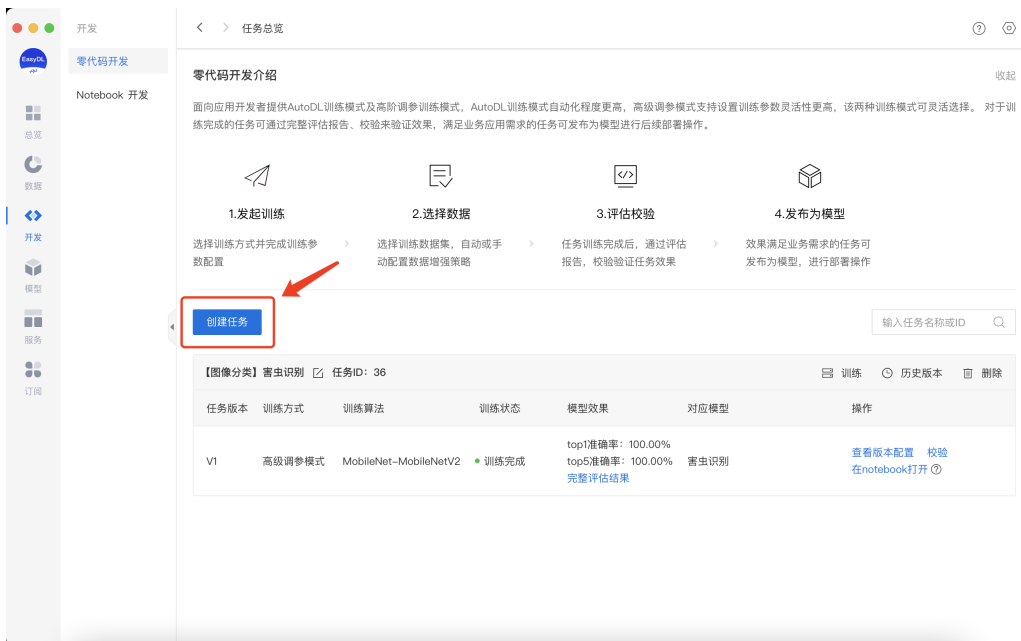


在当前图片下选择右侧标签栏内的某一类别，代表为图片打上相应的标签，点击【保存当前图片】或直接点击下一张图片，在保存标注结果后自动跳转至下一张。标注完所有图片后，该数据集便可用于后续训练任务



Step5 : 创建训练任务

在任务总览界面点击【创建任务】



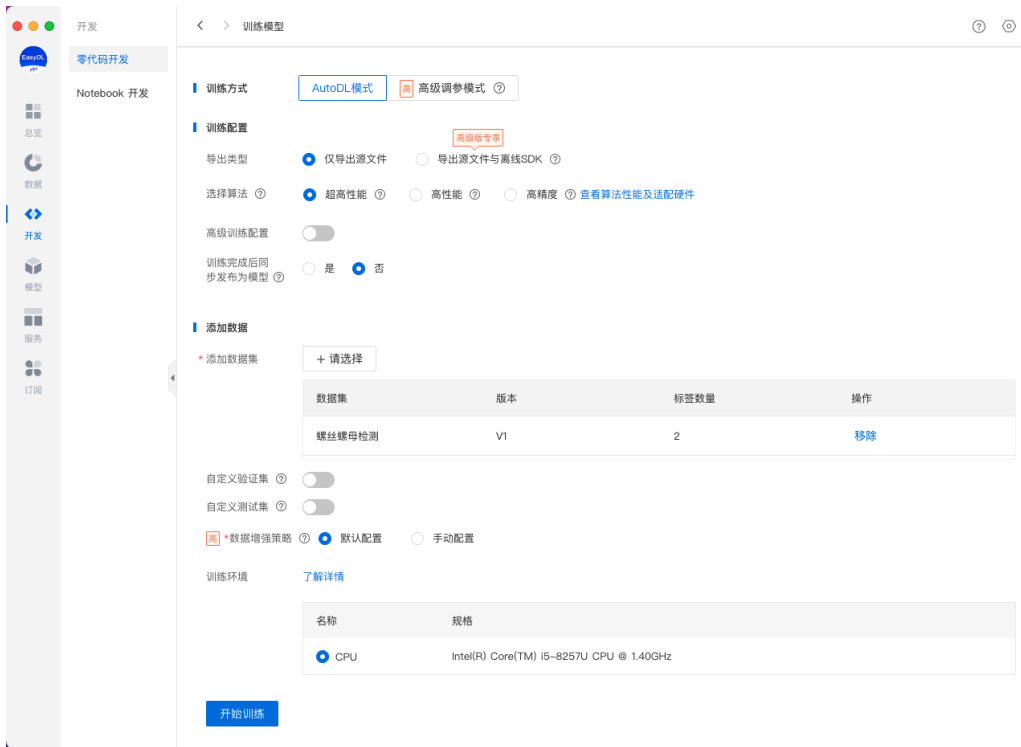
在训练任务创建界面输入任务名称、选择任务类型后点击【创建任务】



任务创建完成后, 点击【训练】, 进入训练配置阶段



根据需求选择各项训练配置后，添加训练数据集，点击【开始训练】



在任务总览页任务列表下，可以看到处于训练状态的训练任务，将鼠标放置感叹号图标处，即可查看训练进度



### Step6：模型校验

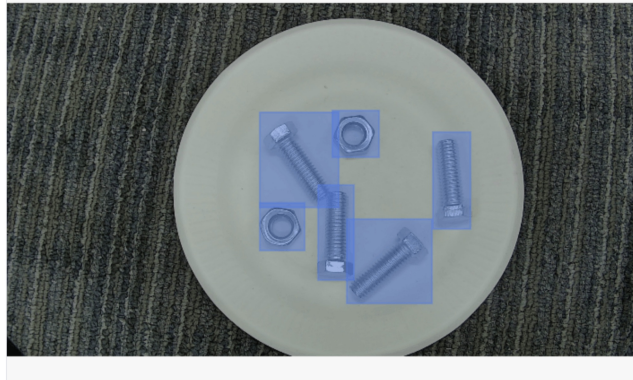
训练完成后，可在任务列表下，点击【校验】



点击【添加图片】，进行模型校验



当前模型mAP平均精度 100.00% [评估报告](#)



[点击添加图片](#)

识别结果 [如何优化效果?](#)

调整阈值  0.3

预测标签	置信度 > 30.00% ▾
7. 螺丝	99.96%
6. 螺丝	99.90%
5. 螺丝	99.84%
4. 螺丝	99.77%
3. 螺母	99.66%
2. 螺母	99.59%

### Step7 : 发布为模型

确认模型效果满意后，可在任务列表下，点击【发布为模型】

< > 任务总览 ? @

零代码开发介绍 展开

面向应用开发者提供AutoDL训练模式及高阶调参训练模式，AutoDL训练模式自动化程度更高，高阶调参训练模式训练参数设置灵活。

[创建任务](#)

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	仅导出源文件-自行部署-高性能	训练完成	top1准确率: 100.00% top5准确率: 100.00% <a href="#">完整评估结果</a>		<a href="#">查看版本配置</a> <a href="#">删除</a> <a href="#">发布为模型</a> <a href="#">校验</a>

输入模型名称、版本描述后点击【确认】正式发布未模型

### 发布至模型仓库

\* 模型名称

模型类型

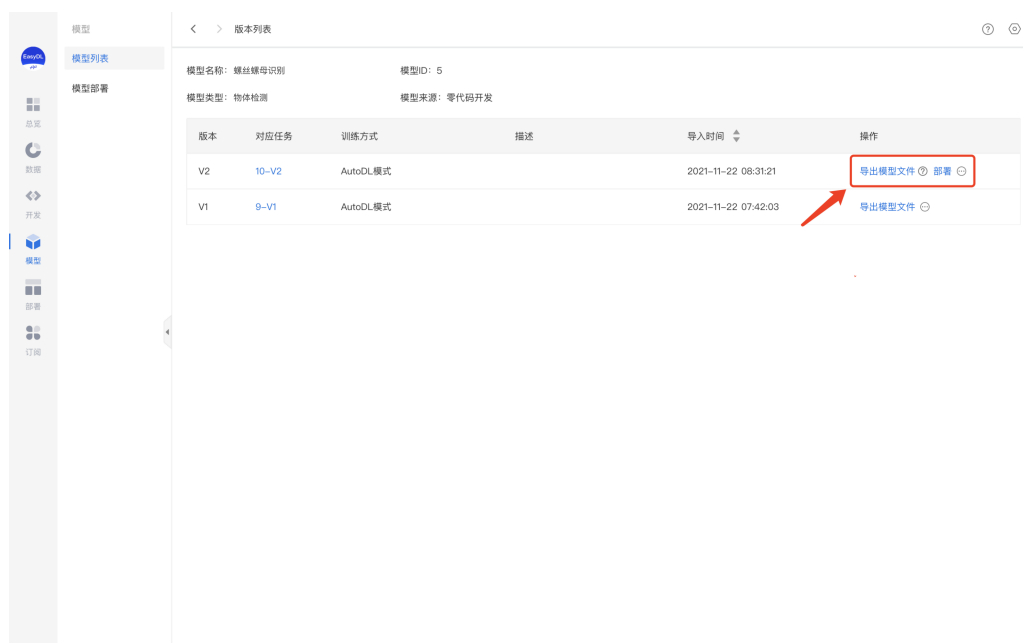
模型版本

版本描述

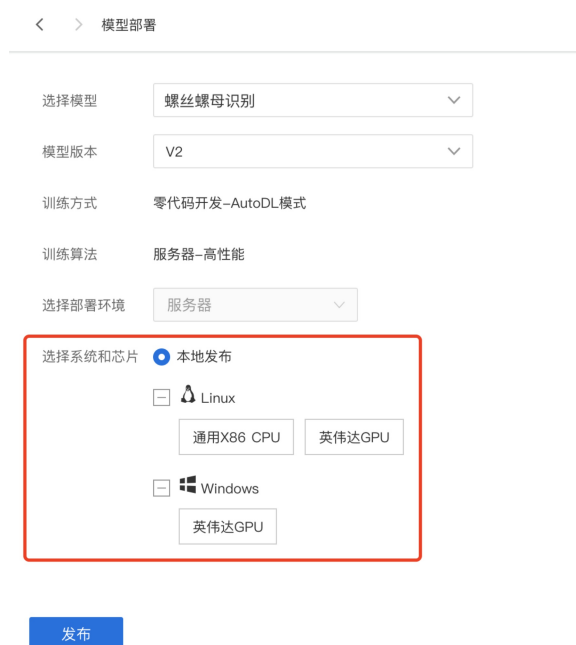
0 / 100

### Step8 : 导出模型文件或部署SDK

根据您在训练时选择的部署方式，可在模型总览版本列表中，点击【导出模型文件】或【部署】



点击部署进入服务发布界面，选择模型部署设备的芯片型号，点击【发布】，跳转至服务列表页



在服务列表中选择需要导出的模型SDK，点击【导出SDK】，选择存储位置后即可完成模型SDK的导出



## 用零代码开发实现实例分割

### 示例说明

对比物体检测，实例分割支持用多边形标注训练数据，且模型可像素级识别目标。适合图中有多个主体、需识别其位置或轮廓的场景。本文以工件分割模型在macOS客户端中的使用为例演示实例分割模型训练全过程。

### 实现步骤

只需八步即可完成自定义AI模型的训练及发布的全过程。

#### Step1：提前准备训练数据

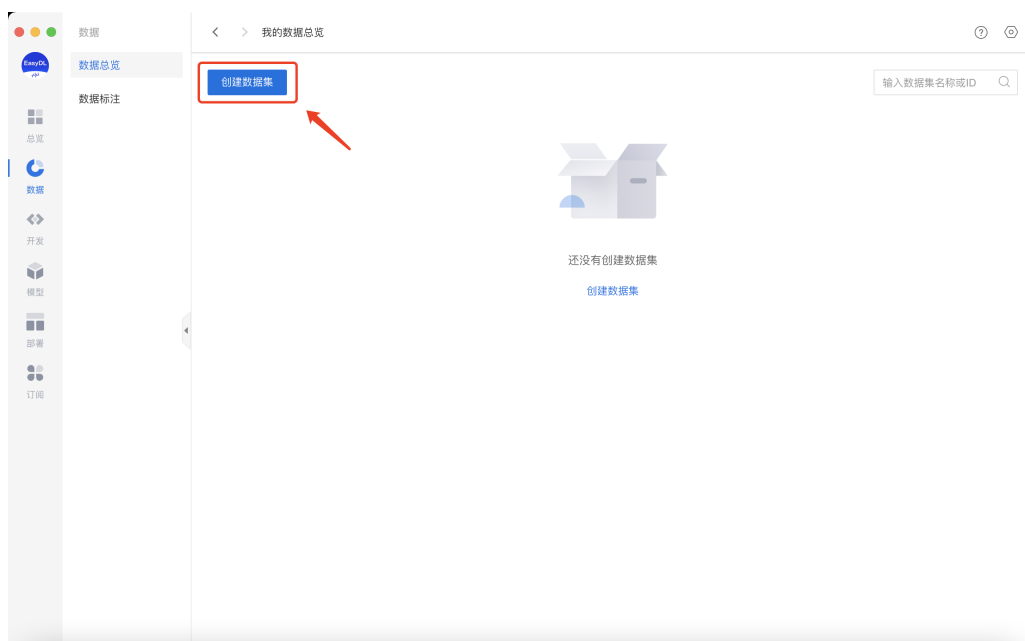
实例分割需要提供包含目标物体的图片并标注物体即可训练分割模型，自动识别图中所有目标物体的位置、轮廓、名称，下面我们来看看这次需要分割的包含螺丝螺母的图片示例：

图片数量越多理论上训练效果越好，物体检测的图片数量建议每个类别不低于20张图片 注意图片需要为业务生产的真实环境所采集的图片，与真实场景越贴近，训练模型效果越佳

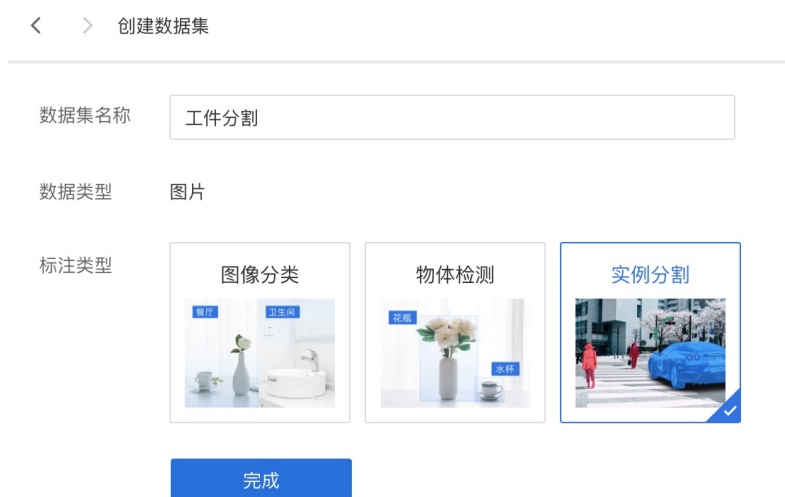


#### Step2：创建数据集

在数据总览界面点击【创建数据集】



在数据集创建界面输入数据集名称、选择标注类型后点击【完成】



### Step3：导入数据

数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面

数据集ID	数据量	标注类型	标注状态	操作
24	0	实例分割	0% (0/0)	<a href="#">导入</a> <a href="#">导出</a>

数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息图片的导入为示例，其余各类型导入方式可参考 [导入图像数据选择数据标注状态与文件路径](#)

#### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

- 提示：** 1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用  
 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例  
[图像分类训练数据集\(coco格式\)](#)

上传图片时，请注意格式要求！

### 3、导入格式要求

#### 图片格式要求

目前支持图片类型为jpg, png, bmp, jpeg, 图片大小限制在14M以内。

图片长宽比在3:1以内, 其中最长边小于4096px, 最短边大于30px。

#### 导入路径要求

无标注信息: 导入请确保将全部图片保存至同一层文件目录。

有标注信息: 导入请确保将全部图片与对应标注信息保存至同一层文件目录。该目录下子文件目录及非相关内容(例如压缩包)不导入。

完成后, 点击【确认并返回】跳转至数据总览页

## 1 导入数据

数据标注状态



无标注信息



有标注信息

导入方式

重新选择

上次导入路径: /Users/heyun02/Desktop/数据...

- 提示:** 1.导入后请避免改动本地该数据, 以免影响数据标注、模型训练功能正常使用  
2.每次导入仅支持选择唯一目录, 如您想快速体验一站式功能, 可联网下载已标训练数据样例  
[图像分类训练数据集\(coco格式\)](#)

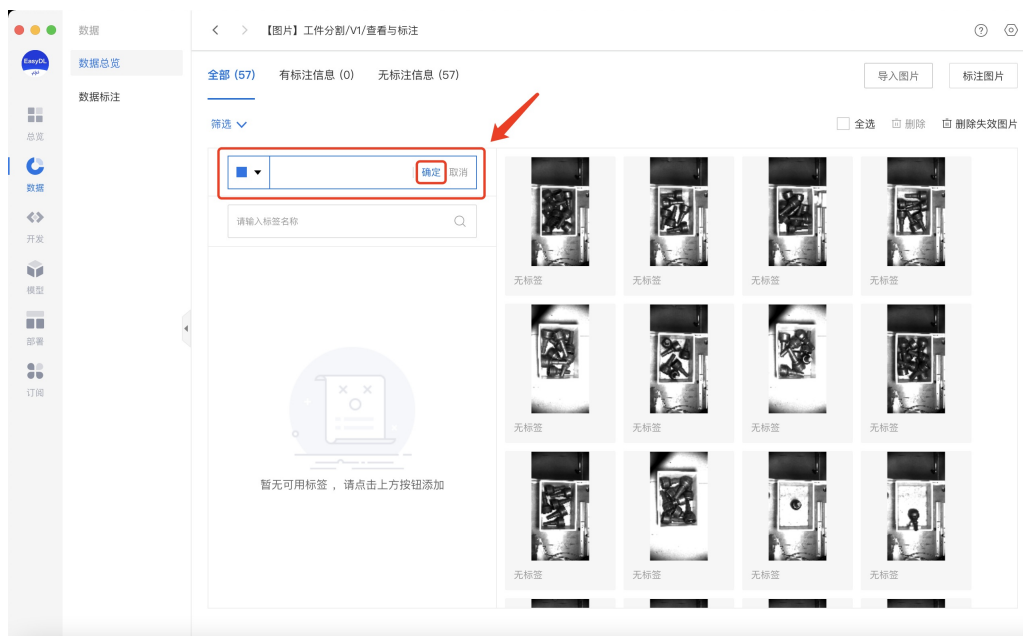
确认并返回

## Step4 : 标注数据

在数据总览页找到需要标注的数据集, 点击【查看与标注】, 跳转至标注页面

数据集ID	数据量	标注类型	标注状态	操作
24	57	实例分割	0% (0/57)	<a href="#">查看与标注</a> <a href="#">导入</a> <a href="#">导出</a>

在左侧标签栏下, 点击【添加标签】创建数据集标签

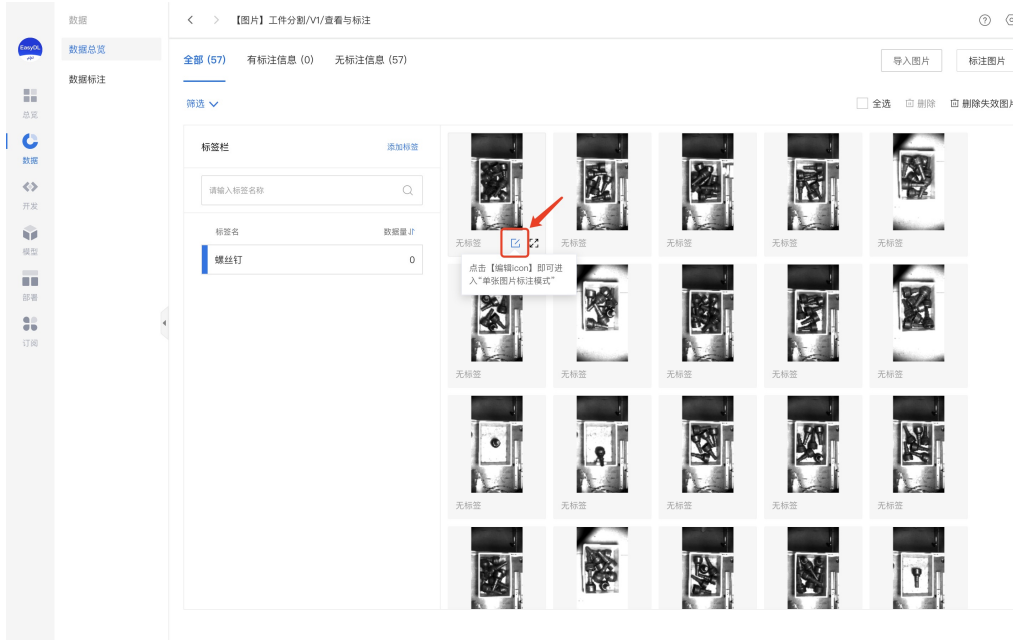


输入螺丝钉并点击【确认】添加数据标签

筛选



点击图片右下角红框内图标进入到数据标注界面

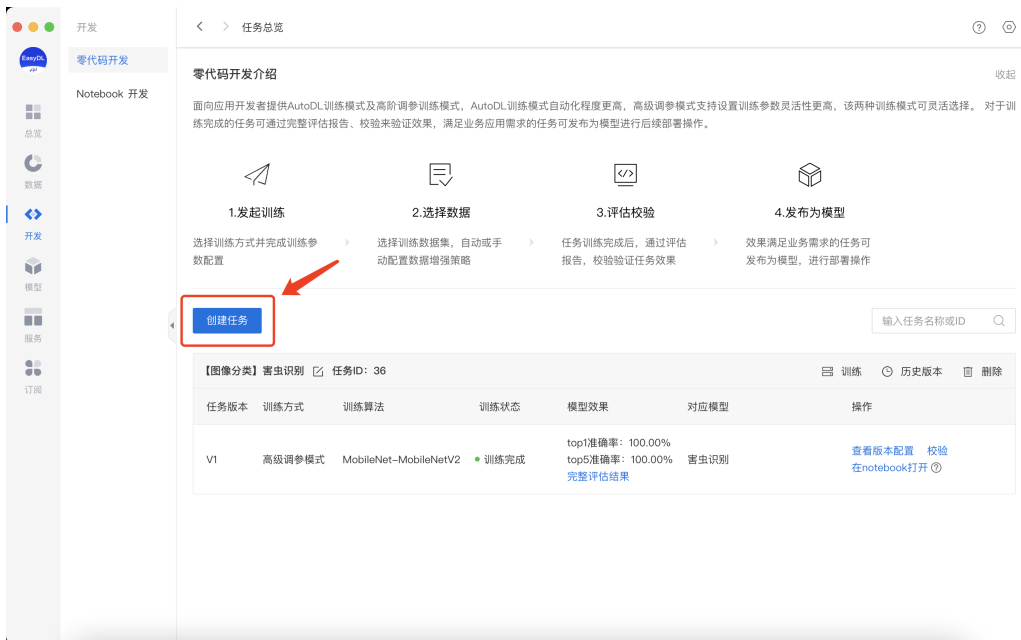


在当前图片下选择右侧标签栏内的某一类别，代表为图片打上相应的标签，点击【保存当前图片】或直接点击下一张图片图标，在保存标注结果后自动跳转至下一张。标注完所有图片后，该数据集便可用于后续训练任务



### Step5：创建训练任务

在任务总览界面点击【创建任务】



在训练任务创建界面输入任务名称、选择任务类型后点击【创建任务】



任务创建完成后，点击【训练】，进入训练配置阶段



根据需求选择各项训练配置后，添加训练数据集，点击【开始训练】



在任务总览页任务列表下，可以看到处于训练状态的训练任务，将鼠标放置感叹号图标处，即可查看训练进度

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	通用小型设备-默认	训练中	-		查看版本配置 删除 终止训练

训练进度: 1%  
剩余时间: 1小时10分钟

### Step6：模型校验

训练完成后，可在任务列表下，点击【校验】

任务总览

零代码开发介绍

创建任务

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	仅导出源文件-自行部署-高性能	训练完成	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果		查看版本配置 发布为模型 校验 删除

点击【添加图片】，进行模型校验

当前模型mAP平均精度 86.94% 评估报告



识别结果 如何优化效果?

调整阈值 0.9

预测标签	置信度 > 90.00%
7. 螺丝	99.99%
6. 螺丝	99.99%
5. 螺丝	99.99%
4. 螺丝	99.98%
3. 螺丝	99.98%
2. 螺丝	99.96%

点击添加图片

### Step7：发布为模型

确认模型效果满意后，可在任务列表下，点击【发布为模型】

任务总览

零代码开发介绍

创建任务

任务版本	训练方式	训练算法	训练状态	模型效果	对应模型	操作
V1	AutoDL模式	仅导出源文件-自行部署-高性能	训练完成	top1准确率: 100.00% top5准确率: 100.00% 完整评估结果		查看版本配置 发布为模型 校验 删除

输入模型名称、版本描述后点击【确认】正式发布未模型



发布至模型仓库

×

\*模型名称

模型类型 实例分割

模型版本 V1

版本描述

0 / 100

### Step8：导出模型文件或部署SDK

根据您在训练时选择的部署方式，可在模型总览版本列表中，点击【导出模型文件】或【部署】

版本	对应任务	训练方式	描述	导入时间	操作
V2	10-V2	AutoDL模式		2021-11-22 08:31:21	导出模型文件 部署
V1	9-V1	AutoDL模式		2021-11-22 07:42:03	导出模型文件

点击部署进入服务发布界面，选择模型部署设备的芯片型号，点击【发布】，跳转至服务列表页

< > 模型部署

选择模型

模型版本

训练方式 零代码开发-AutoDL模式

训练算法 服务器-高性能

选择部署环境

选择系统和芯片  本地发布

Linux

Windows

在服务列表中选择需要导出的模型SDK，点击【导出SDK】，选择存储位置即可完成模型SDK的导出

部署 服务列表

飞桨EasyDL推出智能边缘控制台，助力本地部署场景用户高效管理端侧预测设备与服务，极速完成本地数据与模型串联，提高模型落地效率。 [了解详情](#)

服务器 通用小型设备 专项适配硬件

模型名称	发布版本	应用平台	发布状态	发布方式	发布时间	操作
工件检测	5-V2	通用X86 CPU-Linux	已发布	本地部署	2021-11-22 20:51	<a href="#">导出SDK</a>
猫狗分类	1-V1	英伟达GPU-Windows	已发布	本地部署	2021-09-27 17:34	<a href="#">导出SDK</a>

## 用零代码开发实现语义分割

### 示例说明

对比实例分割，语义分割指将每个像素点归属为对象类的过程。适用于分割目标主体单一的场景，简单举例来说语义分割能够识别出图片中哪些像素是归属于“人”的标签，但无法区分“不同的人”。本文以在macOS客户端中的使用为例演示实例分割模型训练全过程。

### 实现步骤

只需八步即可完成自定义AI模型的训练及发布的全过程。

#### Step1：提前准备训练数据

语义分割需要提供包含目标物体的图片并标注物体即可训练分割模型，下面我们来看看语义分割的城市建筑及道路物体分割图片示例：

我的数据总览 > 【图片】语义分割test/V3/查看与标注

全部 (845) 有标注信息 (845) 无标注信息 (0) 导入图片 质检报告 标注图片

筛选 □ 本页全选 🗑 删除

**标签栏** 添加标签

请输入标签名称

标签名	数据量
window	828
vegetation	630
sky	845
road	703
pavement	614
door	412

building building building building

building building building building

点击【编辑icon】即可进入“单张图片标注模式”

### Step2 : 创建数据集

在数据总览界面点击【创建数据集】

在数据集创建界面输入数据集名称、选择标注类型后点击【完成】

### Step3 : 导入数据

数据集创建完成后可在【数据总览】查看已创建完成的数据集，点击【导入】跳转至数据导入界面

数据导入支持无标注信息、有标注信息两种数据标注状态的数据以及多种导入方式，以下为无标注信息图片的导入为示例，其余各类型导入方式可参考 导入图像数据选择数据标注状态与文件路径

上传图片时，请注意格式要求！

### 3、导入格式要求

#### 图片格式要求

目前支持图片类型为jpg, png, bmp, jpeg, 图片大小限制在14M以内。

图片长宽比在3:1以内，其中最长边小于4096px，最短边大于30px。

#### 导入路径要求

无标注信息：导入请确保将全部图片保存至同一层文件目录。

有标注信息：导入请确保将全部图片与对应标注信息保存至同一层文件目录。该目录下子文件目录及非相关内容（例如压缩包）不导入。

完成后，点击【确认并返回】跳转至数据总览页

#### Step4：标注数据

在数据总览页找到需要标注的数据集，点击【查看与标注】，跳转至标注页面

在左侧标签栏下，点击【添加标签】创建数据集标签

我的数据总览 > 【图片】语义分割test/V3/查看与标注

全部 (845) 有标注信息 (845) 无标注信息 (0)

筛选

The screenshot shows the annotation interface. On the left, there is a label selection dropdown menu with a blue square icon and a search input field. Below the search field is a table of existing labels:

标签名	数据量
window	828
vegetation	630
sky	845
road	703

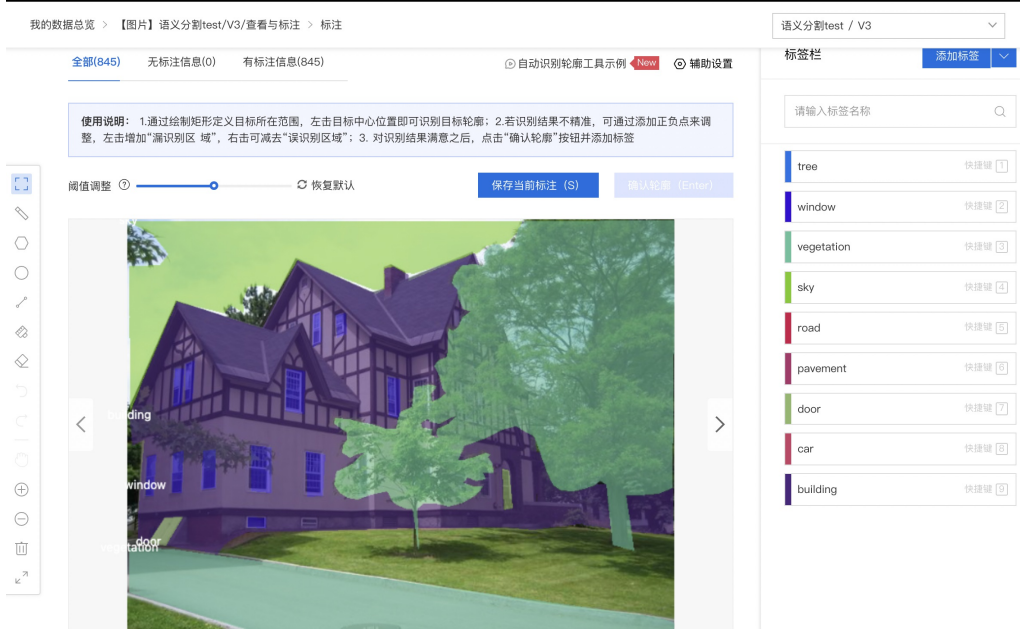
On the right, there is a grid of images with labels. The labels are 'building' for each image. A red box highlights the label selection dropdown menu.

输入并点击【确认】添加数据标签

筛选

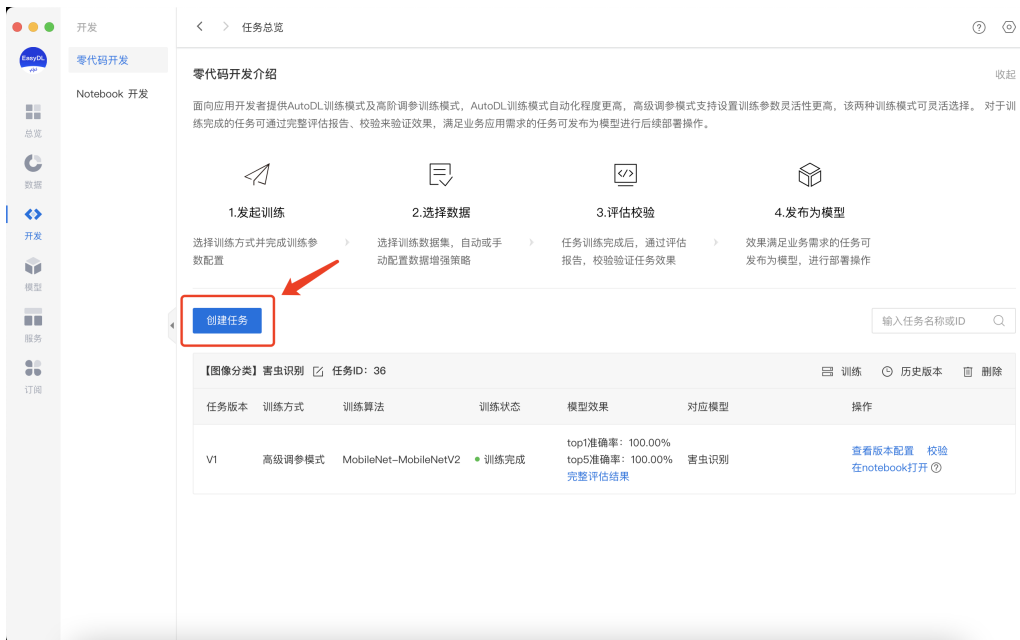
The screenshot shows the label selection dropdown menu with 'tree' selected. A red box highlights the '确定' (Confirm) button.

点击图片右下角编辑图标即可进入到数据标注界面，在当前图片下选择右侧标签栏内的某一类别，代表为图片打上相应的标签，点击【保存当前图片】或直接点击下一张图片图标，在保存标注结果后自动跳转至下一张。标注完所有图片后，该数据集便可用于后续训练任务



Step5：创建训练任务

在任务总览界面点击【创建任务】



在训练任务创建界面输入任务名称、选择任务类型后点击【创建任务】



任务创建完成后，点击

【训练】，进入训练配置阶段



根据需求选择各项训练配置后，添加训练数据集，点击【开始训练】



在任务总览页任务列表下，可以看到处于训练状态的训练任务，将鼠标放置感叹号图标处，即可查看训练进度

## Step6：模型校验

训练完成后，可在任务列表下，点击【校验】进入单图校验页面



点击【添加图片】上传本地图片即可完成模型校验

## Step7：发布为模型

确认模型效果满意后，可在任务列表下，点击【发布为模型】

输入模型名称、版本描述后点击【确认】正式发布为模型

## 发布至模型仓库

\* 模型名称

城市道路及建筑分割

模型类型 实例分割

模型版本 V1

版本描述

0 / 100

取消

确定

### Step8：导出模型文件或部署SDK

根据您在训练时选择的部署方式，可在模型总览版本列表中，点击【导出模型文件】或【部署】

点击部署进入服务发布界面，选择模型部署设备的芯片型号，点击【发布】，跳转至服务列表页

在服务列表中选择需要导出的模型SDK，点击【导出SDK】，选择存储位置后即可完成模型SDK的导出

部署

服务列表

服务总览

飞桨EasyDL推出智能边缘控制台，助力本地部署场景用户高效管理端侧预测设备与服务，极速完成本地数据与模型串联，提高模型落地效率。了解详情 ×

服务器 通用小型设备 专项适配硬件 ②

输入模型名称

模型名称	发布版本	应用平台	发布状态	发布方式	发布时间	操作
工件检测	5-V2	通用X86 CPU-Linux	● 已发布	本地部署	2021-11-22 20:51	导出SDK
猫狗分类	1-V1	英伟达GPU-Windows	● 已发布	本地部署	2021-09-27 17:34	导出SDK

## 数据管理

### 数据导入

#### 数据准备

准备训练所需的训练数据，结合期望得到的模型设计训练数据集的分类或标签

## 设计分类

对于图像分类任务，需确认分类如何设计，每个分类为你希望识别出的一种结果，如您需要识别动物，则可以以“dog”、“cat”等分别作为一个分类。

注意：每张图片都应属于一个分类，一个模型最多支持1000个分类，标签名由数字、中英文、中/下划线组成，长度上限256字符。

基于设计好的分类准备图片，有如下要求：

1. 每个分类需要准备20张以上图片，如果需要较好的效果，建议每个分类准备不少于100张图片
2. 训练图片和实际场景要识别的图片拍摄环境一致，举例：如果实际要识别的图片是摄像头俯拍的，训练时也需要使用俯拍角度的图片
3. 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强

## 设计标签

对于物体检测、实例分割任务，需要确认标签如何设计，每种需要识别的目标为一个标签，一张图片中可以有多种标签出现。

注意：单个数据集的标签上限为1000种，标签名由数字、中英文、中/下划线组成，长度上限256字符。

基于设计好的物体检测准备图片，有如下要求：

1. 训练图片和实际场景要识别的图片拍摄环境一致，举例：如果实际要识别的图片是摄像头俯拍的，训练图片就不能用网上下载的目标正面图片
2. 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
3. 每个模型训练图片量不得低于4张，每个标签建议标注50个框以上

## 数据导入

在数据总览页下找到创建完成的数据集点击【导入】

版本	数据集ID	数据量	标注类型	标注状态	操作
V1	7	0	图像分类	0% (0/0)	导入 导出

## 导入要求

图片要求：

- 支持图片类型包括jpg、png、bmp、jpeg，图片大小限制在14M以内
- 图片长宽比在3：1以内，其中最长边小于4096px，最短边大于30px

路径要求：

- 无标注信息：导入请确保全部图片保存知同一层文件目录下
- 有标注信息：导入请确保将全部图片与对应标注信息保存至同一层文件目录。该目录下子文件目录及非相关内容（例如压缩包）不导入

## 导入方式

数据导入无标注信息图片以及有标注信息图片的导入

注：数据导入后依然存储在您设备本地导入路径下，飞桨EasyDL桌面版不会调整您的数据存储路径，因此如更改本地存储路径下的图片将会导致数据集发生变动，如有正使用当前数据集训练的任务，将会导致任务失败。

## 无标注信息图片导入

进入数据导入界面，在数据标注状态中选择无标注信息



选择训练数据存储的文件夹，选择完成后数据集即导入完成

如需导入多个文件目录的数据，可多次导入

### 创建信息

数据集ID	7	版本号	V1
备注	<input type="text" value=""/>		

### 标注信息

标注类型	图像分类	标注模板	单图单标签
数据总量	0	已标注	0
标签个数	0	目标数	0

### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

- 提示：** 1.导入后请避免改动本地该数据，以免影响数据标注、模型训练功能正常使用
- 2.每次导入仅支持选择唯一目录，如您想快速体验一站式功能，可联网下载已标训练数据样例 [图像分类训练数据集\(coco格式\)](#)

### 有标注信息图片导入

有标注信息导入支持以文件夹命名分类、VOC格式、COCO格式以及平台自定义格式四种

#### 以文件夹命名分类导入

数据标注状态选择有标注信息，并选择标注格式为以文件夹命名分类

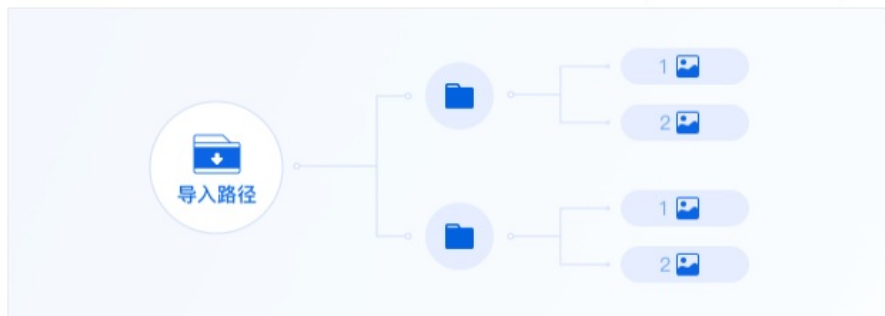
注：以文件夹命名分类仅支持图像分类任务

数据标注状态  无标注信息  有标注信息

导入方式

标注格式  以文件夹命名分类  voc  coco  平台自定义

以文件夹命名分类方式导入，导入路径下的每一个子文件夹将代表一个分类，子文件夹的名称将代表分类名，子文件夹下的图片将被视为当前分类下的数据



#### VOC格式导入

数据标注状态选择有标注信息，并选择标注格式为VOC格式

### 导入数据

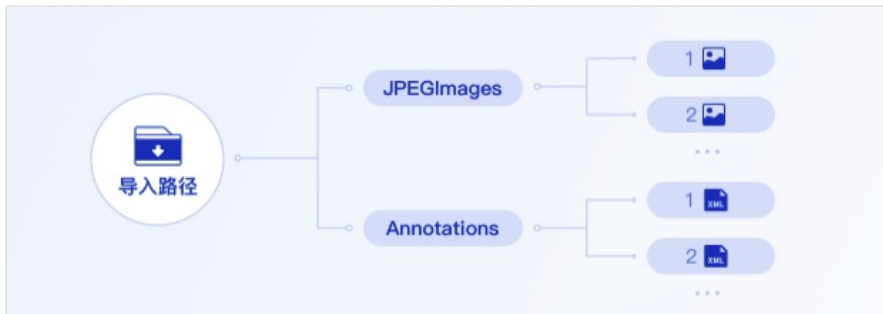
数据标注状态  无标注信息  有标注信息

导入方式

标注格式  以文件夹命名分类?  voc?  coco?  平台自定义?

以VOC格式导入，导入路径下应包含JPEGImages以及Annotations两个子文件夹，JPEGImages下存储图片数据，Annotations下存储xml格式的标注文件，且图片与标注信息一一对应

- 1.导入路径内需包括JPEGImages, Annotations两个文件夹，分别包括图片源文件 (jpg/png/bmp/jpeg) 及与图片具有相同名称的对应标注文件(xml后缀) [下载标注格式样例](#)
- 2.标注文件中标签由数字、中英文、中/下划线组成,长度上限256字符
- 3.图片源文件大小限制在14M内，长宽比在3:1以内，其中最长边需要小于4096pX，最短边需要大于30px



### COCO格式导入

数据标注状态选择有标注信息，并选择标注格式为COCO格式

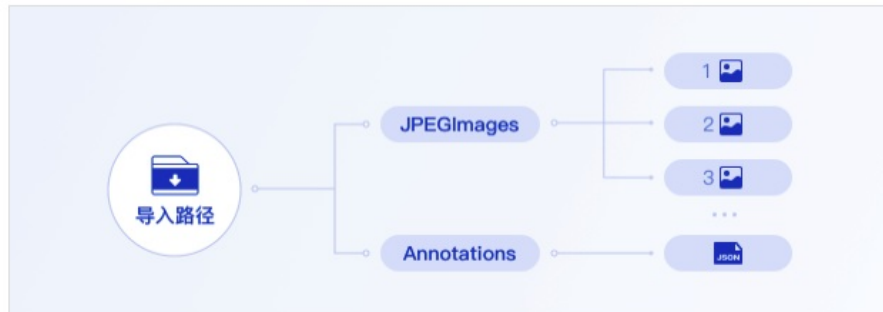
数据标注状态  无标注信息  有标注信息

导入方式

标注格式  以文件夹命名分类?  voc?  coco?  平台自定义?

以COCO格式导入，导入路径下应包含JPEGImages以及Annotations两个子文件夹，JPEGImages下存储图片数据，Annotations下存储Json格式的一个标注文件，所有图片的标注信息均存储在一个Json文件中

- 1.导入路径内需包括JPEGImages, Annotations两个文件夹，分别包括图片源文件 (jpg/png/bmp/jpeg) 及唯一一个命名为coco\_info.json的标注文件 [下载标注格式样例](#)
- 2.标注文件中标签由数字、中英文、中/下划线组成,长度上限256字符
- 3.图片源文件大小限制在14M内，长宽比在3:1以内，其中最长边需要小于4096pX，最短边需要大于30px



### 平台自定义格式导入

数据标注状态选择有标注信息，并选择标注格式为平台自定义格式

### 导入数据

数据标注状态  无标注信息  有标注信息

导入方式

标注格式  以文件夹命名分类  voc  coco  平台自定义

以平台自定义格式导入，导入路径不包含子文件夹，图片数据及标注文件均直接存储在导入路径下，标注信息以Json格式与图片一一对应

- 1.导入路径内需包括图片源文件（jpg/png/bmp/jpeg）及同名的json格式标注文件 [下载标注格式样例](#)
- 2.标注文件中标签由数字、中英文、中/下划线组成,长度上限256字符
- 3.图片源文件大小限制在14M内，长宽比在3:1以内，其中最长边需要小于4096px，最短边需要大于30px



导入路径选择完成后，点击【确认并返回】即完成数据导入

### 查看数据集

数据标注完成后可在数据总览页查看数据情况

#### 查看数据集

数据总览页展示数据集名称、数据集ID、数据量、标注类型、标注状态，鼠标放置在省略号处可查看详细数据集信息

物体检测数据集 <span>🗑️ 删除</span>				
数据集ID	数据量	标注类型	标注状态	操作
15 <span>⋮</span>	69	物体检测	100% (69/69)	<a href="#">查看与标注</a> <a href="#">导入</a> <a href="#">导出</a>

**创建信息**

数据集ID: 15      备注: [📝](#)

---

**标注信息**

标注类型: 物体检测      标注模板: 矩形框标注

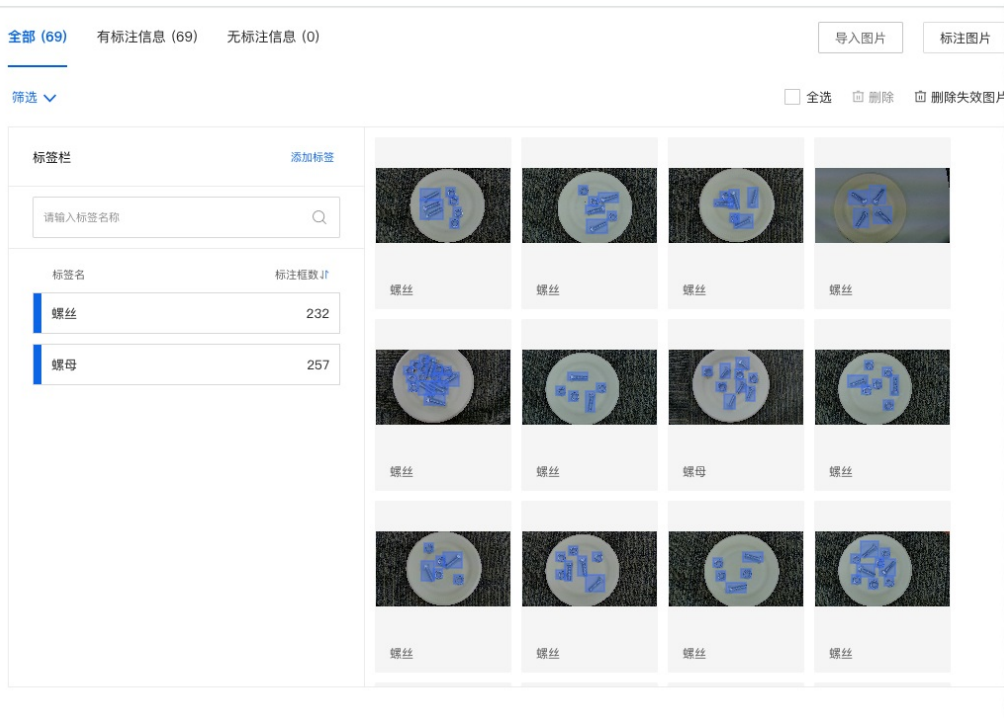
数据总量: 69      已标注: 69 (进度100.00%)

标签个数: 2      目标数: 0

点击【查看与标注】可查看数据集详情

支持查看数据集中数据标注情况，可手动删除数据集中的图片

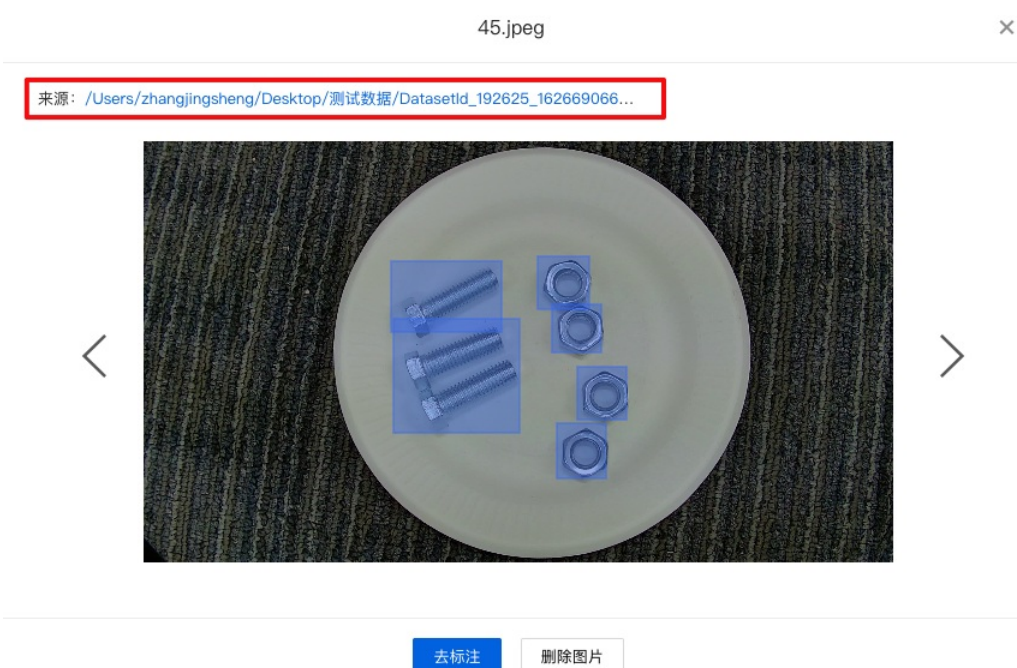
支持一键删除失效图片



点击图片右下角查看大图, 可查看放大图片



放大图片后可查看图片本地地址, 点击可打开当前文件存储路径



### 数据集导出

支持将数据集导出到指定路径下, 点击【导出】将导出数据源文件以及当前数据集已有的标注文件, 支持将标注数据导出为voc、coco或平台自定义格式

## 创建导出任务



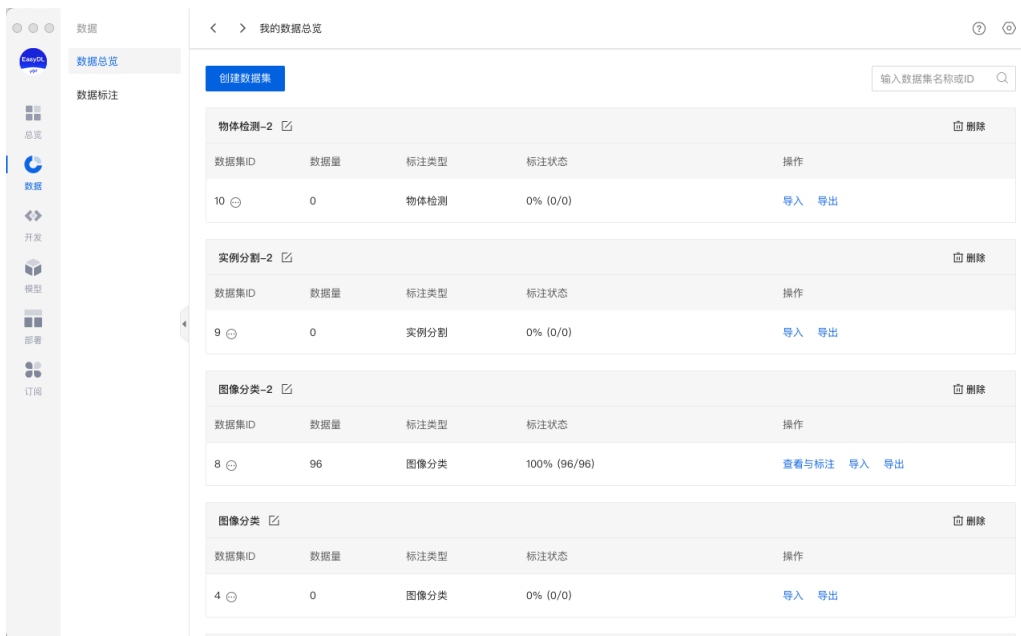
导出数据集名称 物体检测数据集V1

导出数据 导出全部数据，包含源文件及已有的标注文件

标注格式  VOC  COCO  平台自定义导出路径  上次导出路径: /Users/zhangjingsheng/Deskto...

## 创建数据集

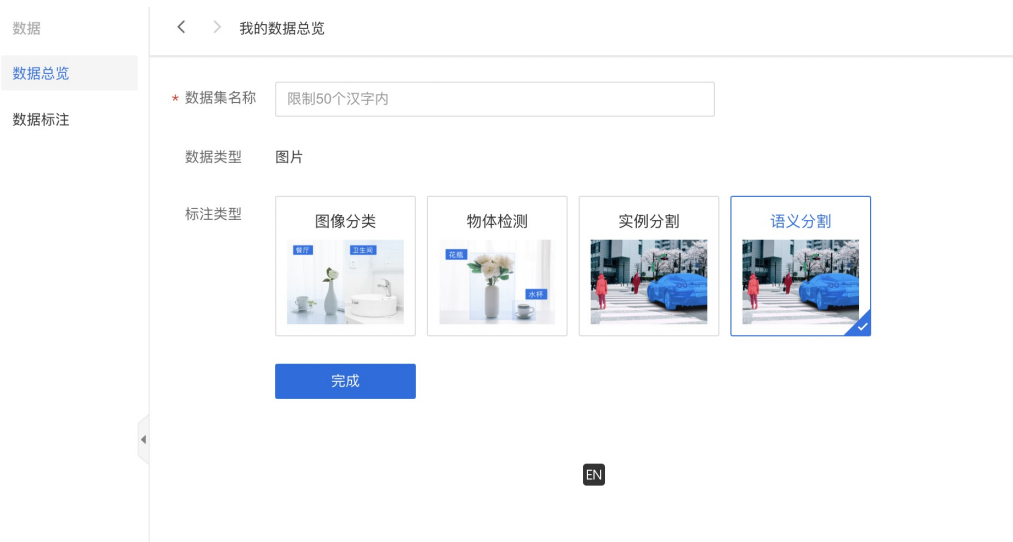
数据模块主要用于训练数据的导入、标注、导出以及管理



发起训练任务前，需要提前准备训练数据，在数据总览页点击创建数据集



输入数据集名称，选择数据集标注类型后点击完成即可完成数据集创建，飞桨EasyDL桌面版提供图像分类、物体检测、实例分割三种标注类型可供选择



数据集创建完成后可在数据总览页查看创建完成的数据集



## 数据标注

飞桨EasyDL桌面版提供了图像分类、物体检测、实例分割的数据标注能力，可在本地高效完成数据标注，数据标注启动前，需先创建数据集中的标签

### 🔗 设定标签

在数据总览页找到导入完成的数据集，并点击【查看与标注】

数据集ID	数据量	标注类型	标注状态	操作
12	96	图像分类	0% (0/96)	查看与标注 导入 导出

在查看与标注页面点击【添加标签】创建本次数据中的数据标签

全部 (96) 有标注信息 (0) 无标注信息 (96)

筛选 ▾

标签栏 添加标签

根据图片内容，选择标签

在左侧列表栏可看到已添加的标签，并支持搜索标签名称、按照数据量多少对标签进行排序

筛选

标签栏 添加标签

请输入标签名称 Q

根据图片内容，选择标签

标签名	数据量
cat	0
dog	0

### 数据标注

#### 图像分类

图像分类场景提供常规标注与批量标注两种标注方式

#### 常规标注

标签添加完成后，在【查看与标注】界面，点击图片右下角标注icon进入标注界面

全部 (96)    有标注信息 (0)    无标注信息 (96)


筛选

标签栏 添加标签

请输入标签名称 Q

根据图片内容，选择标签

标签名	数据量
cat	0
dog	0



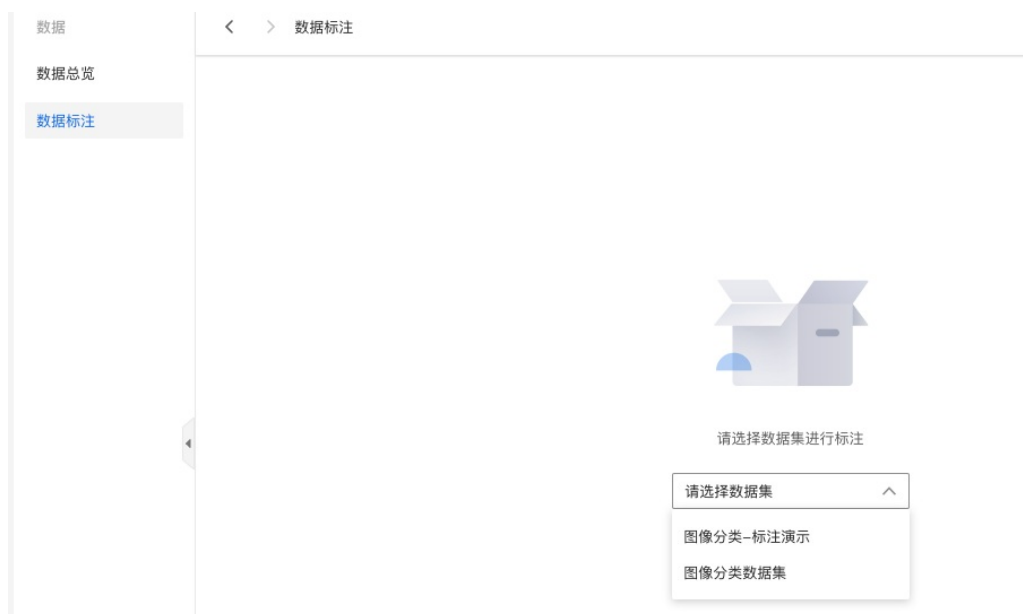
无标签 [编辑] [放大]    无标签

点击【编辑icon】即可进入“单张图片标注模式”

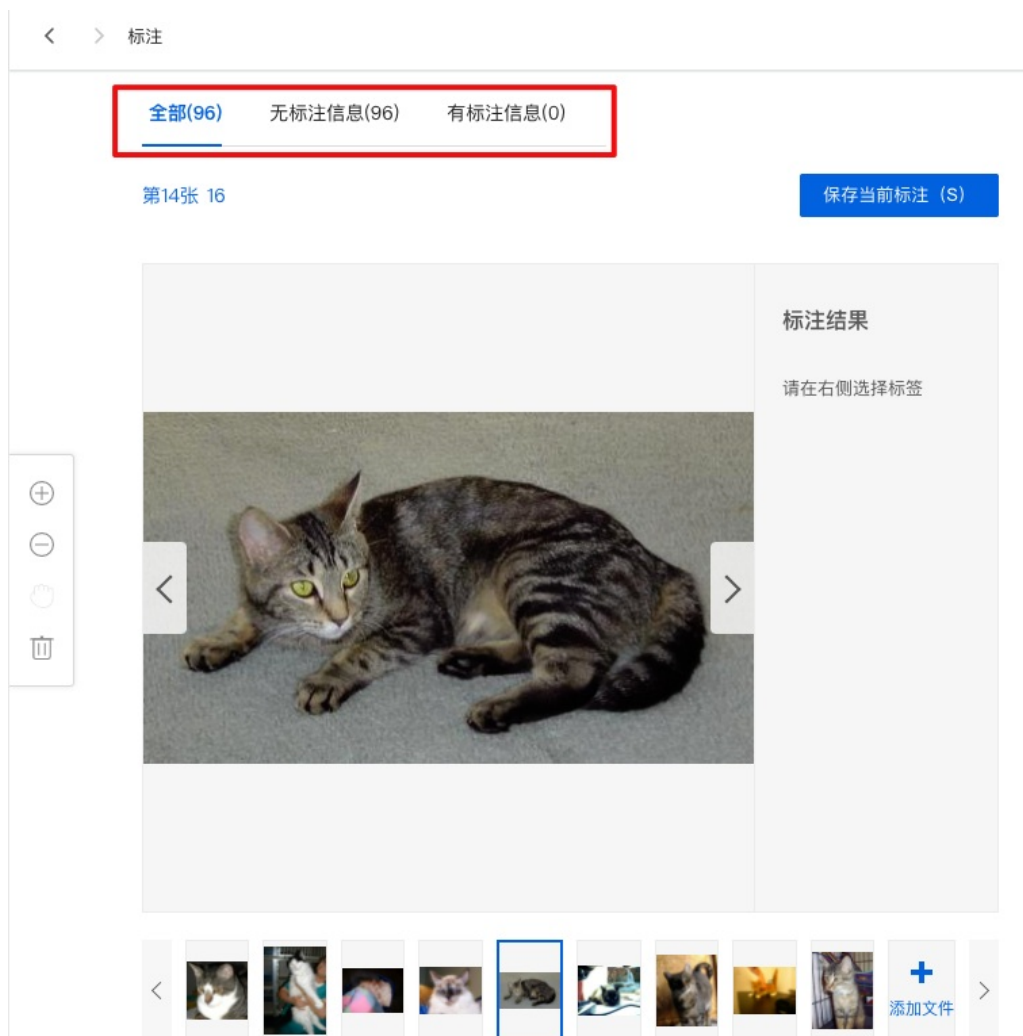
无标签    无标签

或在左侧导航栏点击数据标注，选择需要标注的数据集进入到标注界面



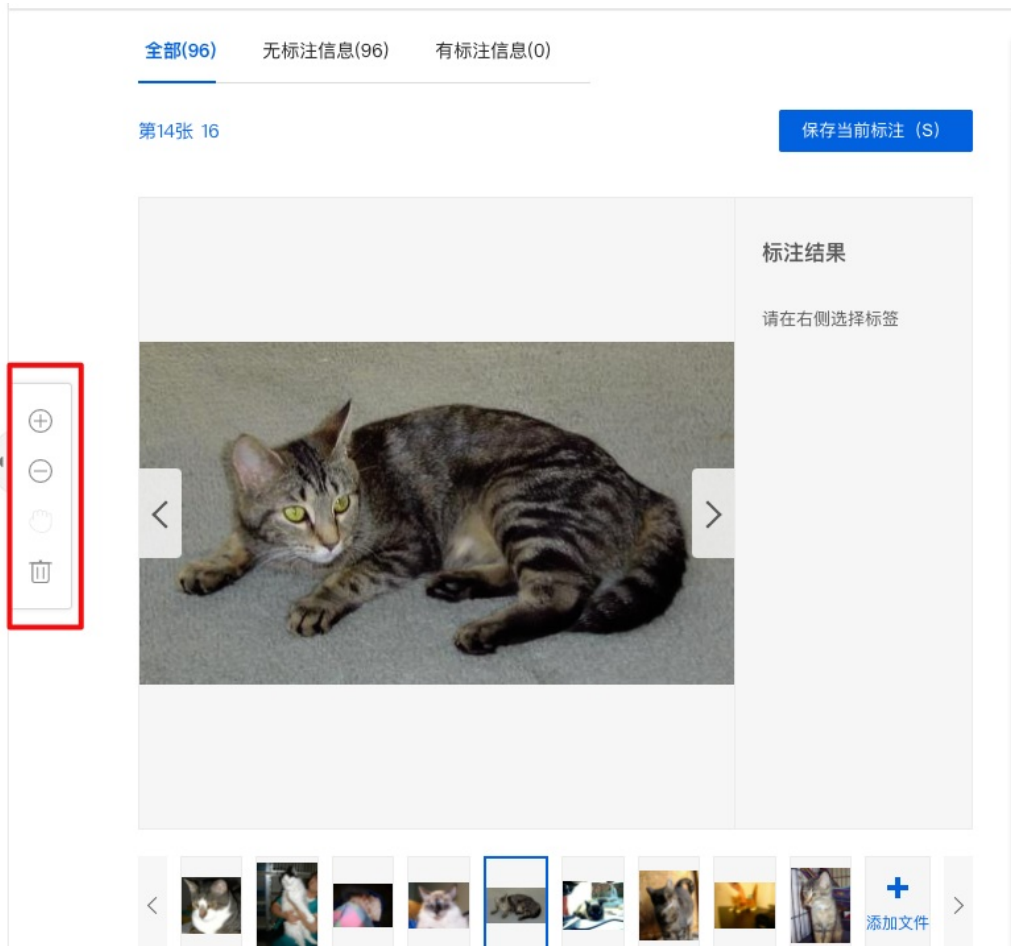


进入标注界面后，可在上方切换查看全部数据、无标注信息数据或有标注信息数据

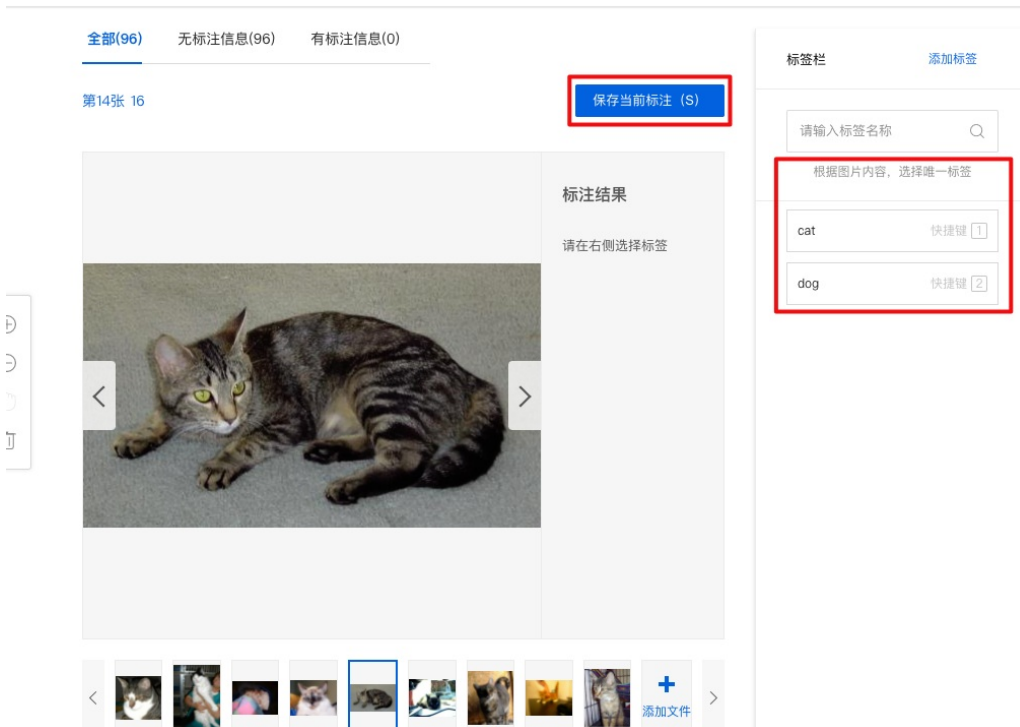


左侧工具栏提供图片放大、缩小、移动或删除功能





根据图片内容在左侧正确选择图片标签并点击【保存当前标注】完成数据标注



### 批量标注

除常规标注外, 飞桨EasyDL桌面版提供了批量标注工具, 可提高数据标注效率

在数据总览页找到已导入数据的数据集, 点击【查看与标注】

猫狗识别 <input checked="" type="checkbox"/> 数据集组ID: 6 <span style="float: right;">删除</span>					
版本	数据集ID	数据量	标注类型	标注状态	操作
V1 <input checked="" type="checkbox"/>	6	96	图像分类	0% (0/96)	<span style="border: 1px solid red;">查看与标注</span> 导入 导出

点击【批量标注】进入批量标注界面

全部 (96) 有标注信息 (0) 无标注信息 (96) 导入图片 批量标注

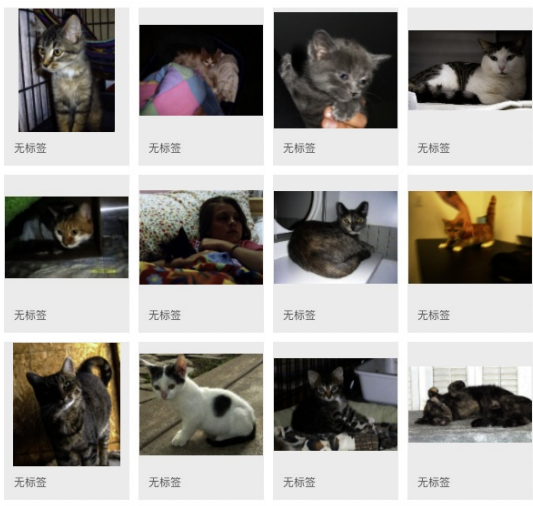
筛选  全选  删除  删除失效图片

标签栏 添加标签

请输入标签名称

根据图片内容, 选择标签

标签名	数据量
dog	0
cat	0

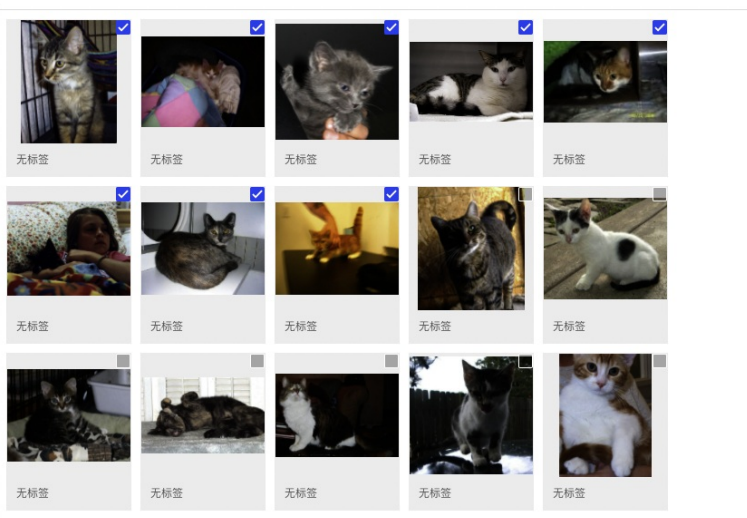


在批量标注界面支持多选或全选本页图片，图片勾选完成后选择相应标签即完成勾选图片的标注

全部 (96) 有标注信息 (0) 无标注信息 (96) 导入图片 关闭批量标注

筛选  全选  删除  删除失效图片

已选择8个 取消选择



标签栏 添加标签

请输入标签名称

根据图片内容, 选择标签

标签名	数据量
dog	0
cat	0

### 物体检测

物体检测数据标注方式为通过矩形框框出图片中需要检测的目标，并对标注框选择标签名，以下为物体检测数据标注流程

在【查看与标注】界面，点击图片右下角标注icon进入标注界面

全部 (69) 有标注信息 (0) 无标注信息 (69) 导入图片 标注图片

筛选 ☑ 全选 🗑 删除 🗑 删除失效图片

标签栏 添加标签

请输入标签名称

标签名	标注框数
螺母	0
螺丝	0

或在左侧导航栏点击数据标注，选择需要标注的数据集进入到标注界面

数据 < > 数据标注

数据总览

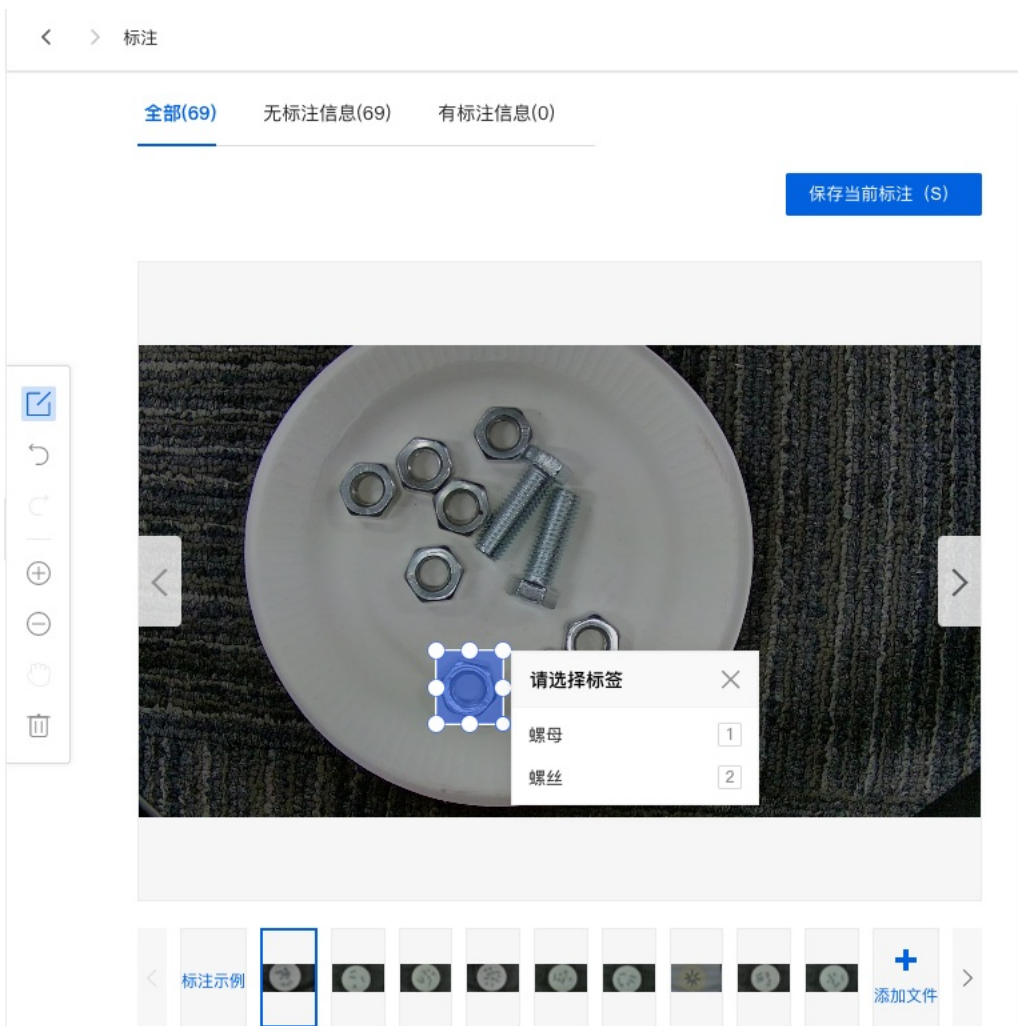
数据标注

请选择数据集进行标注

请选择数据集

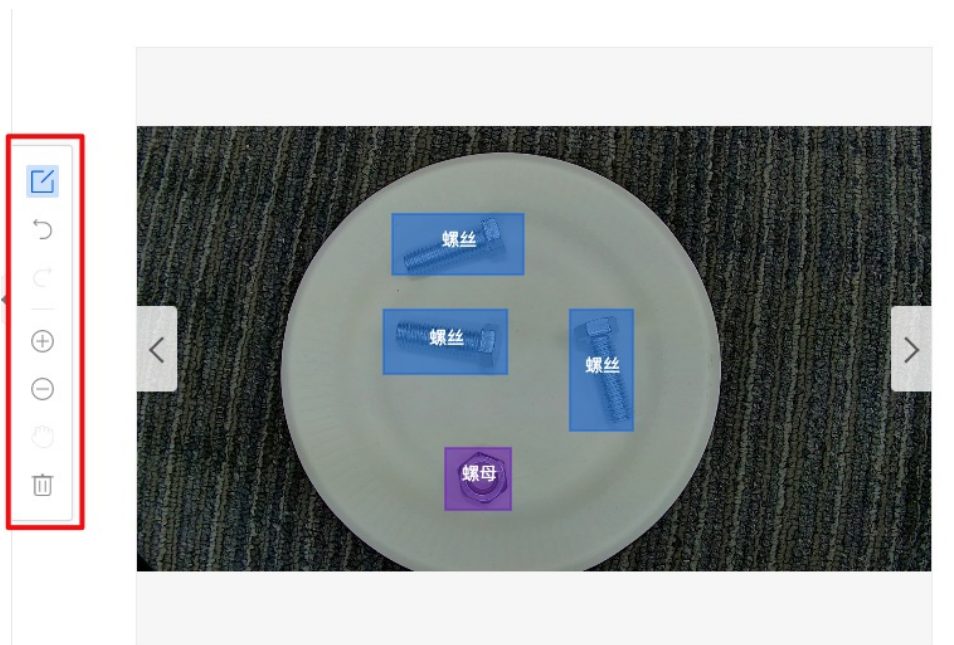
- 图像分类-标注演示
- 图像分类数据集

在标注界面将需要检测的目标框选出，并为目标选择对应的标签



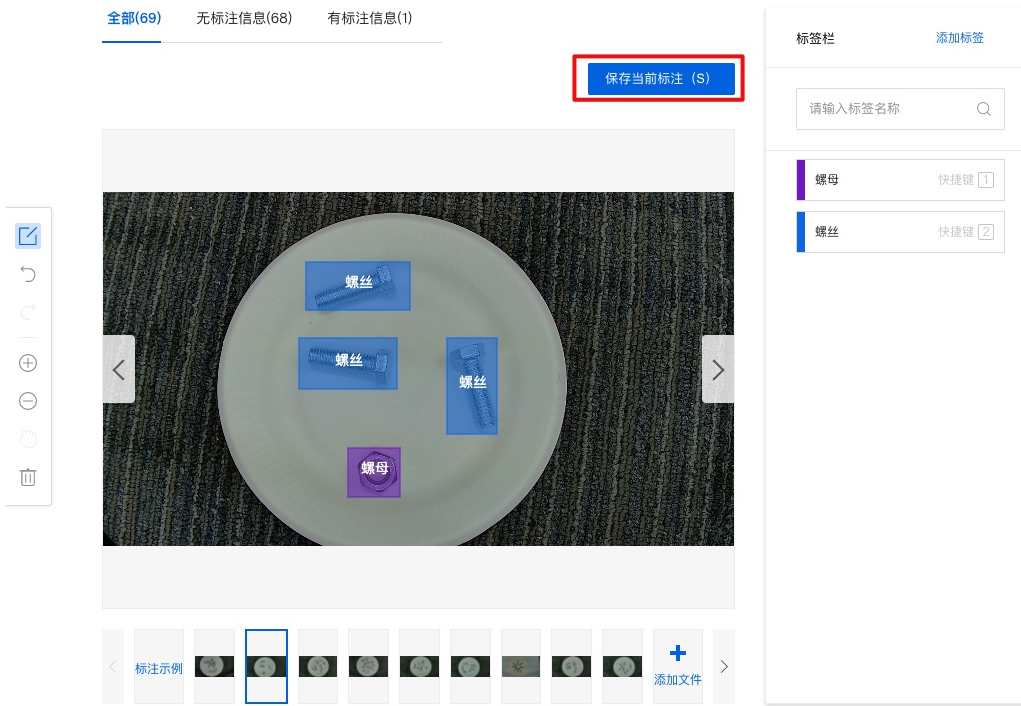
标注过程中应尽可能贴合目标物体，避免未完全框选目标物体或矩形框远大于目标物体，可通过拉动标注框边界调整标注框大小

左侧工具栏支持用户放大、缩小、移动、删除图片或撤销上一步操作



单张图片标注完成后点击【保存当前标注】即完成单张图片标注





### 实例分割

为实例分割场景数据标注过程负责，飞桨EasyDL-桌面版提供了多种数据标注工具，可根据实际使用需求选择使用

### 自动识别轮廓标注工具

标注界面工具栏第一个工具为自动识别轮廓工具，开启自动识别工具后，鼠标左键点击目标物体后可自动识别目标物体轮廓



对于误识别区域，可通过右键点击来进行修正



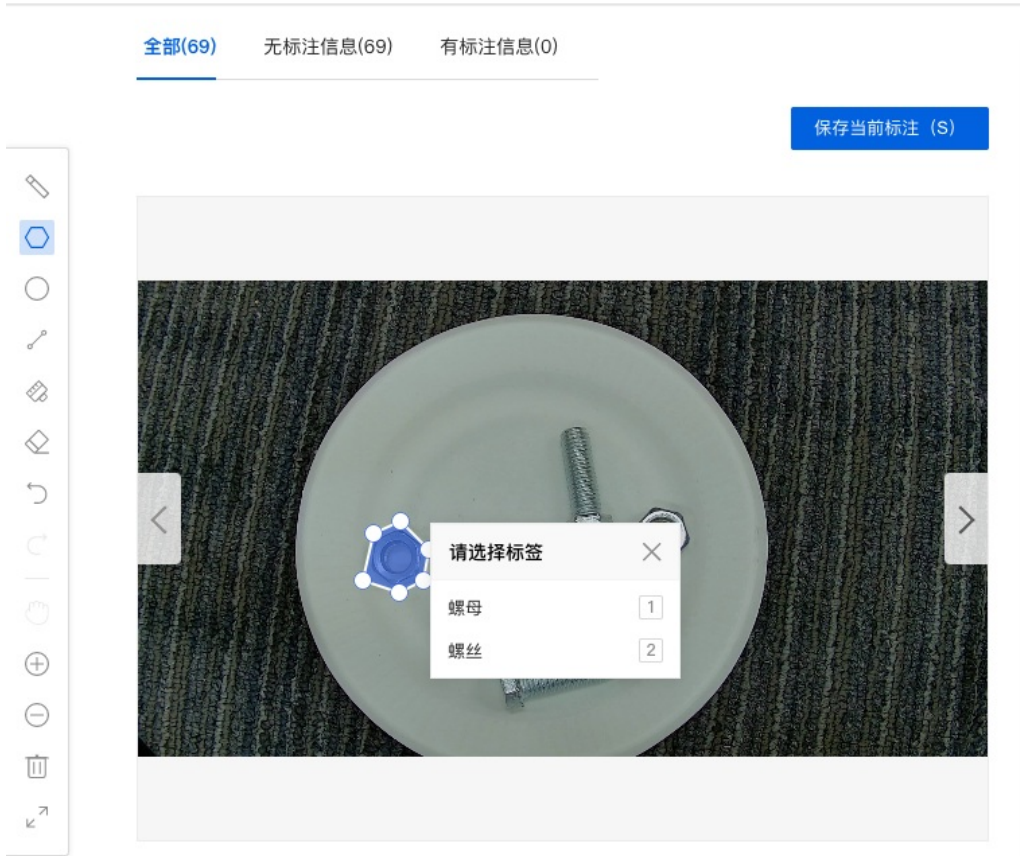
轮廓标注正确后点击【确认轮廓】并为目标添加标签



### 多边形标注

标注界面工具栏第二个工具为多边形标注工具，多边形标注工具是通过多点首尾相连的方式将目标物体圈出的方式完成标注，适用于目标物体结构相对负责的情况

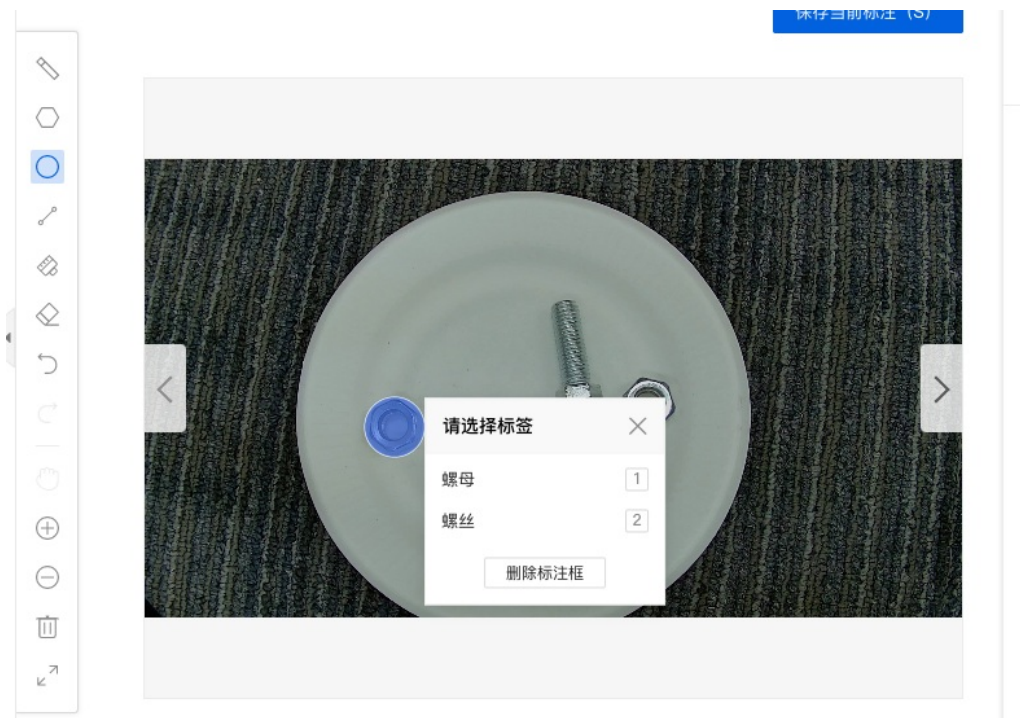
标注完成后可拖动任意一点调整标注结果，标注结果符合要求后为当前标注物选择标签



#### 圆形标注工具

标注界面工具栏第三个工具为圆形标注工具，圆形标注工具是通过拉取一个圆将目标物体的方式完成标注，适用于目标物体形状是圆形或近似圆形的情况

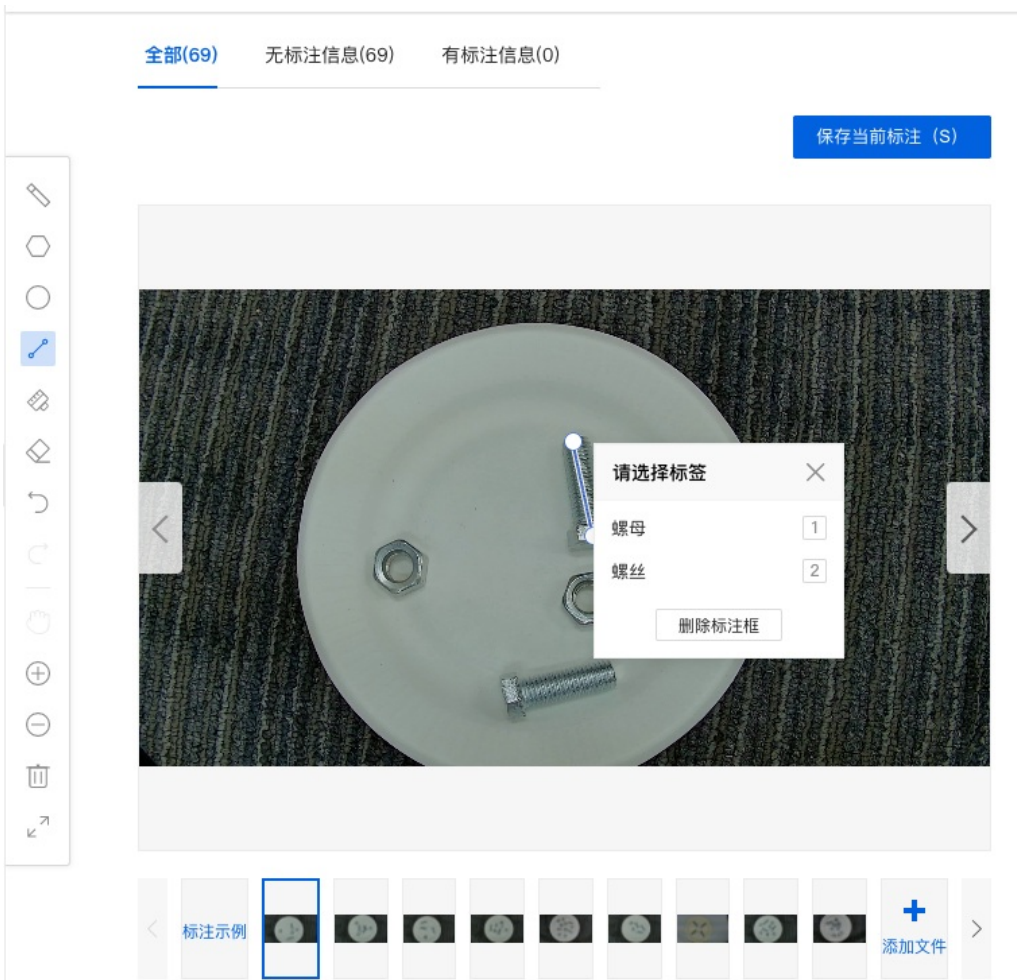
标注完成后可拖动调整圆形位置或拖动圆形边缘调整圆形大小，标注结果符合要求后为当前标注物选择标签



#### 直线标注工具

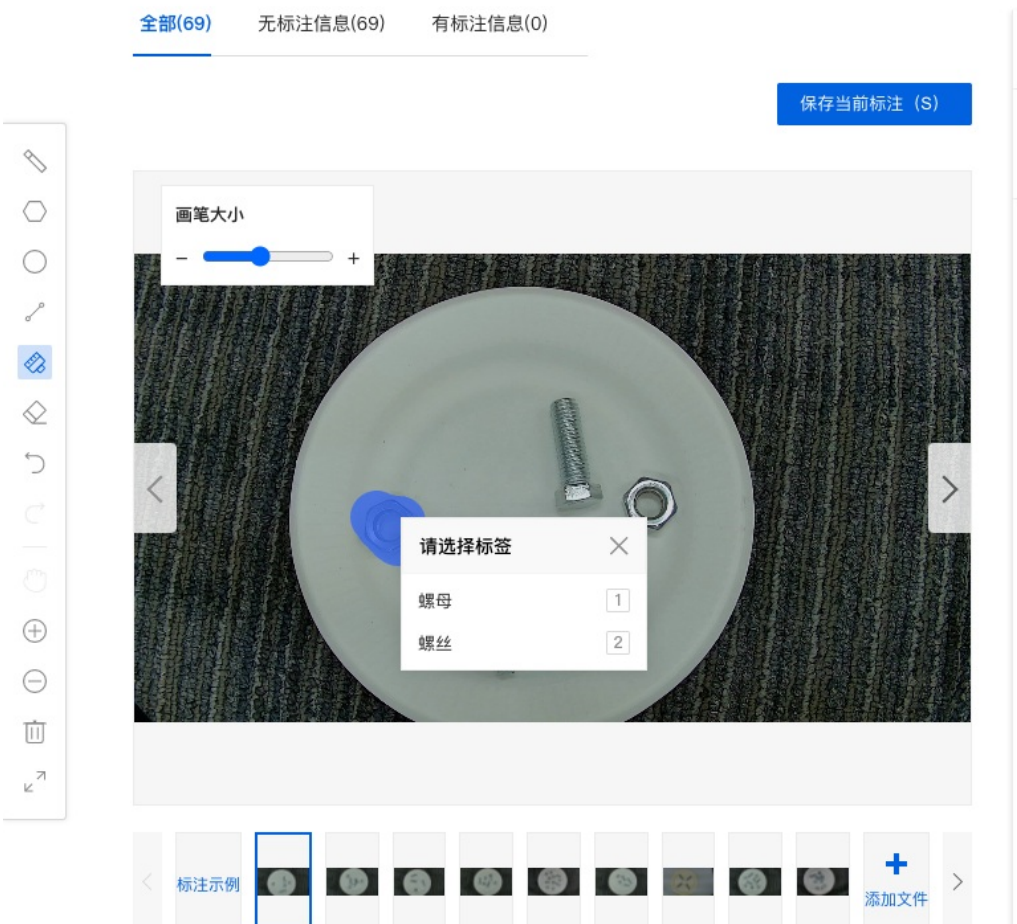
标注界面工具栏第四个工具为直线标注工具，直线工具是通过选择两点连成一条直线的方式完成标注，适用于目标物体形状是直线的情况





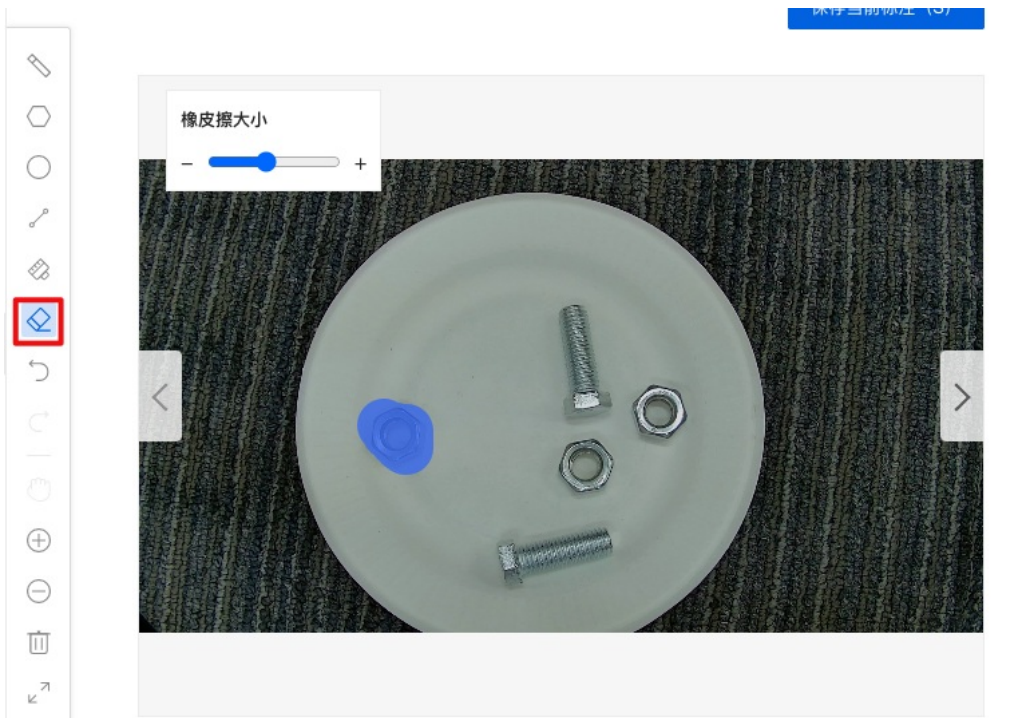
### 画笔标注工具

标注界面工具栏第五个工具为画笔标注工具，画笔标注工具是通过画笔覆盖一定区域并进行标注的方式，可在标注界面调整画笔大小



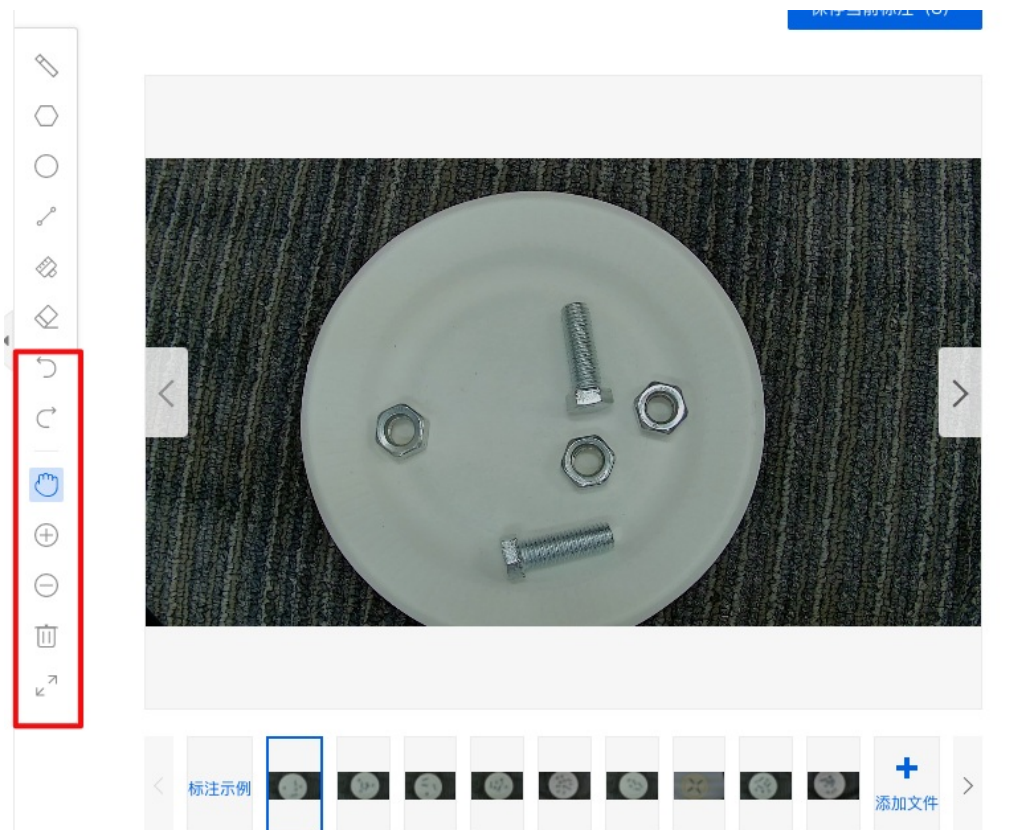


如标注区域大于目标物体，可通过橡皮擦将多余区域擦除



#### 其他辅助工具

除标注方式外，飞桨EasyDL桌面版仍提供了其他辅助工具，如：操作撤回、操作恢复、移动图片、放大图片、缩小图片、删除以及图片最大化



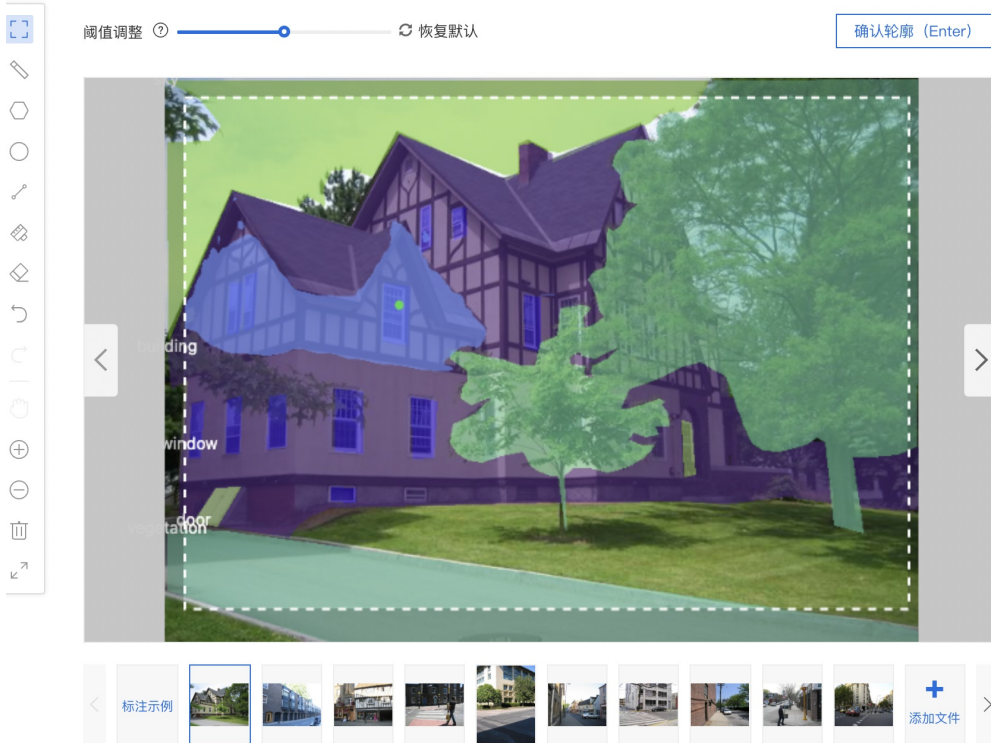
#### 语义分割

为语义分割场景数据标注过程负责，飞桨EasyDL-桌面版提供了多种数据标注工具，可根据实际使用需求选择使用。自动识别轮廓标注工具

标注界面工具栏第一个工具为自动识别轮廓工具，开启自动识别工具后，鼠标左键点击目标物体后可自动识别目标物体轮廓，对于误识别区域，可通过右键点击来进行修正，轮廓标注正确后点击【确认轮廓】并为目标添加标签

我的数据总览 > 【图片】语义分割test/V3/查看与标注 > 标注

使用说明：1.通过绘制矩形定义目标所在范围，左击目标中心位置即可识别目标轮廓；2.若识别结果不精准，可通过添加正负点来调整，左击增加“漏识别区域”，右击可减去“误识别区域”；3.对识别结果满意之后，点击“确认轮廓”按钮并添加标签



### 多边形标注

标注界面工具栏第二个工具为多边形标注工具，多边形标注工具是通过多点首尾相连的方式将目标物体圈出的方式完成标注，适用于目标物体结构相对复杂的情况

标注完成后可拖动任意一点调整标注结果，标注结果符合要求后为当前标注物选择标签

### 圆形标注工具

标注界面工具栏第三个工具为圆形标注工具，圆形标注工具是通过拉取一个圆将目标物体的方式完成标注，适用于目标物体形状是圆形或近似圆形的情况

标注完成后可拖动调整圆形位置或拖动圆形边缘调整圆形大小，标注结果符合要求后为当前标注物选择标签

### 直线标注工具

标注界面工具栏第四个工具为直线标注工具，直线工具是通过选择两点连成一条直线的方式完成标注，适用于目标物体形状是直线的情况

### 画笔标注工具

标注界面工具栏第五个工具为画笔标注工具，画笔标注工具是通过画笔覆盖一定区域并进行标注的方式，可在标注界面调整画笔大小

如标注区域大于目标物体，可通过橡皮擦将多余区域擦除

### 其他辅助工具

除标注方式外，飞桨EasyDL桌面版仍提供了其他辅助工具，如：操作撤回、操作恢复、移动图片、放大图片、缩小图片、删除以及图片最大化



## 开发与训练

### 零代码开发

#### AutoDL模式开发

为零AI开发基础的用户提供的建模方式，内置基于百度文心大模型的成熟预训练模型，可针对用户数据进行算法自动优化，助用户使用少量数据也能获得具备出色效果与性能模型。

#### 训练配置

AutoDL模式下支持对模型文件导出类型、模型SDK部署方式、训练使用算法类型等内容进行设置。

**训练配置**

高级版专享

导出类型  仅导出源文件  导出源文件与离线SDK ?

选择算法 ?  超高性能 ?  高性能 ?  高精度 ? [查看算法性能及适配硬件](#)

高级训练配置

以下高级配置选项一般情况推荐不做更改，如实际任务场景有需求，请根据实际调整

输入图片分辨率 ?

epoch ?  自动  手动设置

训练完成后同步发布为模型 ?  是  否

\*模型名称

模型类型

模型版本

版本描述1

0 / 100

#### 导出类型

- 导出模型源文件：训练完成后支持将模型源文件导出，模型源文件可通过Paddle-Inference转化至实际应用场景中使用
- 导出模型源文件与离线SDK：训练完成的模型可发布为在服务器、小型设备、专项适配硬件上直接部署的SDK，覆盖主流芯片与操作系统，可直接用于业务集成，省去繁琐转化过程

模型SDK适配设备类型详见 AutoDL模式算法适配硬件

## 部署方式

- 可发布为在服务器、小型设备、专项适配硬件上直接部署的SDK
- 模型SDK适配设备类型详见 AotuDL模式算法适配硬件 [选择算法](#)
- **高精度**：相同训练数据情况下，训练出的模型准确率更高，训练及预测耗时更长，模型体积更大
- **高性能**：相同训练数据情况下，较高精度算法，训练及预测耗时更短，模型准确率平均比高精度算法低3%~5%
- **超高性能**：相对高性能算法，模型体积更小，CPU环境预测速度提升近60%，GPU环境预测速度提升近10%

模型精度与训练数据量的大小、质量成正比相关 训练速度与训练数据量的大小以及训练设备的算力情况相关

## 高级训练配置

高级训练配置开启状态下支持手动设置图像分辨率及迭代轮次（epoch）

- **图像分辨率**：如检测目标在图片中占比较小，可适当调高图片分辨率以得到更高的精度，如检测目标在图片中占比较大，可适当调低图片分辨率以得到更高的训练效率
- **迭代轮次（epoch）**：训练集完整参与训练的次数，如模型精度较低，可适当调高迭代轮次，使模型训练更完整

系统会根据本地训练资源以及所选算法类型自动采用最优组合，通常情况下推荐不做更改，如实际任务场景有需要，则可根据实际需求调整

## 训练完成后同步发布为模型

任务训练完成后可通过评估、校验验证任务效果，任务效果满足实际使用要求后发布为模型完成模型部署流程，如当前任务已经过多轮迭代且任务效果较有保证可勾选训练完成后同步发布为模型，并输入发布为的模型名称以及版本描述，训练成功后将会自动发布为模型

任务与模型一一对应，如当前任务已有版本发布为模型，则当前任务下的其他版本发布时仅支持发布在当前模型下

## 数据配置

### 添加训练数据

- 先选择数据集，再按分类选择数据集里的图片，可从多个数据集选择图片
- 训练时间与数据量大小有关，数据量越大，训练耗时越长

#### Tips：

- 图像分类任务类型，如只有1个分类需要识别，或者实际业务场景所要识别的图片内容不可控，可以在训练前勾选“增加识别结果为[其他]的默认分类”。勾选后，模型会将与训练集无关的图片识别为“其他”
- 如果同一个分类的数据分散在不同的数据集里，可以在训练时同时从这些数据集里选择分类，模型训练时会合并分类名称相同的图片

### 添加自定义验证集

AI模型在训练时，每训练一批数据会进行模型效果检验，以某一张验证图片作为验证数据，通过验证结果反馈去调节训练。可以简单地把AI模型训练理解为学生学习，训练集则为每天的上课内容，验证集即为每周的课后作业，质量更高的每周课后作业能够更好的指导学生学习和找寻自己的不足，从而提高成绩。同理AI模型训练的验证集也是这个功效。

注：学生的课后作业应该与上课内容对应，这样才能巩固知识。因此，验证集的标签也应与训练集完全一致。

### 添加自定义测试集

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反

映模型效果。

注：期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可。

## 配置数据增强策略

深度学习模型的成功很大程度上要归功于大量的标注数据集。通常来说，通过增加数据的数量和多样性往往能提升模型的效果。当在实践中无法收集到数目庞大的高质量数据时，可以通过配置数据增强策略，对数据本身进行一定程度的扰动从而产生“新”数据。模型会通过学习大量的“新”数据，提高泛化能力。

你可以在「默认配置」、「手动配置」2种方式中进行选择，完成数据增强策略的配置。

### 默认配置

如果你不需要特别配置数据增强策略，就可以选择默认配置。后台会根据你选择的算法，自动配置必要的数据增强策略。

### 手动配置

飞桨EasyDL提供了大量的数据增强算子供开发者手动配置。你可以通过下方的算子功能说明或训练页面的效果展示，来了解不同算子的功能：

算子名	功能
ShearX	剪切图像的水平边
ShearY	剪切图像的垂直边
TranslateX	按指定距离（像素点个数）水平移动图像
TranslateY	按指定距离（像素点个数）垂直移动图像
Rotate	按指定角度旋转图像
AutoContrast	自动优化图像对比度
Contrast	调整图像对比度
Invert	将图像转换为反色图像
Equalize	将图像转换为灰色值均匀分布的图像
Solarize	为图像中指定阈值之上的所有像素值取反
Posterize	减少每个颜色通道的bits至指定位数
Color	调整图像颜色平衡
Brightness	调整图像亮度
Sharpness	调整图像清晰度
Cutout	通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例

### 效果展示



剪切图像的水平边，能更好地识别发生了水平方向形变的图像



关闭

## 训练环境配置



飞桨EasyDL支持以Intel CPU，或可通过NVIDIA旗下不同型号显卡加速训练。

如果您的计算机有NVIDIA® GPU，且需要使用GPU环境进行训练，请确保满足以下条件：

- Windows 7/8/10/11：需安装 CUDA 11.2 与 cuDNN v8.2.1
- Ubuntu 16.04/18.4/20.4：需 CUDA 11.2 与 cuDNN v8.1.1
- CentOS 7：需 CUDA 11.2 与 cuDNN v8.1.1

## CUDA、cuDNN安装指南

您可参考NVIDIA官方文档了解CUDA和CUDNN的安装流程和配置方法，详见：

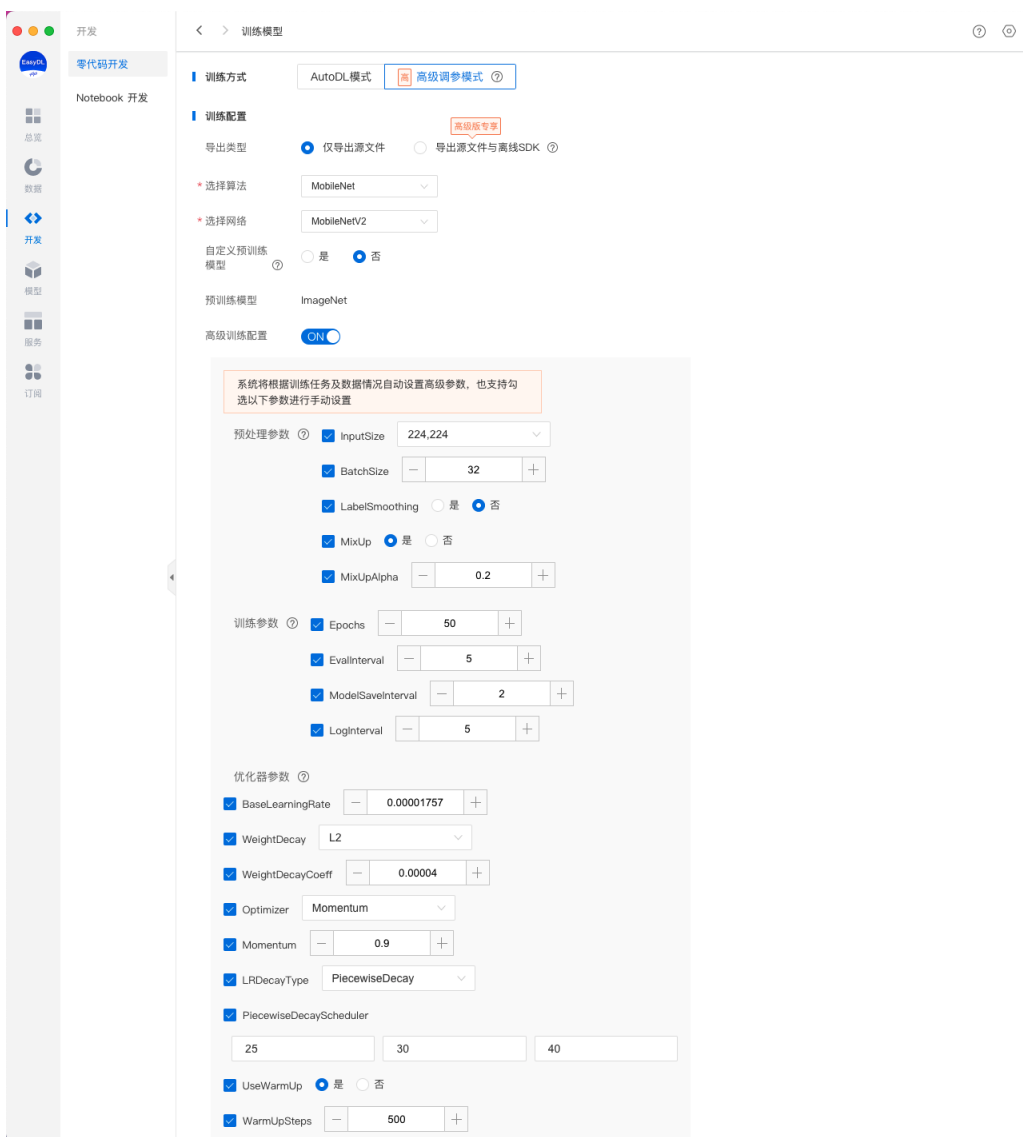
- CUDA 安装指南：<https://docs.nvidia.com/cuda/>
- cuDNN安装指南：<https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html>

## 🔗 预置模型调参模式开发

为有一定AI开发基础的开发者提供预置模型调参建模方式，涵盖ResNet、YOLO、PicoDet、MaskRCNN等近30种网络类型，适配大部分场景，开发者只需选择合适的预训练模型以及网络，根据自身经验进行调整，以获得更适合特定场景的模型。

## 训练配置

预置模型调参模式下支持对模型文件导出类型、训练使用算法与网络类型、网络参数等内容进行设置。



## 导出类型

- 导出模型源文件：训练完成后支持将模型源文件导出，模型源文件可通过Paddle-Inference转化至实际应用场景中使用
- 导出模型源文件与离线SDK：训练完成后可直接将模型发布为SDK包，可直接用于业务集成，省去繁琐转化过程

## 预置算法及网络选择

预置模型调参模式中，现已含盖ResNet50\_vd、YOLOv3\_MobileNetV1、SSD\_MobileNetV1、Mask\_RCNN\_R50\_vd\_FPN等图像分类、物体检测、实例分割3类场景下14类网络，后续还将进一步提升预置网络数量。

### 图像分类

算法	网络	适用场景	预训练模型
ResNet	ResNet50_vd	整体效果稳定、预测时间较短	ImageNet
SENet	SE_ResNeXt50_32x4d	整体效果稳定、预测时间较短	ImageNet
ResNext	ResNeXt101_32x16d_wsl	适用于数据量较大且准确率要求高时，预测时间将会更长	ImageNet
EfficientNet	EfficientNetB0_small	适用于要求预测时间或模型体积尽量小的场景	ImageNet
	EfficientNetB4	适用于数据量较大且准确率要求高时，预测时间将会更长	ImageNet
MobileNet	MobileNetV2	适用于要求预测时间或模型体积尽量小的场景	ImageNet
	MobileNetV3_large_x1_0	适用于要求预测时间或模型体积尽量小的场景	ImageNet

### 物体检测

算法	网络	适用场景	预训练模型
CascadeRCNN	Cascade_RCNN_R50_FPN	适用于模型精度要求更高的场景	COCO
FasterRCNN	Faster_RCNN_R50_FPN	适用于模型精度要求更高的场景	COCO
YOLOv3	YOLOv3_MobileNetV1	适用于预测性能要求更高的场景	COCO
	YOLOv3_DarkNet53	适用于预测性能和模型精度相对折中的场景	COCO
	YOLOv3_R50_vd_DCN	适用于模型精度要求更高的场景	COCO

### 实例分割

算法	网络	适用场景	预训练模型
MaskRCNN	Mask_RCNN_R50_vd_FPN	模型效果稳定，预测时间适中	COCO

## 高级训练配置（模型参数）

高级训练配置中提供了预处理参数、训练参数、优化器参数可供选择调整，系统将根据训练任务及数据情况自动设置高级参数，也支持勾选部分参数进行手动设置。

### 图像分类

#### 预处理参数

参数	说明
InputSize	输入图像分辨率，值越大训练时显存占用越大。
BatchSize	一次训练所选取的样本数，值越大训练时显存占用越大。
LabelSmooth	标签平滑，是机器学习领域的一种正则化方法，通常用于分类问题，目的是防止模型在训练时过于自信地预测标签，改善泛化能力差的问题。
MixUpAlpha	图像混合系数，MixUp是一种数据增强方式，通过将随机的两张样本按比例混合，在训练过程中不会出现非信息像素，从而能够提高训练效率。

#### 训练参数

参数	说明
Epochs	训练集完整参与训练的次数，迭代轮数越多训练耗时越长。
EvalInterval	训练中评估的次数，评估间隔越小评估越频繁，训练耗时越长。
ModelSaveInterval	训练中模型保存间隔，间隔越小训练过程中保存的模型越多。
LogInterval	训练日志输出间隔，间隔越小训练日志内容越多。

#### 优化器参数

参数	说明
BaseLearningRate	单BatchSize对应的学习率，会根据BatchSize和卡数线性增大。学习率是在梯度下降的过程中更新权重时的超参数。学习率过高会导致模型难以收敛；过低会可能导致模型收敛速度过慢，建议根据经验设定合理值
WeightDecay	分位L1和L2正则。L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
WeightDecayCoeff	分位L1和L2正则。L1或L2正则：L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
Optimizer	优化器是在梯度下降中影响参数更新方式的选项，有Momentum，RMSProp，Adam等。
Momentum	参数更新的计算公式中的动量因子
LRDecayType	学习率衰减策略，随着训练迭代数的增加，学习率慢慢减小，以帮助模型收敛，常见的策略有阶梯式下降（PiecewiseDecay），余弦下降（CosineDecay）等。
UseWarmUp	是否使用学习率热身策略。
WarmUpSteps	学习率热身迭代数，通过该迭代次数后，学习率慢慢增加到指定值。

## 物体检测

### 网络参数

参数	说明
KeepTopK	每个图像可以保留的总边界框数。
ScoreThreshold	过滤掉低置信度分数的边界框的阈值。
NmsThreshold	对检测框进行非极大值抑制(NMS)时使用的IOU阈值。
Anchors	基于锚框的检测框架里对预置锚框(anchor)大小的设置。

### 预处理参数

参数	说明
InputSize	输入图像分辨率，值越大训练时显存占用越大。
BatchSize	一次训练所选取的样本数，值越大训练时显存占用越大。

### 训练参数

参数	说明
Epochs	训练集完整参与训练的次数，迭代轮数越多训练耗时越长。
EvalInterval	训练中评估的次数，评估间隔越小评估越频繁，训练耗时越长。
ModelSaveInterval	训练中模型保存间隔，间隔越小训练过程中保存的模型越多。
LogInterval	训练日志输出间隔，间隔越小训练日志内容越多。

### 优化器参数



参数	说明
BaseLearningRate	单BatchSize对应的学习率，会根据BatchSize和卡数线性增大。学习率是在梯度下降的过程中更新权重时的超参数。学习率过高会导致模型难以收敛；过低会可能导致模型收敛速度过慢，建议根据经验设定合理值
WeightDecay	分位L1和L2正则。L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
WeightDecayCoeff	分位L1和L2正则。L1或L2正则：L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
Optimizer	优化器是在梯度下降中影响参数更新方式的选项，有Momentum，RMSProp，Adam等。
Momentum	参数更新的计算公式中的动量因子
LRDecayType	学习率衰减策略，随着训练迭代数的增加，学习率慢慢减小，以帮助模型收敛，常见的策略有阶梯式下降（PiecewiseDecay），余弦下降（CosineDecay）等。
UseWarmUp	是否使用学习率热身策略。
WarmUpSteps	学习率热身迭代数，通过该迭代次数后，学习率慢慢增加到指定值。

### 实例分割

#### 网络参数

参数	说明
KeepTopK	每个图像可以保留的总边界框数。
ScoreThreshold	过滤掉低置信度分数的边界框的阈值。
NmsThreshold	对检测框进行非极大值抑制(NMS)时使用的IOU阈值。
Anchors	基于锚框的检测框架里对预置锚框(anchor)大小的设置。

#### 预处理参数

参数	说明
InputSize	输入图像分辨率，值越大训练时显存占用越大。
BatchSize	一次训练所选取的样本数，值越大训练时显存占用越大。

#### 训练参数

参数	说明
Epochs	训练集完整参与训练的次数，迭代轮数越多训练耗时越长。
EvalInterval	训练中评估的次数，评估间隔越小评估越频繁，训练耗时越长。
ModelSaveInterval	训练中模型保存间隔，间隔越小训练过程中保存的模型越多。
LogInterval	训练日志输出间隔，间隔越小训练日志内容越多。

#### 优化器参数

参数	说明
BaseLearningRate	单BatchSize对应的学习率，会根据BatchSize和卡数线性增大。学习率是在梯度下降的过程中更新权重时的超参数。学习率过高会导致模型难以收敛；过低会可能导致模型收敛速度过慢，建议根据经验设定合理值
WeightDecay	分位L1和L2正则。L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
WeightDecayCoeff	分位L1和L2正则。L1或L2正则：L1正则化系数，用于权重矩阵稀疏；L2正则化系数，用于防止模型对训练数据过拟合。但系数过大，可能导致欠拟合。
Optimizer	优化器是在梯度下降中影响参数更新方式的选项，有Momentum，RMSProp，Adam等。
Momentum	参数更新的计算公式中的动量因子
LRDecayType	学习率衰减策略，随着训练迭代数的增加，学习率慢慢减小，以帮助模型收敛，常见的策略有阶梯式下降（PiecewiseDecay），余弦下降（CosineDecay）等。
UseWarmUp	是否使用学习率热身策略。
WarmUpSteps	学习率热身迭代数，通过该迭代次数后，学习率慢慢增加到指定值。

### 训练完成后同步发布为模型

任务训练完成后可通过评估、校验验证任务效果，任务效果满足实际使用要求后发布为模型完成模型部署流程，如当前任务已经过多轮迭代且任务效果较有保证可勾选训练完成后同步发布为模型，并输入发布为的模型名称以及版本描述，训练成功后将会自动发布为模型

任务与模型一一对应，如当前任务已有版本发布为模型，则当前任务下的其他版本发布时仅支持发布在当前模型下

### 数据配置

#### 添加训练数据

- 先选择数据集，再按分类选择数据集里的图片，可从多个数据集选择图片
- 训练时间与数据量大小有关，数据量越大，训练耗时越长 **Tips**：
- 图像分类任务类型，如只有1个分类需要识别，或者实际业务场景所要识别的图片内容不可控，可以在训练前勾选“增加识别结果为[其他]的默认分类”。勾选后，模型会将与训练集无关的图片识别为“其他”
- 如果同一个分类的数据分散在不同的数据集里，可以在训练时同时从这些数据集里选择分类，模型训练时会合并分类名称相同的图片 **添加自定义验证集**

AI模型在训练时，每训练一批数据会进行模型效果检验，以某一张验证图片作为验证数据，通过验证结果反馈去调节训练。可以简单地把AI模型训练理解为学生学习，训练集则为每天的上课内容，验证集即为每周的课后作业，质量更高的每周课后作业能够更好的指导学生寻找自己的不足，从而提高成绩。同理AI模型训练的验证集也是这个功效。

注：学生的课后作业应该与上课内容对应，这样才能巩固知识。因此，验证集的标签也应与训练集完全一致。

#### 添加自定义测试集

如果学生的期末考试是平时的练习题，那么学生可能通过记忆去解题，而不是通过学习的方法去做题，所以期末考试的试题应与平时作业不能一样，才能检验学生的学习成果。那么同理，AI模型的效果测试不能使用训练数据进行测试，应使用训练数据集外的数据测试，这样才能真实的反映模型效果。

注：期末考试的内容属于学期的内容，但不一定需要完全包括所学内容。同理，测试集的标签是训练集的全集或者子集即可。

### 配置数据增强策略

深度学习模型的成功很大程度上要归功于大量的标注数据集。通常来说，通过增加数据的数量和多样性往往能提升模型的效果。当在实践中无法收集到数目庞大的高质量数据时，可以通过配置数据增强策略，对数据本身进行一定程度的扰动从而产生“新”数据。模型会通过学习大量

的"新"数据，提高泛化能力。

你可以在「默认配置」、「手动配置」2种方式中进行选择，完成数据增强策略的配置。

### 默认配置

如果你不需要特别配置数据增强策略，就可以选择默认配置。后台会根据你选择的算法，自动配置必要的数据增强策略。

### 手动配置

飞桨EasyDL提供了大量的数据增强算子供开发者手动配置。你可以通过下方的算子功能说明或训练页面的效果展示，来了解不同算子的功能：

算子名	功能
ShearX	剪切图像的水平边
ShearY	剪切图像的垂直边
TranslateX	按指定距离（像素点个数）水平移动图像
TranslateY	按指定距离（像素点个数）垂直移动图像
Rotate	按指定角度旋转图像
AutoContrast	自动优化图像对比度
Contrast	调整图像对比度
Invert	将图像转换为反色图像
Equalize	将图像转换为灰色值均匀分布的图像
Solarize	为图像中指定阈值之上的所有像素值取反
Posterize	减少每个颜色通道的bits至指定位数
Color	调整图像颜色平衡
Brightness	调整图像亮度
Sharpness	调整图像清晰度
Cutout	通过随机遮挡增加模型鲁棒性，可设定遮挡区域的长宽比例

### 效果展示



剪切图像的水平边，能更好地识别发生了水平方向变形的图像



关闭

### 训练环境配置

飞桨EasyDL支持以Intel CPU，或可通过NVIDIA旗下不同型号显卡加速训练。

如果您的计算机有NVIDIA® GPU，且需要使用GPU环境进行训练，请确保满足以下条件：

- Windows 7/8/10/11：需安装 CUDA 11.2 与 cuDNN v8.2.1
- Ubuntu 16.04/18.4/20.4：需 CUDA 11.2 与 cuDNN v8.1.1
- CentOS 7：需 CUDA 11.2 与 cuDNN v8.1.1

## CUDA、cuDNN安装指南

您可参考NVIDIA官方文档了解CUDA和cuDNN的安装流程和配置方法，详见：

- CUDA 安装指南：<https://docs.nvidia.com/cuda/>
- cuDNN安装指南：<https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html>

## 效果评估

训练完成后，可通过模型评估报告或模型校验了解模型效果。

- 模型评估报告：训练完成后，可以在【任务总览】列表中看到模型效果，以及详细的模型评估报告。
- 模型校验：训练完成后，可以在【任务总览】操作中点击「校验」，实时校验模型效果。

## 模型评估报告

### 图像分类

#### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、准确率、F1-score等。这部分模型效果的指标是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如图片数量低于100个），参与评估的数据可能不超过30个，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

查看模型评估结果时，需要思考在当前业务场景，更关注精确率与召回率哪个指标。是更希望减少误识别，还是更希望减少漏识别。前者更需要关注精确率的指标，后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况，对于希望准确率与召回率兼具的场景，F1-score越接近1效果越好。评估指标具体的说明如下。

- **F1-score**：对某类别而言为精确率和召回率的调和平均数，评估报告中指各类别F1-score的平均数
- **准确率**：基于随机测试集进行计算，为正确分类的样本数与总样本数之比

注意：若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口批量测试，获取更准确的模型效果。

#### 整体评估



- **top1-top5准确率** 对于每一个评估的图片文件，模型会给根据置信度高低，依次给出top1-top5的识别结果，其中top1置信度最高，top5的置信度最低。那么top1的准确率值是指对于评估标准为“top1结果识别为正确时，判定为正确”给出准确率。top2准确率值是指对于评估标准为“top1或者top2只要有一个命中正确的结果，即判定为正确”给出的准确率。……以此类推。

## 详细评估

这个部分支持查看模型识别错误的图片示例，以及使用混淆矩阵定位易混淆的分类。

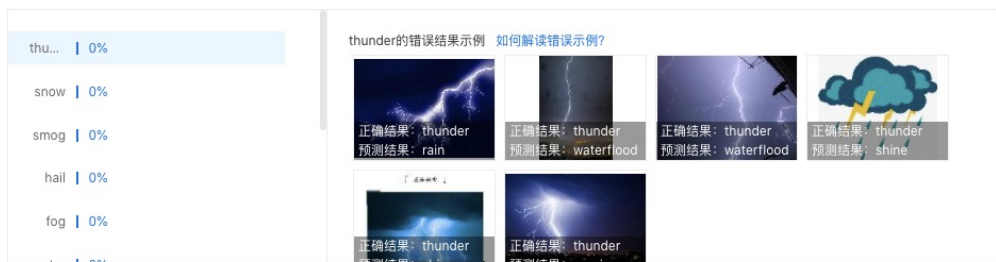
### 识别错误图片示例

通过分标签查看模型识别错误的图片，寻找其中的共性，进而有针对性的扩充训练数据。

详细评估

按分类查看错误示例

不同分类的F1-score及对应的识别错误的图片 (不包含训练时可能勾选的“其他”类识别错误的图片)



例如, 你训练了一个将小番茄和樱桃分类的模型。在查看小番茄分类的错误示例时, 发现错误示例中有好几张图片都是带着绿色根茎的小番茄 (与樱桃比较相似)。这种情况下, 就需要在小番茄分类的训练集中, 多增加一些带绿色根茎的图片, 让模型有足够的学习数据能够学习到带根茎的小番茄和樱桃的区别。

这个例子中, 我们找到的是识别错误的图片中, 目标特征上的共性。除此之外, 还可以观察识别错误的图片在以下维度是否有共性, 比如: 图片的拍摄设备、拍摄角度, 图片的亮度、背景等等。



物体检测

整体评估

在这个部分可以看到模型训练整体的情况说明, 包括基本结论、mAP、精确率、召回率。这部分模型效果的指标是基于训练数据集, 随机抽出部分数据不参与训练, 仅参与模型效果评估计算得来。所以当数据量较少时 (如图片数量低于100个), 参与评估的数据可能不超过30个, 这样得出的模型评估报告效果仅供参考, 无法完全准确体现模型效果。

注意: 若想要更充分了解模型效果情况, 建议发布模型为API后, 通过调用接口批量测试, 获取更准确的模型效果。

整体评估

cat\_dog V2效果优异, 建议针对识别错误的图片示例继续优化模型效果。由于目前训练集数据量较少, 该结论仅供参考, 建议扩充训练集得到更准确的评估效果。 [如何优化效果?](#)



查看模型评估结果时, 需要思考在当前业务场景, 更关注精确率与召回率哪个指标。是更希望减少误识别, 还是更希望减少漏识别。前者更需要关注精确率的指标, 后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况, 对于希望准确率与召回率兼具的场景, F1-score越接近1效果越好。评估指标说明如下

- **F1-score**: 对某类别而言为精确率和召回率的调和平均数, 评估报告中指各类别F1-score的平均数
- **mAP**: mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务, 每一类object都可以计算出其精确率(Precision)和召回率(Recall), 在不同阈值下多次计算/试验, 每个类都可以得到一条P-R曲线, 曲线下的面积就是average
- **精确率**: 正确预测的物体数与预测物体总数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

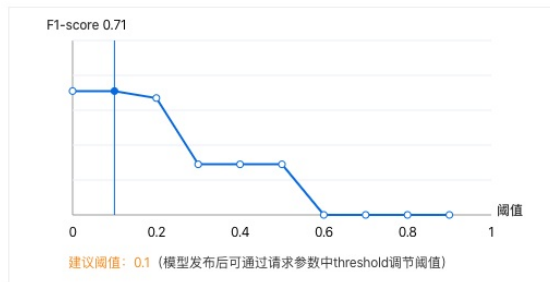
- **召回率**：正确预测的物体数与真实物体数之比。评估报告中具体指经比较F1-score最高的阈值下的结果

### 详细评估

在这个部分可以看到不同阈值下的F1-score、模型识别错误的图片示例，以及使用混淆矩阵定位易混淆的标签。

#### 详细评估

不同阈值下F1-score表现



不同标签的mAP及对应的识别错误的图片

cat	85%	cat的错误结果示例 (点击查看识别错误详情)
dog	90%	

### 识别错误图片示例

通过分标签查看模型识别错误的图片，直寻找其中的共性，进而有针对性的扩充训练数据；或发现是标注错误，从而直接点击修改标注来将标注修正

如下图所示，可以通过勾选「误识别」、「漏识别」来分别查看两种错误识别的情况：

错误详情 ×

原标注结果	模型识别结果

■ 正确识别    ■ 误识别    ■ 漏识别    
 [修改标注](#)    [如何解读错误示例?](#)    
  正确识别     误识别     漏识别

- **误识别**：红框内没有目标物体（准备训练数据时没有标注），但模型识别到了目标物体 观察误识别的目标有什么共性：例如，一个检测电动车的模型，把很多自行车误识别成了电动车（因为电动车和自行车外观上比较相似）。这时，就需要在训练集中为自行车特别建立一个标签，并且在所有训练集图片中，将自行车标注出来。

可以把模型想象成一个在认识世界的孩童，当你告诉他电动车和自行车分别是什么样时，他就能认出来；当你没有告诉他的时候，他就有可能把自行车认成电动车。

- **漏识别**：橙框内应该有目标物体（准备训练数据时标注了），但模型没能识别出目标物体 观察漏识别的目标有什么共性：例如，一个检测会议室参会人数的模型，会漏识别图片中出现的白色人种。这大概率是因为训练集中缺少白色人种的标注数据造成的。因此，需要在训练集中添加包含白色人种的图片，并将白色人种标注出来。

黄色人种和白色人种在外貌的差别上是比较明显的，由于几乎所有的训练数据都标注的是黄色人种，所以模型很可能认不出白色人种。需要增加白色人种的标注数据，让模型学习到黄色人种和白色人种都属于「参会人员」这个标签。

以上例子中，我们找到的是识别错误的图片中，目标特征上的共性。除此之外，还可以观察识别错误的图片在以下维度是否有共性，比如：图片的拍摄设备、拍摄角度，图片的亮度、背景等等。

## 实例分割

### 整体评估

在这个部分可以看到模型训练整体的情况说明，包括基本结论、mAP、精确率、召回率。这部分模型效果的指标是基于训练数据集，随机抽出部分数据不参与训练，仅参与模型效果评估计算得来。所以当数据量较少时（如图片数量低于100个），参与评估的数据可能不超过30个，这样得出的模型评估报告效果仅供参考，无法完全准确体现模型效果。

注意：若想要更充分了解模型效果情况，建议发布模型为API后，通过调用接口批量测试，获取更准确的模型效果。

#### 整体评估

fenge01 V14效果优异，建议针对识别错误的图片示例继续优化模型效果。由于目前训练集数据量较少，该结论仅供参考，建议扩充训练集得到更准确的评估效果。 [如何优化效果？](#)



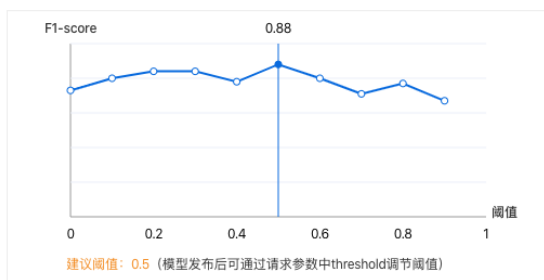
查看模型评估结果时，需要思考在当前业务场景，更关注精确率与召回率哪个指标。是更希望减少误识别，还是更希望减少漏识别。前者更需要关注精确率的指标，后者更需要关注召回率的指标。同时F1-score可以有效关注精确率和召回率的平衡情况，对于希望准确率与召回率兼具的场景，F1-score越接近1效果越好。评估指标说明如下

- **F1-score**：对某类别而言为精确率和召回率的调和平均数，评估报告中指各类别F1-score的平均数
- **mAP**：mAP(mean average precision)是物体检测(Object Detection)算法中衡量算法效果的指标。对于物体检测任务，每一类object都可以计算出其精确率(Precision)和召回率(Recall)，在不同阈值下多次计算/试验，每个类都可以得到一条P-R曲线，曲线下的面积就是average
- **精确率**：正确预测的物体数与预测物体总数之比。评估报告中具体指经比较F1-score最高的阈值下的结果
- **召回率**：正确预测的物体数与真实物体数之比。评估报告中具体指经比较F1-score最高的阈值下的结果 [详细评估](#)

在这个部分可以看到不同阈值下的F1-score，以及模型识别错误的图片示例。

#### 详细评估

不同阈值下F1-score表现



不同标签的mAP及对应的识别错误的图片



### 识别错误图片示例

通过分标签查看模型识别错误的图片，寻找其中的共性，进而有针对性的扩充训练数据。



如下图所示，可以通过勾选「误识别」、「漏识别」来分别查看两种错误识别的情况：



- **误识别**：红色遮盖内没有目标物体（准备训练数据时没有标注），但模型识别到了目标物体 观察误识别的目标有什么共性：例如，一个检测电动车的模型，把很多自行车误识别成了电动车（因为电动车和自行车外观上比较相似）。这时，就需要在训练集中为自行车特别建立一个标签，并且在所有训练集图片中，将自行车标注出来。

可以把模型想象成一个在认识世界的孩童，当你告诉他电动车和自行车分别是什么样时，他就能认出来；当你没有告诉他时，他就有可能把自行车认成电动车。

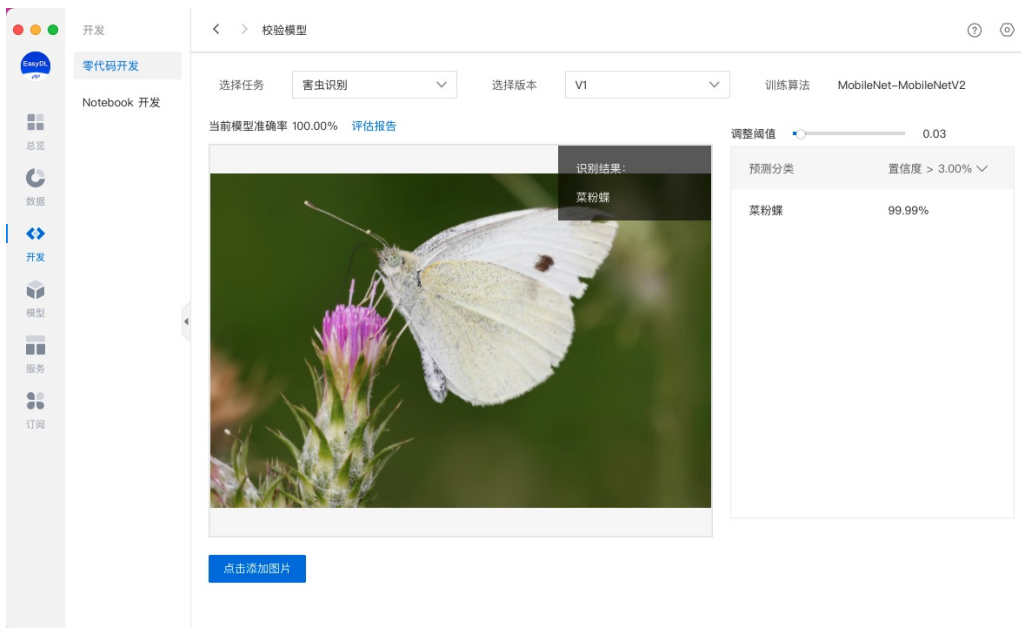
- **漏识别**：橙色遮盖内应该有目标物体（准备训练数据时标注了），但模型没能识别出目标物体 观察漏识别的目标有什么共性：例如，一个检测会议室参会人数的模型，会漏识别图片中出现的白色人种。这大概率是因为训练集中缺少白色人种的标注数据造成的。因此，需要在训练集中添加包含白色人种的图片，并将白色人种标注出来。

黄色人种和白色人种在外貌的差别上是比较明显的，由于几乎所有的训练数据都标注的是黄色人种，所以模型很可能认不出白色人种。需要增加白色人种的标注数据，让模型学习到黄色人种和白色人种都属于「参会人员」这个标签。

以上例子中，我们找到的是识别错误的图片中，目标特征上的共性。除此之外，还可以观察识别错误的图片在以下维度是否有共性，比如：图片的拍摄设备、拍摄角度，图片的亮度、背景等等。

### 模型校验

#### 图像分类



#### 物体检测



当前模型mAP平均精度 89.13% [评估报告](#)




识别结果 [如何优化效果?](#)

调整阈值  当前阈值: 0.3

预测标签	置信度>30%
1. person	98.41%
2. positive	94.63%

### 实例分割

当前模型mAP平均精度 100.00% [评估报告](#)



识别结果 [如何优化效果?](#)

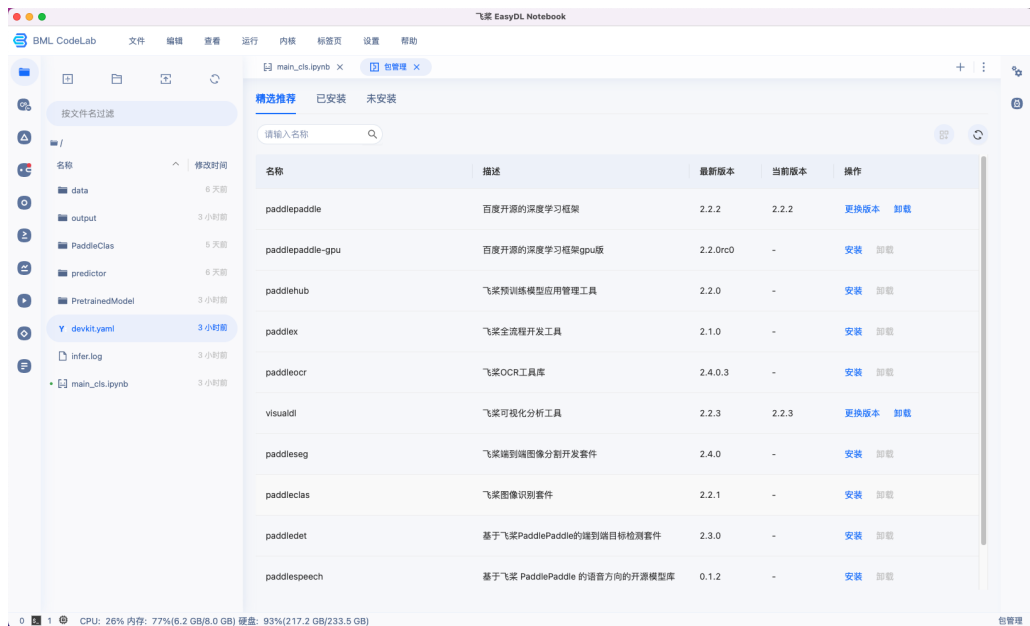
调整阈值  当前阈值: 0.3

预测标签	置信度>30%
1. head	79.28%

### Notebook开发

#### 🔗 Notebook简介

集成了包括PaddleX、PaddleDetection、PaddleSeg、PaddleClas等端到端开发套件的轻量级IDE，用户可在该模块内进行代码编辑、调试等开发工作，快速高效的完成各类任务的实现，可对预置模型调参中的模型进行代码级优化。



### 预置开发套件

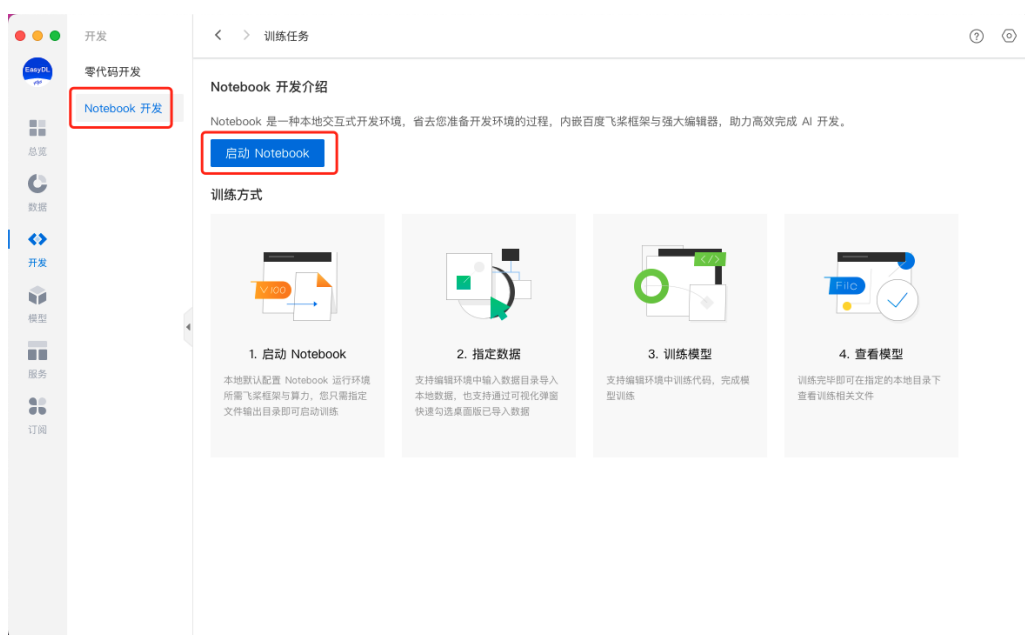
名称	说明
PaddlePaddle	百度开源的深度学习框架
PaddlePaddle (GPU)	百度开源的深度学习框架GPU版
PaddleX	飞桨全流程开发工具
PaddleClas	基于飞桨PaddlePaddle的端到端图像识别开发套件
PaddleDetection	基于飞桨PaddlePaddle的端到端目标检测开发套件
PaddleSeg	基于飞桨PaddlePaddle的端到端图像分割开发套件
PaddleOCR	飞桨OCR工具库
PaddleSpeech	基于飞桨 PaddlePaddle语音方向的开源模型库
PaddleHub	飞桨预训练模型应用管理工具
PaddleSlim	专注于深度学习模型压缩的工具库，提供剪裁、量化、蒸馏、和模型结构搜索等模型压缩策略，帮助用户快速实现模型的小型化
Visualdl	飞桨可视化分析工具

## 🔗 Notebook建模示例

本文以物体检测任务类型为例，从启动Notebook到引入数据、训练模型，再到保存模型的全流程。

### 启动Notebook

step1：在左侧导航栏中选择开发->Notebook开发



step2：选择开发语言、AI框架、资源规格与工作目录后启动Notebook



## 训练物体检测模型

### 下载 PaddleDetection 套件

打开进入 Notebook，点击进入终端，本文以 PaddleDetection 代码库 release/2.3 分支为例，输入如下命令克隆 PaddleDetection 代码库并切换至 release/2.3 分支。整个过程需要数十秒，请耐心等待。

```
##### gitee 国内下载比较快
git clone https://gitee.com/paddlepaddle/PaddleDetection.git -b release/2.3
##### github
##### git clone https://github.com/PaddlePaddle/PaddleDetection.git -b release/2.3
```

### 安装环境

在终端环境中，安装该版本的 PaddleDetection 代码包依赖的 paddlepaddle-gpu，执行如下命令：

```
python -m pip install paddlepaddle-gpu==2.1.3.post101 -f https://www.paddlepaddle.org.cn/whl/linux/mkl/avx/stable.html
```

安装完成后，使用 python 或 python3 进入 python 解释器，输入 import paddle，再输入 paddle.utils.run\_check() 如果出现 PaddlePaddle is installed successfully!，说明成功安装。**准备训练数据**

训练数据是模型生产的重要条件，优质的数据集可以很大程度上的提升模型训练效果，准备数据可以参考[链接](#)。本文所用的安全帽检测数据集可前往此链接进行下载：[下载链接](#)。

#### step1：导入用户数据

目前在 Notebook 中不能直接访问您在 飞桨 EasyDL 中创建的数据集，需要通过在终端输入数据所在路径。

#### step2：数据转换

PaddleDetection 训练所需要的数据格式与 飞桨 EasyDL 默认的数据格式有所不同，所以需要利用脚本将导入的数据转为 PaddleDetection 支持的数据格式，并进行 3:7 切分。

PaddleDetection 默认支持的标注格式为 COCO 格式，转换脚本如下：

```
import os
import cv2
import json
import glob
import codecs
import random
from pycocotools.coco import COCO

def parse_bml_json(json_file):
    """
    解析标注文件
    :return:
    """
    annos = json.loads(codecs.open(json_file).read())
```

```

labels = annos['labels']
bboxes = []
for label in labels:
    x1 = label["x1"]
    y1 = label["y1"]
    x2 = label["x2"]
    y2 = label["y2"]
    id = label["name"]
    bboxes.append([x1, y1, x2, y2, id])
return bboxes

```

```

def bbox_transform(box):
    """
    x1, y1, x2, y2 转为 x1, y1, width, height
    :return
    """
    box = list(map(lambda x: float(x), box))
    box[2] = box[2] - box[0]
    box[3] = box[3] - box[1]
    return box

```

```

def parse_label_list(src_data_dir, save_dir):
    """
    遍历标注文件，获取label_list
    :return:
    """
    label_list = []
    anno_files = glob.glob(src_data_dir + "*.json")
    for anno_f in anno_files:
        annos = json.loads(codecs.open(anno_f).read())
        for lb in annos["labels"]:
            label_list.append(lb["name"])
    label_list = list(set(label_list))
    with codecs.open(os.path.join(save_dir, "label_list.txt"), 'w', encoding="utf-8") as f:
        for id, label in enumerate(label_list):
            f.writelines("%s:%s\n" % (id, label))
    return len(label_list), label_list

```

```

def bml2coco(src_dir, coco_json_file):
    """
    将标注格式转为COCO标注格式
    :return:
    """
    coco_images = []
    coco_annotations = []

    image_id = 0
    anno_id = 0
    image_list = glob.glob(src_dir + ".*[JpPbB][PpNnMm]*")
    for image_file in image_list:
        anno_f = image_file.split(".")[0] + ".json"
        if not os.path.isfile(anno_f):
            continue
        bboxes = parse_bml_json(anno_f)
        im = cv2.imread(image_file)
        h, w, _ = im.shape
        image_i = {"file_name": os.path.basename(image_file), "id": image_id, "width": w, "height": h}
        coco_images.append(image_i)

```

```

    for id, bbox in enumerate(bboxes):
        # bbox : [x1, y1, x2, y2, label_name]
        anno_i = {"image_id": image_id, "bbox": bbox_transform(bbox[:4]), 'category_id': label_list.index(bbox[4]),
                  'id': anno_id, 'area': 1.1, 'iscrowd': 0, "segmentation": None}
        anno_id += 1
        coco_annotations.append(anno_i)

    image_id += 1

coco_categories = [{"id": id, "name": label_name} for id, label_name in enumerate(label_list)]
coco_dict = {"info": "info", "licenses": "BMLCloud", "images": coco_images, "annotations": coco_annotations,
            "categories": coco_categories}
with open(coco_json_file, 'w', encoding="utf-8") as fin:
    json.dump(coco_dict, fin, ensure_ascii=False)

def split_det_origin_dataset(
    origin_file_path,
    train_file_path,
    eval_file_path,
    ratio=0.7):
    """
    按比例切分物体检测原始数据集
    :return:
    """
    coco = COCO(origin_file_path)
    img_ids = coco.getImgIds()
    items_num = len(img_ids)
    train_indexes, eval_indexes = random_split_indexes(items_num, ratio)
    train_items = [img_ids[i] for i in train_indexes]
    eval_items = [img_ids[i] for i in eval_indexes]

    dump_det_dataset(coco, train_items, train_file_path)
    dump_det_dataset(coco, eval_items, eval_file_path)

    return items_num, len(train_items), len(eval_items)

def random_split_indexes(items_num, ratio=0.7):
    """
    按比例分割整个list的index
    :return:分割后的两个index子列表
    """
    offset = round(items_num * ratio)
    full_indexes = list(range(items_num))
    random.shuffle(full_indexes)
    sub_indexes_1 = full_indexes[:offset]
    sub_indexes_2 = full_indexes[offset:]

    return sub_indexes_1, sub_indexes_2

def dump_det_dataset(coco, img_id_list, save_file_path):
    """
    物体检测数据集保存
    :return:
    """
    imgs = coco.loadImgs(img_id_list)
    img_anno_ids = coco.getAnnIds(imgIds=img_id_list, iscrowd=0)
    instances = coco.loadAnns(img_anno_ids)
    cat_ids = coco.getCatIds()

```

```
categories = coco.loadCats(cat_ids)
common_dict = {
    "info": coco.dataset["info"],
    "licenses": coco.dataset["licenses"],
    "categories": categories
}
img_dict = {
    "image_nums": len(imgs),
    "images": imgs,
    "annotations": instances
}
img_dict.update(common_dict)

json_file = open(save_file_path, 'w', encoding='UTF-8')
json.dump(img_dict, json_file)

class_nums, label_list = parse_label_list("/home/work/data/${dataset_id}/", "/home/work/PretrainedModel/")
bml2coco("/home/work/data/${dataset_id}/", "/home/work/PretrainedModel/org_data_list.json")
split_det_origin_dataset("/home/work/PretrainedModel/org_data_list.json", "/home/work/PretrainedModel/train_data_list.json",
"/home/work/PretrainedModel/eval_data_list.json")
```

将上述脚本存放为 `convert.py` 代码脚本，并将脚本最后两行的 `"/home/work/data/${dataset_id}/"` 均替换为所指定数据集路径，在终端中运行即可。运行代码。

```
python covert.py
```

注意：如果报错 `No module named 'pycocotools'`，需要通过如下命令安装相关依赖包，再运行 `covert.py` 代码。

```
pip install pycocotools
```

运行 `covert.py` 代码成功之后将在 `PretrainedModel/` 文件夹下生成对应的数据文件，包括 `label_list.txt`、`train_data_list.json`、`eval_data_list.json`、`org_data_list.json`。

## 训练模型

开发者准备好训练数据和安装环境之后即可开始训练物体检测模型。

**step1：在终端中打开 `PaddleDetection` 目录**

```
cd /PaddleDetection
```

**step2：修改yaml配置文件**

在 `PaddleDetection 2.0` 后续版本，采用了模块解耦设计，用户可以组合配置模块实现检测器，并可自由修改覆盖各模块配置，本文以 `configs/yolov3/yolov3_darknet53_270e_coco.yml` 为例：

```
yolov3_darknet53_270e_coco.yml 主配置入口文件
coco_detection.yml 主要说明了训练数据和验证数据的路径
runtime.yml 主要说明了公共的运行参数，比如说是否使用GPU、每多少个epoch存储checkpoint等
optimizer_270e.yml 主要说明了学习率和优化器的配置。
yolov3_darknet53.yml 主要说明模型、和主干网络的情况。
yolov3_reader.yml 主要说明数据读取器配置，如batch size，并发加载子进程数等，同时包含读取后预处理操作，如resize、数据增强等等
```

需要修改/覆盖的参数均可写在主配置入口文件中，主要修改点为训练、验证数据集路径、运行epoch数、学习率等，修改后的主配置文件如下（注释行即为需要修改的点）：

```
_BASE_: [  
  './datasets/coco_detection.yml',  
  './runtime.yml',  
  '_base_/optimizer_270e.yml',  
  '_base_/yolov3_darknet53.yml',  
  '_base_/yolov3_reader.yml',  
]  
  
snapshot_epoch: 5  
weights: output/yolov3_darknet53_270e_coco/model_final  
  
##### 预训练权重地址  
pretrain_weights: https://paddledet.bj.bcebos.com/models/yolov3_darknet53_270e_coco.pdparams  
  
##### coco_detection.yml  
num_classes: 2 #实际类别数  
TrainDataset:  
!COCODataset  
  image_dir: data/${dataset_id}/ # 图片地址  
  anno_path: PretrainedModel/train_data_list.json # 标注文件  
  dataset_dir: /home/work/ # 数据集根目录  
  data_fields: ['image', 'gt_bbox', 'gt_class', 'is_crowd']  
  
EvalDataset:  
!COCODataset  
  image_dir: data/${dataset_id}/ # 图片地址  
  anno_path: PretrainedModel/eval_data_list.json # 标注文件  
  dataset_dir: /home/work/ # 数据集根目录  
  
##### optimizer_270e.yml  
epoch: 50 # 迭代轮数  
LearningRate:  
  base_lr: 0.0001 # 学习率  
  schedulers:  
  - !PiecewiseDecay  
    gamma: 0.1  
    milestones:  
    - 30  
    - 45  
  - !LinearWarmup  
    start_factor: 0.  
    steps: 400
```

### step3 : 训练模型

在终端中执行以下命令，开始模型训练。

```
cd /PaddleDetection/  
python tools/train.py -c configs/yolov3/yolov3_darknet53_270e_coco.yml --eval
```

注意：如果报错 No module named 'lap' 和 No module named 'motmetrics'，则需要通过如下命令安装相关依赖包，再运行 `conversion.py` 代码。（如果缺失其他模块，也可用类似命令下载安装）

```
pip install lap motmetrics
```

### step4 : 模型评估

在终端中执行以下命令，开始模型评估。

```
python tools/eval.py -c configs/yolov3/yolov3_darknet53_270e_coco.yml \  
  -o weights=output/yolov3_darknet53_270e_coco/model_final
```

运行完成输出如下结果：

```

Average Precision (AP) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.279
Average Precision (AP) @[ IoU=0.50 | area= all | maxDets=100 ] = 0.534
Average Precision (AP) @[ IoU=0.75 | area= all | maxDets=100 ] = 0.251
Average Precision (AP) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.298
Average Precision (AP) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Precision (AP) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 1 ] = 0.086
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets= 10 ] = 0.331
Average Recall (AR) @[ IoU=0.50:0.95 | area= all | maxDets=100 ] = 0.351
Average Recall (AR) @[ IoU=0.50:0.95 | area= small | maxDets=100 ] = 0.351
Average Recall (AR) @[ IoU=0.50:0.95 | area=medium | maxDets=100 ] = -1.000
Average Recall (AR) @[ IoU=0.50:0.95 | area= large | maxDets=100 ] = -1.000
[01/30 02:28:27] ppdet.engine INFO: Total sample number: 150, averge FPS: 25.36864530446337

```

### step5: 模型预测

在终端中执行以下命令，开始模型预测（注意修改图片路径）。

```

python tools/infer.py -c configs/yolov3/yolov3_darknet53_270e_coco.yml \
    --infer_img=/home/work/data/${task_id}/xxx.jpeg \
    --output_dir=infer_output/ \
    --draw_threshold=0.5 \
    -o weights=output/yolov3_darknet53_270e_coco/model_final

```

### step6: 导出模型

在终端中执行以下命令，将最佳模型转为可以用于发布的 inference 模型

```

python tools/export_model.py -c configs/yolov3/yolov3_darknet53_270e_coco.yml \
    --output_dir=/home/work/PretrainedModel/ \
    -o weights=output/yolov3_darknet53_270e_coco/model_final

```

在终端中执行以下命令，将导出模型移至 /PretrainedModel/ 目录。

```

mv /PretrainedModel/yolov3_darknet53_270e_coco/* /home/work/PretrainedModel/

```

### Http服务运行端注意事项

模型部署启动服务时需对以下文件进行修改：

1. demo\_servering.py，启动服务设备为cpu/gpu

```

try:
    _model_dir = sys.argv[1]
except Exception as e:
    print("Usage: python3 demo_serving.py {model_dir} {host} {port}")
    exit(-1)
# python3 demo_serving.py C:\\predictor\\RES\\cls_output 设备IP地址 端口号
def test():
    """
    http serving
    :return:
    """
    arg_num = len(sys.argv)
    host = "0.0.0.0"
    port = "24401"
    if arg_num >= 3:
        host = sys.argv[2]
    if arg_num >= 4:
        port = sys.argv[3]

    server = Serving(model_dir=_model_dir)
    server.run(host=host, port=port, use_gpu=False)

```

模型文件路径 设备IP地址 端口号任意如8821

是否开启gpu推理服务

2. demo\_client.py，发送请求设备任意，只有保证ip和端口号与起服务设备一致即可

```

"""
def http_client_test():
    import requests
    url = 'http://0.0.0.0:24401/' #ip地址和端口号根据启动的serving打印信息填写

    try:
        # 图像预测示例
        import cv2
        img = cv2.imread("C:\\predictor\\tipper_truck_s_000086.png")
        ret, buffer = cv2.imencode('.png', img)
        data = buffer.tobytes()

```

起服务设备ip+端口号

测试图片

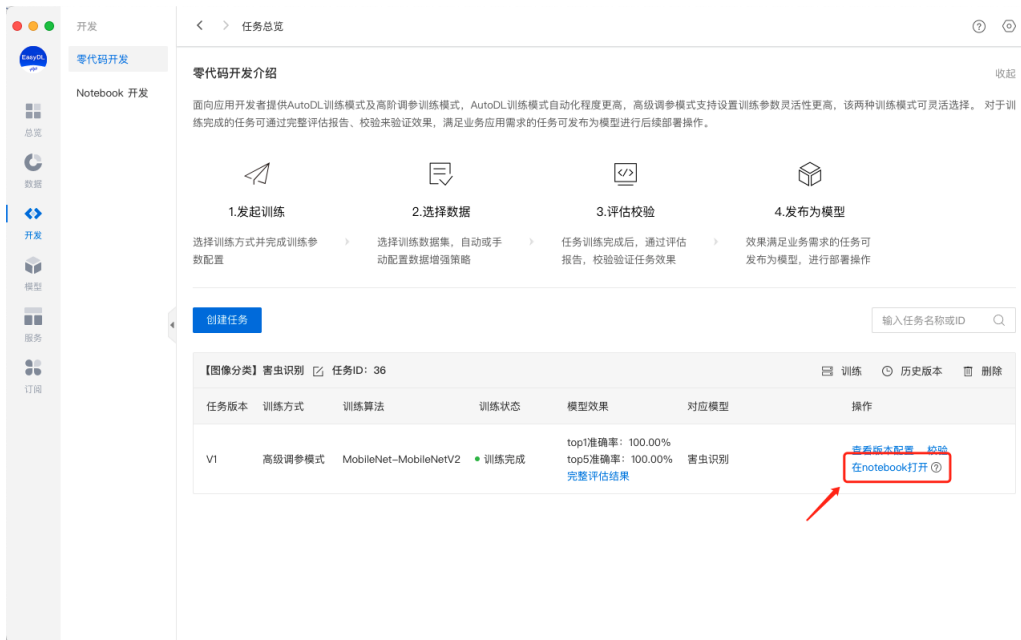


## 零代码开发转Notebook开发

飞桨EasyDL支持将通过零代码开发-预置模型调参模式开发的模型转为对应模型文件在Notebook中打开进行优化。

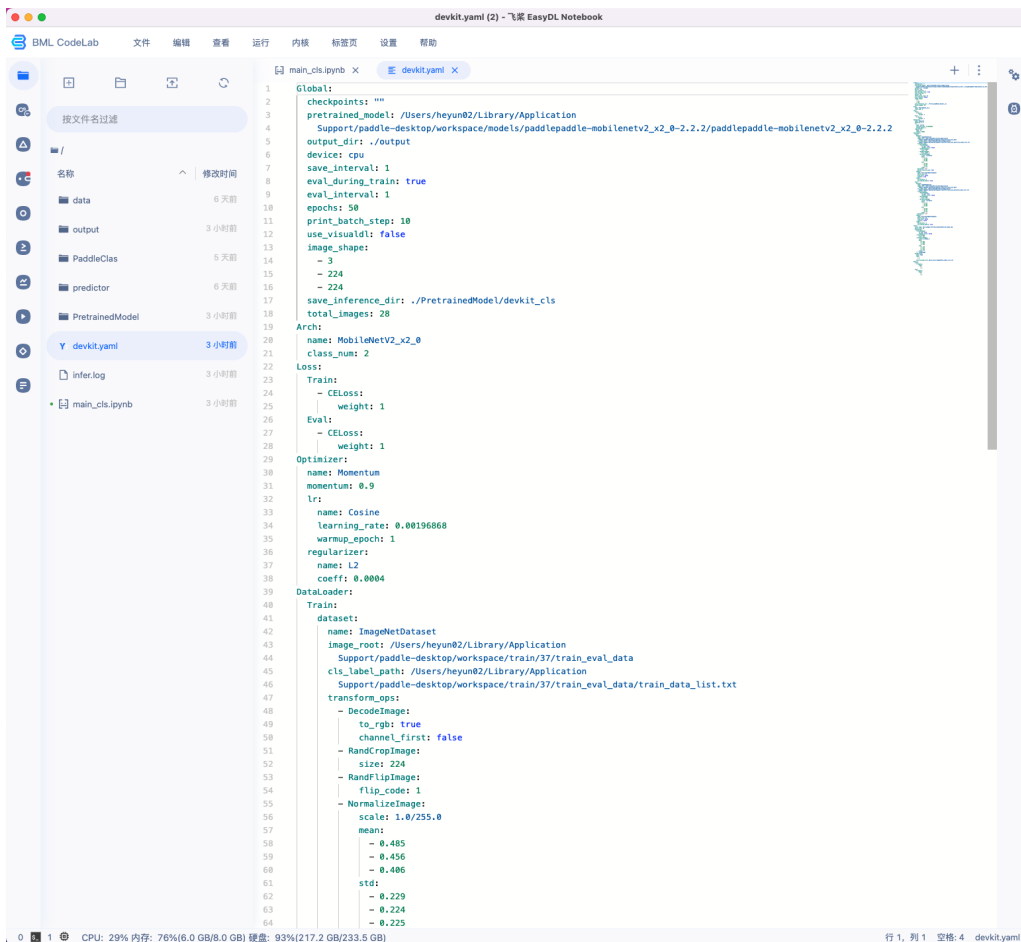
### step1：启动Notebook

通过预置模型调整模式完成训练后，在任务总览对应任务中点击【打开Notebook】。



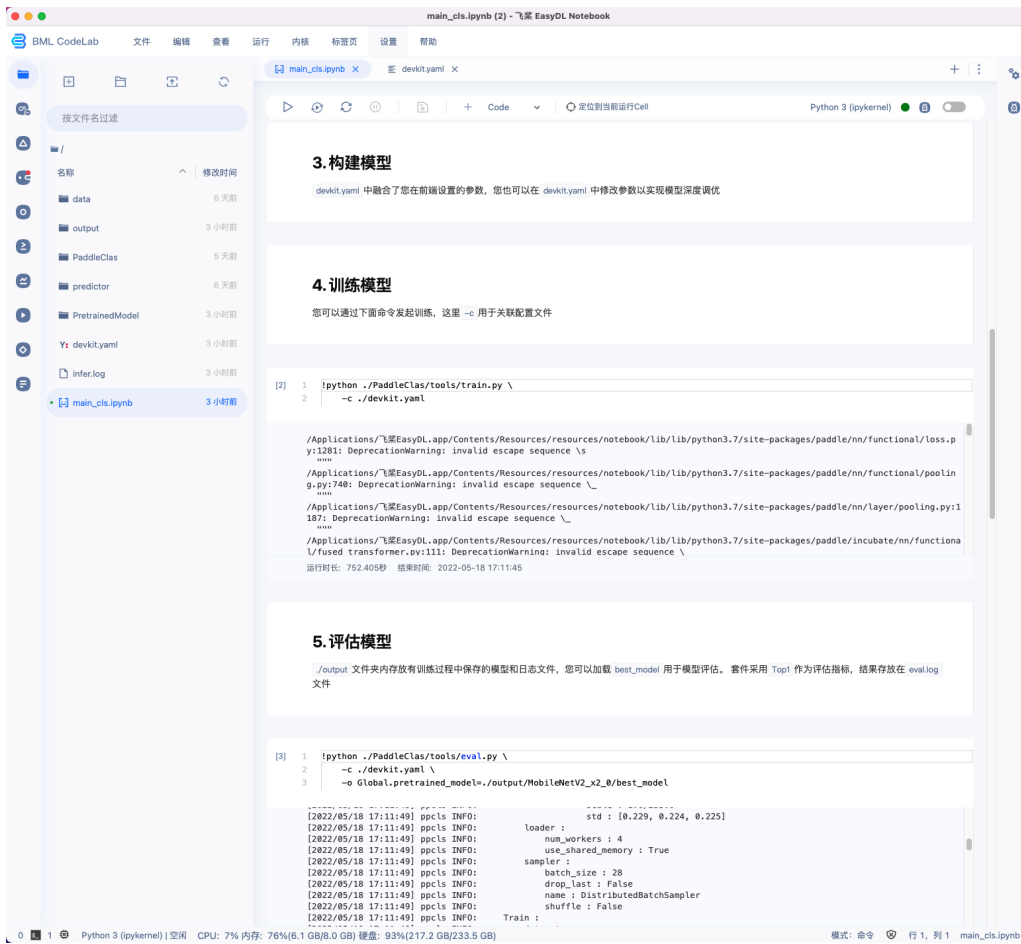
### step2：调整权重参数与网络结构

结合实际需求，在Yaml文件中对网络参数及结构进行调整。



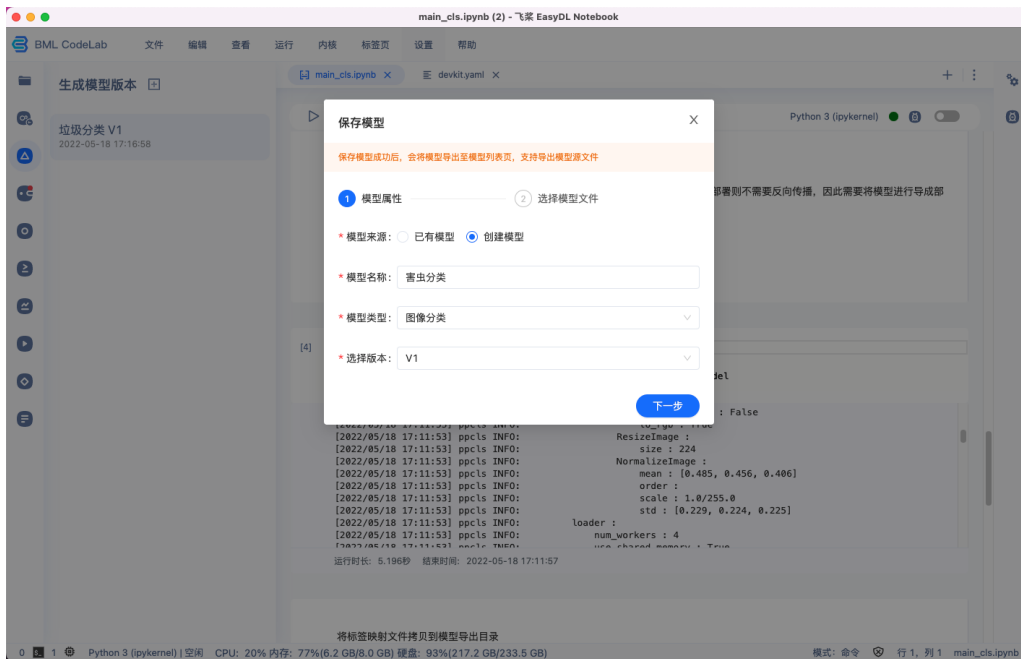
### step3：运行模型文件

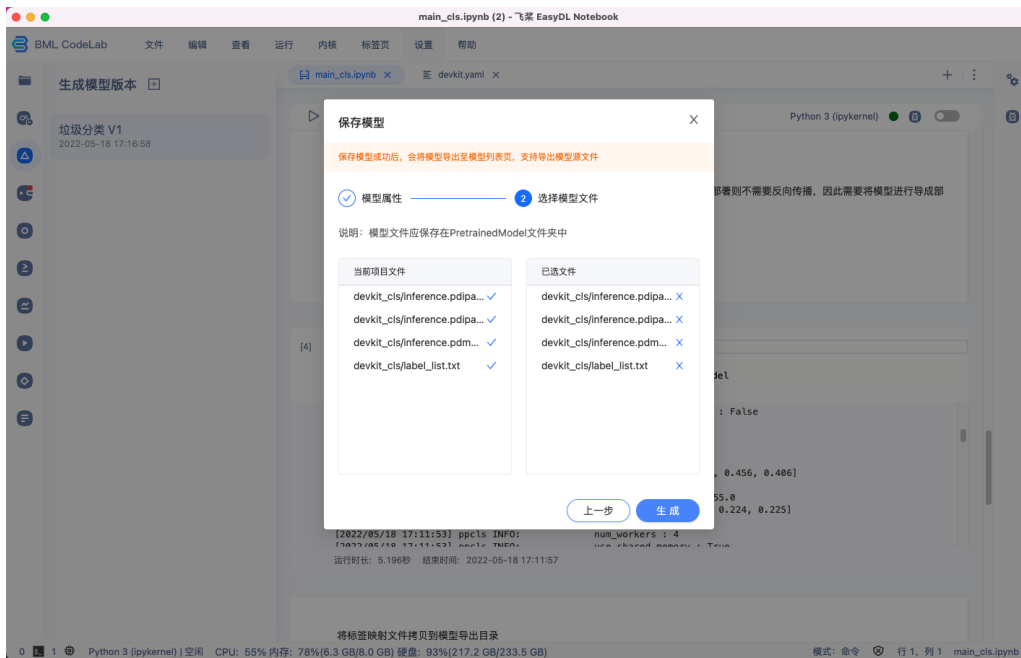
完成网络参数及结构调整后，运行对应main\_xxx.ipynb文件，即可完成环境配置、数据集准备、构建模型、训练模型、评估模型、模型部署等流程。



step4 : 生成模型版本

模型训练成功后, 点击左侧菜单栏【生产模型版本】, 将训练成功的模型发布至模型中心。





## 模型中心

### 模型列表

模型列表提供模型管理能力，可查看所有已发布为模型的任务

模型列表

模型部署

全部模型类型

模型名称	模型ID	模型来源	模型类型	版本数量	创建时间	操作
模型测试1	3	零代码开发	图像分类	1	2021-11-24 04:59:07	<a href="#">查看</a> <a href="#">删除</a>
IEC测试	1	零代码开发	图像分类	1	2021-11-16 11:03:24	<a href="#">查看</a> <a href="#">删除</a>

### 查看模型

可通过下拉框筛选需要查看的任务类型



点击【查看】可查看模型详情，在模型详情页可查看当前模型下的全部版本、模型版本对应的任务、训练方式、版本描述以及导入时间

点击对应任务可跳转至当前模型对应的任务版本

模型名称: 模型列表演示      模型ID: 4

模型类型: 图像分类      模型来源: 零代码开发

版本	对应任务	训练方式	描述	导入时间	操作
V2	<a href="#">7-V2</a>	高级调参模式	这里是版本描述	2021-11-24 02:44:56	<a href="#">导出模型文件</a> <a href="#">部署</a>
V1	<a href="#">6-V1</a>	AutoDL模式		2021-11-22 01:07:31	<a href="#">导出模型文件</a>

如需将模型投入实际应用可选择导出模型源文件或将模型部署为离线SDK

仅有训练时导出类型选择为【导出源文件与离线SDK】的任务支持部署为离线SDK

版本	对应任务	训练方式	描述	导入时间	操作
V2	<a href="#">7-V2</a>	高级调参模式	这里是版本描述	2021-11-24 02:44:56	<a href="#">导出模型文件</a> <a href="#">部署</a>
V1	<a href="#">6-V1</a>	AutoDL模式		2021-11-22 01:07:31	<a href="#">导出模型文件</a>

### 模型部署

服务发布界面可选择将模型发布为离线SDK，发布包含以下流程

#### 选择模型及版本

选择需要发布的模型及版本，选择完成后当前版本对应的训练方式及训练算法将自动展示

选择模型

模型版本

训练方式 零代码开发-高级调参模式

训练算法 ResNet-ResNet18\_vd

#### 选择部署环境

部署环境分为服务器、通用小型设备、专项适配硬件，不同类别下对应不同的操作系统及芯片，选择模型在实际应用中部署的系统及芯片点击发布即可完成模型发布

模型发布分为本地发布与云端发布两种方式，由于模型发布过程对发布环境的操作系统等环境有依赖，因此部分操作系统及芯片类型不支持在本地发布，飞桨EasyDL为您提供云端环境完成模型发布过程，发布过程请确保本地设备网络链接

选择部署环境

选择系统和芯片  本地发布  云端发布

Linux

选择系统和芯片不能为空

### 服务列表

服务列表主要用于管理已发布模型SDK，可查看全部已发布模型SDK。

模型SDK可在智能边缘控制台中可视化上完成部署流程，使用流程可参考[智能边缘控制台使用文档](#)。

模型名称	发布版本	应用平台	发布状态	发布方式	发布时间	操作
猫狗分类	1-V1	英伟达GPU-Windows	已发布	本地部署	2021-09-27 17:34	<a href="#">导出SDK</a>
害虫识别	5-V2	通用X86 CPU-Linux	已发布	本地部署	2021-11-22 20:51	<a href="#">导出SDK</a>
猫狗分类	10-V1	通用X86 CPU-Linux	已发布	本地部署	2021-12-07 20:33	<a href="#">导出SDK</a>
猫狗识别	11-V1	通用X86 CPU-Linux	已发布	本地部署	2021-12-23 16:12	<a href="#">导出SDK</a>

### 智能边缘控制台

EasyEdge Intelligent Edge Console（以下简称IEC）是EasyEdge推出的边缘设备管理的本地化方案。可以运行于多种架构、多系统、多类型的终端之上。通过IEC，用户可以方便地在本地进行

- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活，服务管理
- 接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关事件
- 硬件信息的可视化查看 操作系统支持：Linux、Windows

系统CPU架构支持：x86\_64、arm32、arm64

支持添加为服务的SDK版本包括：

- 通用CPU版（基础版/加速版）
- 通用ARM版
- NVIDIA-GPU版（基础版/加速版）

## 🔗 快速开始

根据本地设备类型下载智能边缘控制台

[Linux系统-amd64](#)：intel、AMD的64位CPU

[Linux系统-arm](#)：树莓派等32位arm CPU

[Linux系统-arm64](#)：rk3399、飞腾等 aarch64，64位的arm CPU

[Windows系统-amd64](#)：intel、AMD的64位CPU

根据您的设备的操作系统和硬件架构，将二进制和 etc 文件夹拷贝到任一目录，运行即可。（可以通过-cfg参数指定配置文件的路径，默认为 `./etc/easyedge-iec.yml`）

如果是Windows系统，双击运行 `easyedge-iec.exe` 即可 如果是Linux系统，终端输入 `./easyedge-iec` 即可运行

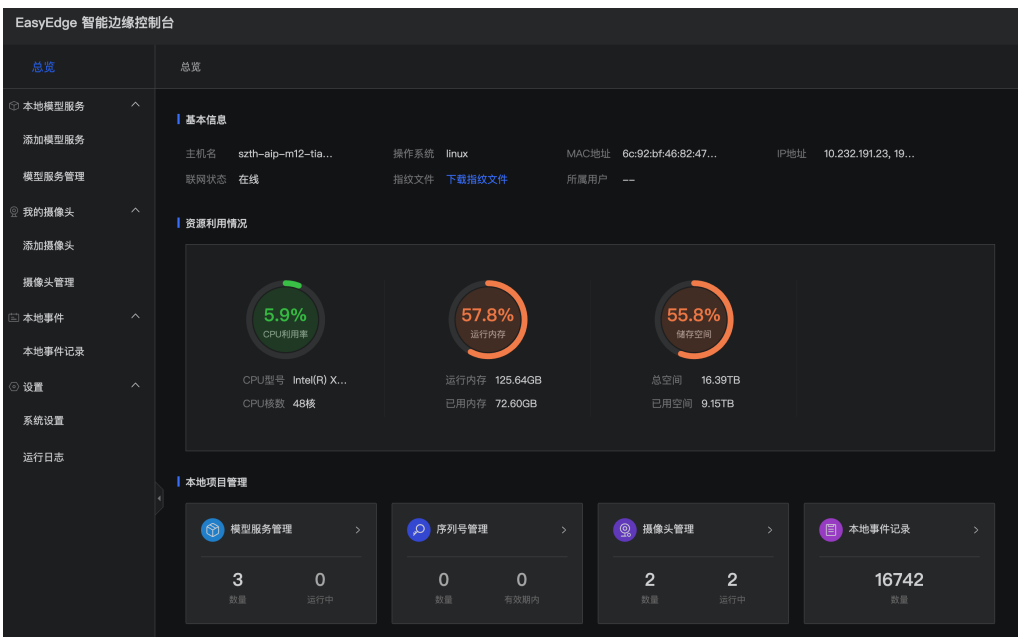
```
./easyedge-iec-linux-amd64
```

```
Loading cfg from ./etc/easyedge-iec.yml
2021-09-14T14:49:20 INFO [EasyEdgeIEC] Start stream service ...
2021-09-14T14:49:20 INFO [*EdgeStream] "2021-09-14 06:49:20,443 INFO [EasyEdge] 60439872 "
2021-09-14T14:49:20 WARN [*EdgeStream] "2021-09-14 06:49:20,443 WARNING [EasyEdge] 139803591698176 EdgeStream is
now serving at 127.0.0.1:24402"
2021-09-14T14:49:21 INFO [EasyEdgeIEC] HLS server disable....
2021-09-14T14:49:21 INFO [EasyEdgeIEC] HTTP-FLV listen On 0.0.0.0:8103
2021-09-14T14:49:21 WARN [EasyEdgeIEC] Baidu EasyEdge Intelligent Edge Console release 1.0.0, build 20210914
2021-09-14T14:49:21 INFO [EasyEdgeIEC] RTMP Listen On 0.0.0.0:1935
2021-09-14T14:49:21 INFO [EasyEdgeIEC] Webservice is now serving at 0.0.0.0:8702
```

启动之后，打开浏览器，访问 `http://{设备ip}:8702/easyedge/iec` 即可：



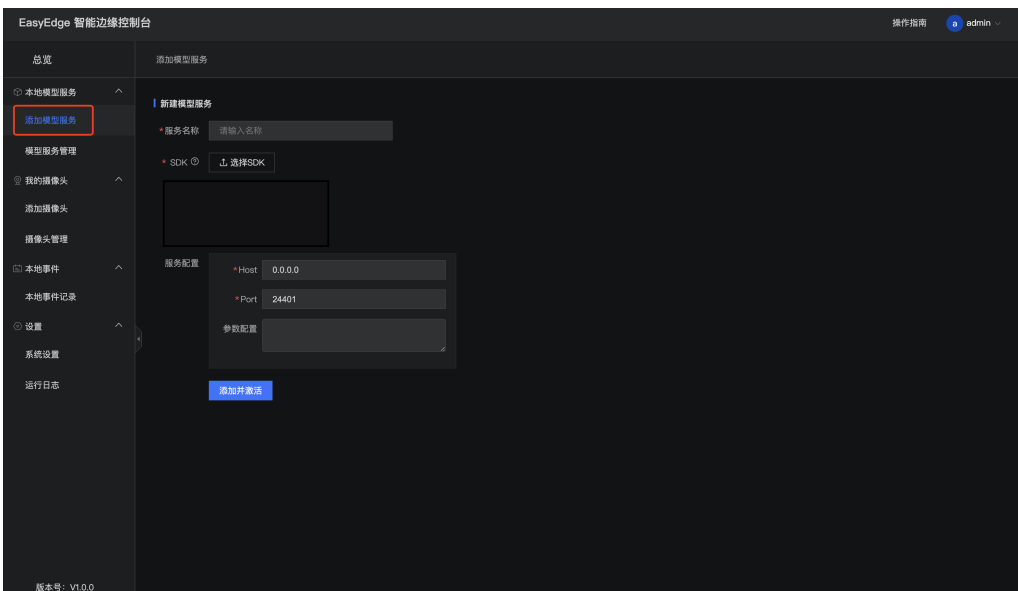
默认用户名密码为 admin / easyedge



### 功能使用说明

#### ① 添加模型服务

首先，点击导航栏的「本地模型服务」-「添加模型服务」。在页面中定义服务名称后，将已经下载好的Linux/Windows版本的SDK与IEC关联。



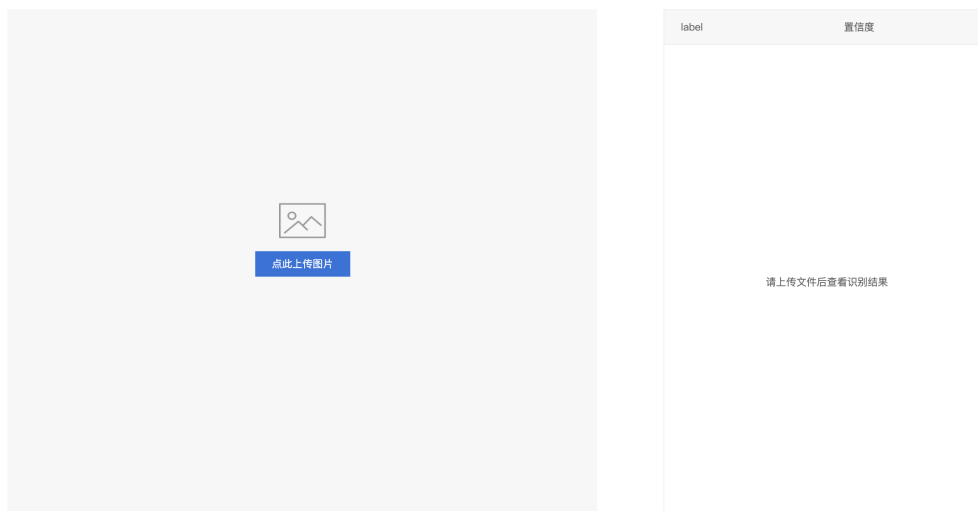
配置服务，在服务端口不冲突占用的情况下，使用默认即可，点击【添加并激活】

激活完成后即可在「模型服务管理」列表中启动服务，使用后续的操作栏功能。

### 体验本地demo

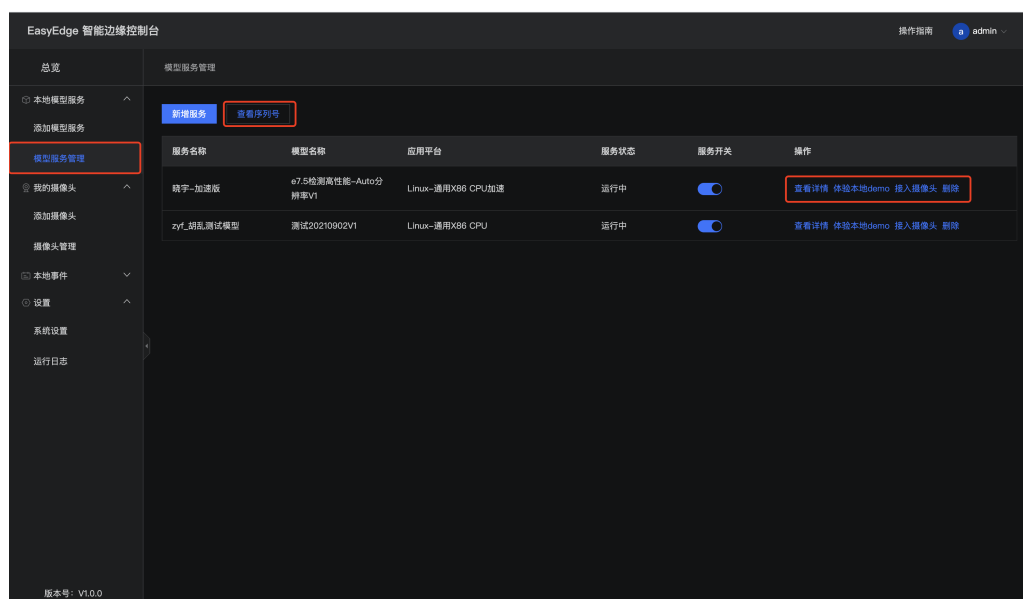
点击「本地demo体验」即可在立即上传图片进行预测

【物体检测】 97741 e7.5检测高性能-Auto分辨率V1



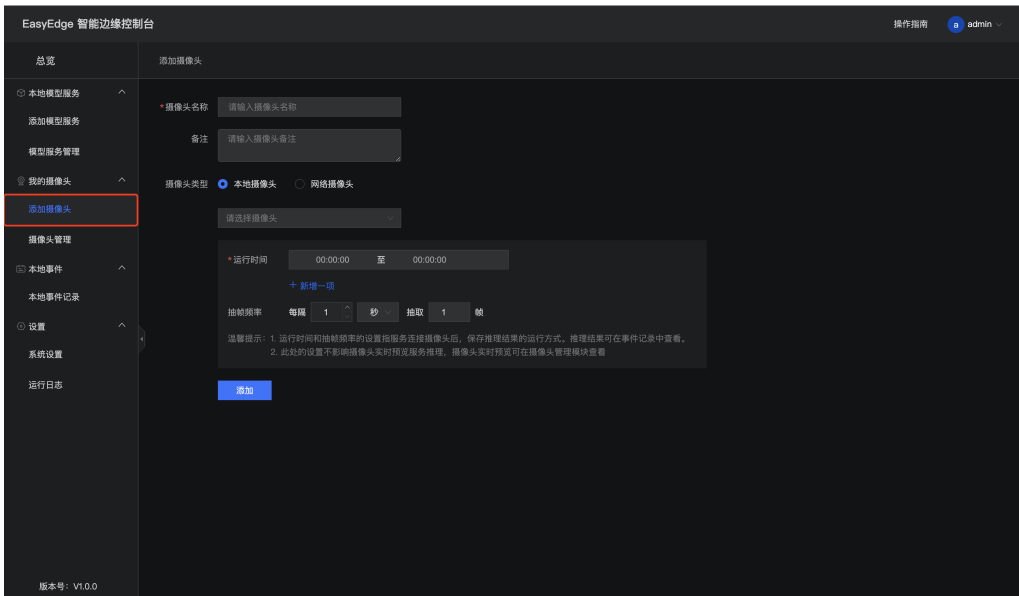
### 接入摄像头

使用接入摄像头功能首先需要添加摄像头，请参考第②步，完成后按照第③步操作



### ② 添加摄像头

导航栏点击「我的摄像头」-「添加摄像头」，定义摄像头名称、备注后即可添加摄像头。支持本地摄像头和网络摄像头。摄像头添加成功后即可设置摄像头的运行时间和频率



### ③ 摄像头接入模型服务预测

点击「本地模型服务」-「模型服务管理」中，所需接入预测的服务的「接入摄像头」



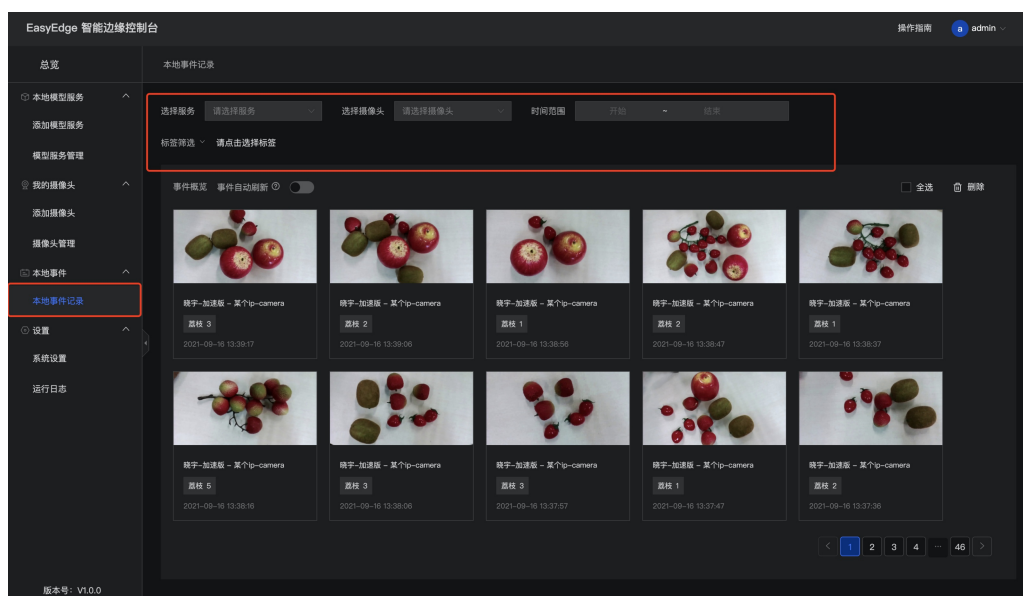
在弹出的弹窗中选择第②步中添加的摄像头，此时点击确认即可在「摄像头管理」中的实时预览功能中查看摄像头预测结果，识别结果默认不保存。如需保存识别结果，可设置对应的「本地事件触发条件」，根据标签和置信度，将识别结果保存至本地事件记录当中。设置多个标签条件时，IEC会以“或”的逻辑来将所有满足条件的识别结果保存



### ④ 本地事件

点击导航栏「本地事件记录」，可通过服务名称、摄像头名称、事件记录的时间、标签及置信度来筛选识别结果查看，多个标签及置信度同样也是“或”的逻辑记录。如有想要删除的事件数据可选择后删除，全选为本页全选。





## 配置项

配置文件 `etc/easyedge-iec.yml` 中有关于IEC的各项配置说明，一般无需修改，请确保理解配置项含义之后，再做修改。

```
##### IEC系统配置
com:
# 硬件利用率刷新时间间隔：过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# 事件监测触发扫描周期
eventTriggerIntervalSecond: 10
# IEC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: default
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: yes
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge）
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
toFile: yes
loggingFile: ./log/easyedge-iec.log

webservice:
# WEB服务的监听端口
listenPort: 8702
listenHost: 0.0.0.0

sdk:
# GPU SDK所使用的cuda版本：9 / 10 / 10.2 / 11.0 / 11.1。请安装完cuda之后，这设置正确的版本号。
cudaVersion: 10.2

##### ----- 以下高级配置一般无需修改 -----
##### !!!注意!!! 请确保理解配置项含义后再做修改
##### 数据库相关配置
db:
  sqliteDbFile: ./etc/easyedge-iec.db
  eventDbFile: ./etc/easyedge-event.db

##### 推流相关配置
livego:
listenHost: 0.0.0.0
hlsPort: 7002
apiPort: 8090
flvPort: 8101
rtmpPort: 1935
server:
  - appName: cameraPreview
    hls: false
    api: false
    flv: true

##### 视频流相关配置
edgestream:
listenHost: 127.0.0.1
listenPort: 24402
# 摄像头预览：识别结果绘制延迟消失
renderExtendFrames: 10
# 预测队列大小: fps=30时，延迟约为2秒
inferenceQueueSize: 60
videoEncodeQueueSize: 20
videoEncodeBitRate: 400000
```

## 离线SDK部署说明

### 专项适配硬件离线部署

## EdgeBoard（FZ）专用SDK集成文档

## 简介

本文档介绍 EasyEdge/EasyDL在EdgeBoard®边缘计算盒/Lite计算卡上的专用软件的使用流程。

EdgeBoard系列硬件可直接应用于AI项目研发与部署，具有高性能、易携带、通用性强、开发简单等四大优点。

详细硬件参数请在[AI市场](#)浏览。

EdgeBoard产品使用手册：<https://ai.baidu.com/ai-doc/HWCE/Yk3b86gvp>

## 软核版本

SDK版本	对应软核
0.5.2+	1.4
0.5.7+	1.5+

SDK升级需配合EdgeBoard硬件软核升级，建议升级软核为SDK对应版本，否则可能出现结果错误或者其他异常。

可以通过 `dmesg | grep "DRIVER Version"` 命令获取EdgeBoard当前的软核版本

## Release Notes

SDK对应的软核说明：[baidu\\_easyedge\\_linux\\_cpp\\_aarch64\\_EdgeBoardFZ1.X\\_gcc7.5\\_v1.Y.Z\\_20210813.tar.gz](#)

软核以及SDK更新情况如下表所示：

时间	版本 (1.Y.X)	说明	EdgeBoardFZ1.5对应的软核	EdgeBoardFZ1.4对应的软核
2021.8.23	1.0.0	第一版！	<a href="https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x">https://ai.baidu.com/ai-doc/HWCE/lkqgcqt5x</a>	<a href="https://ai.baidu.com/ai-doc/HWCE/Lkqjwlziw">https://ai.baidu.com/ai-doc/HWCE/Lkqjwlziw</a>

注意：升级完成相应的软核之后需要重启机器生效。

## 快速开始

开发者从EasyEdge/EasyDL下载的软件部署包中，包含了简单易用的SDK和Demo。只需简单的几个步骤，即可快速部署运行EdgeBoard计算盒。

部署包中包含多版本SDK：

- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.5*`：适用于EdgeBoard 1.5+软核
- `baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.4*`：适用于EdgeBoard 1.4+软核

SDK文件结构

```

baidu_easyedge_linux_cpp_aarch64_EdgeBoardFZ1.5_*
├── README.txt
├── bin
│   ├── easyedge_image_inference
│   ├── easyedge_serving
│   └── easyedge_video_inference
├── include
│   └── easyedge
├── lib
│   ├── libeasyedge.so -> libeasyedge.so.1
│   ├── libeasyedge.so.1 -> libeasyedge.so.1.3.1
│   ├── libeasyedge.so.1.3.1
│   ├── libeasyedge_static.a
│   ├── libeasyedge_videoio.so -> libeasyedge_videoio.so.1
│   ├── libeasyedge_videoio.so.1 -> libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio.so.1.3.1
│   ├── libeasyedge_videoio_static.a
│   ├── libpaddle_full_api_shared.so -> libpaddle_full_api_shared.so.1.8.0
│   ├── libpaddle_full_api_shared.so.1.8.0
│   ├── libverify.so -> libverify.so.1
│   ├── libverify.so.1 -> libverify.so.1.0.0
│   └── libverify.so.1.0.0
├── now_sre.log
├── src
│   ├── CMakeLists.txt
│   ├── cmake
│   ├── common
│   ├── demo_image_inference
│   ├── demo_serving
│   └── demo_video_inference
└── thirdparty
    └── opencv

```

1.1.0+的SDK自带OpenCV，src编译的时候会引用thirdparty/opencv路径下的头文件和库文件。

## Demo使用流程

用户在AI市场购买计算盒之后，请参考以下步骤进行集成和试用。

### 1. 将计算盒连接电源

指示灯亮起，等待约1分钟。

- 参考[EdgeBoard使用文档](#)配置网口或串口连接。登录EdgeBoard计算盒。
- 加载驱动（开机加载一次即可）。

```
insmod /home/root/workspace/driver/{zu9|zu5|zu3}/fpgadv.ko
```

根据购买的版本，选择合适的驱动。若未加载驱动，可能报错：

```
Failed to to fpga device: -1
```

- 设置系统时间（系统时间必须正确）

```
date --set "2019-5-18 20:48:00"
```

### 2. (可选) 启动HTTP服务

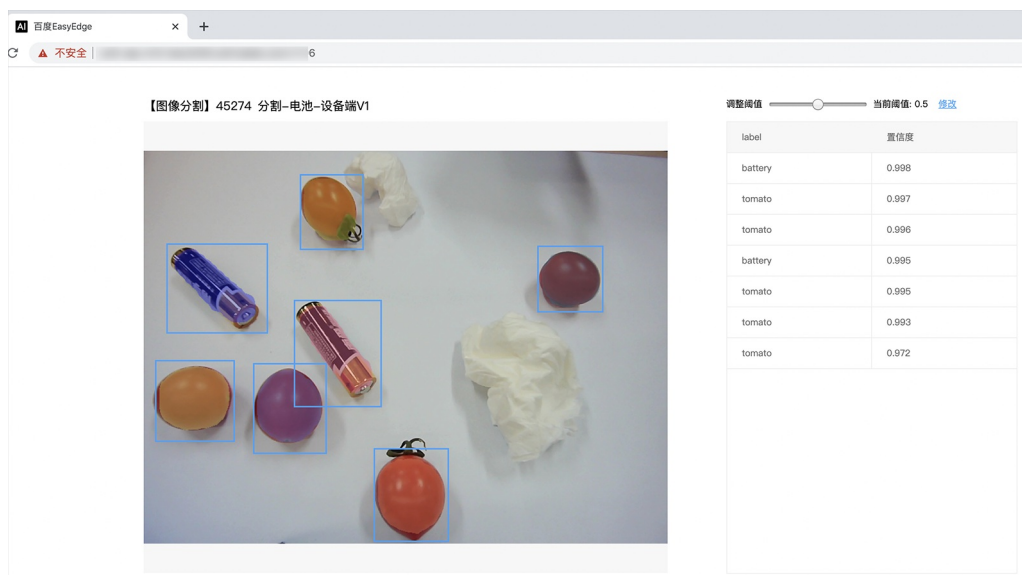
部署包中附带了HTTP服务功能，开发者可以进入SDK根目录，运行easyedge\_serving程序启动HTTP服务。

```
##### ./easyedge_serving {RES目录} "" {绑定的host, 默认0.0.0.0} {绑定的端口, 默认24401}
cd ${SDK_ROOT}
export LD_LIBRARY_PATH=./lib
./demo/easyedge_serving ../../RES ""
```

日志显示

```
2019-07-18 13:27:05,941 INFO [EasyEdge] [http_server.cpp:136] 547974369280 Serving at 0.0.0.0:24401
```

则启动成功。此时可直接在浏览器中输入 `http://{EdgeBoard计算盒ip地址}:24401/`，在h5中测试模型效果。



同时，可以调用HTTP接口来访问盒子。具体参考下文接口说明。

EdgeBoard HTTP Server 目前使用的是单线程处理请求。

### 3. 编译运行Demo

编译：

```
cd src
mkdir build && cd build
cmake .. && make
```

运行

```
./easyedge_image_inference {RES资源文件夹路径} {测试图片路径}
```

便可看到识别结果。

使用说明

使用流程

激活成功之后，有效期内可离线使用。

1. 配置PaddleFluidConfig
2. 新建Predictor :`global_controller()->CreateEdgePredictor(config);`
3. 初始化 predictor->init()
4. 传入图片开始识别predictor->infer(img, ...);

目前EdgeBoard暂不支持并行多模型计算。

接口说明

预测图片

```

/**
 * @brief 同步预测接口
 * inference synchronous
 * Supported by most chip and engine
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @param threshold
 * @return
 */
virtual int infer(
    cv::Mat &image, std::vector<EdgeResultData> &result, float threshold = 0.1
) = 0;

```

## 识别结果说明

EdgeResultData中可以获取对应的分类信息、位置信息。

```

struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // object detection field
    float x1, y1, x2, y2; // (x1, y1): 左上角 , (x2, y2): 右下角 ; 均为0~1的长宽比例值。
};

```

## 关于矩形坐标

- x1 \* 图片宽度 = 检测框的左上角的横坐标
- y1 \* 图片高度 = 检测框的左上角的纵坐标
- x2 \* 图片宽度 = 检测框的右下角的横坐标
- y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考demo文件中使用opencv绘制矩形的逻辑。

## HTTP 私有服务请求说明

### http 请求参数

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤， 0~1	0.1

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

### Python

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

### Cpp label=C#

```
    FileStream fs = new FileStream("./img.jpg", FileMode.Open);
    BinaryReader br = new BinaryReader(fs);
    byte[] img = br.ReadBytes((int)fs.Length);
    br.Close();
    fs.Close();
    string url = "http://127.0.0.1:8402?threshold=0.1";
    HttpRequest request = (HttpRequest)HttpRequest.Create(url);
    request.Method = "POST";
    Stream stream = request.GetRequestStream();
    stream.Write(img, 0, img.Length);
    stream.Close();

    HttpResponse response = request.GetResponse();
    StreamReader sr = new StreamReader(response.GetResponseStream());
    Console.WriteLine(sr.ReadToEnd());
    sr.Close();
    response.Close();
```

Cpp label=C++ 需要安装curl

```

#include <sys/stat.h>
#include <curl/curl.h>
#include <iostream>
#include <string>
#define S_ISREG(m) (((m) & 0170000) == (0100000))
#define S_ISDIR(m) (((m) & 0170000) == (0040000))

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"
", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"
", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s
", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

### Java请求示例

#### http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms, 不含网络交互时间

#### 返回示例



```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

## 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类 `VideoDecoding`，此类提供了获取视频帧数据的便利函数。通过 `VideoConfig` 结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为 SDK `infer` 接口的参数进行预测。

- 接口

class `VideoDecoding` :

```

/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;

```

#### struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};      // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
    is_needed置为false
    int input_fps{0};            // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};      // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;        // frame存储为视频文件的路径
    bool save_all{false};        // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<string, std::string> conf;
};

```

source\_type：输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。source\_value：若source\_type为视频文件，该值为指向视频文件的完整路径；若source\_type为摄像头，该值为摄像头的index，如对于/dev/video0的摄像头，则index为0；若source\_type为网络视频流，则为该视频流的完整地址。skip\_frames：设置跳帧，每隔skip\_frames帧抽取一帧，并把该抽取帧的is\_needed置为true，标记为is\_needed的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。retrieve\_all：若置该项为true，则无论是否设置跳帧，所有的帧都会被

抽取返回，以作为显示或存储用。 `input_fps`：用于抽帧前设置fps。 `resolution`：设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。 `conf`：高级选项。部分配置会通过该map来设置。

#### 注意:

1. 如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
2. 使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

3. 部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。  
具体接口调用流程，可以参考SDK中的`demo_video_inference`。

#### 错误说明

SDK所有主动报出的错误，均覆盖在EdgeStatus枚举中。同时SDK会有详细的错误日志，开发者可以打开Debug日志查看额外说明：

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

#### FAQ

##### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3`

可以通过安装`libcurl3 libcurl-openssl1.0-dev`来解决。如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库`easyedge_static.a`，自己指定需要的Library的版本。

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} paddle-mobile)
```

##### 2. error while loading shared libraries: libeasyedge.so.0.4.0: cannot open shared object file: No such file or directory

类似错误包括`libpaddle-mobile.so`找不到。

直接运行SDK自带的二进制可能会有这个问题，设置`LD_LIBRARY_PATH`为SDK部署包中的lib目录即可。开发者自行使用CMake编译的二进制可以有效管理.so的依赖。

##### 3. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

##### 4. 预测过程中报内存不足“Killed”

此问题仅出现在ZU5，因为FZ5A带vcu，给他预留的内存过大导致，如果用不到VCU可以把这部分改小。修改`/run/media/mmblk1p1/uEnv.txt`：

```
ethaddr=00:0a:35:00:00:09
uenvcmd=fatload mmc 1 0x3000000 image.ub && bootm 0x3000000

bootargs=earlycon console=ttyPS0,115200 clk_ignore_unused cpuidle.off=1 root=/dev/mmblk1p2 rw rootwait cma=128M
```

注意中间空行要保留。

##### 5. 预测结果异常

如果购买的计算盒较早，驱动文件较旧，而SDK比较新（或SDK比较旧，但是计算盒较新），可能出现结果异常，如结果均为空或者nan。请参考“软核版本”小节更新软核和驱动版本。

## 6. 编译过程报错file format not recognized

```
libeasymedge.so: file format not recognized; treating as linker script
```

下载的SDK zip包需要放到板子内部后，再解压、编译。

## 7. 提示 driver\_version(1.4.0) not match paddle\_lite\_version(1.5.1)

需更新驱动，否则可能导致结果异常。参考“软核版本”小节。

## 服务器离线部署

### Linux集成文档-C

#### 简介

本文档介绍Linux CPP SDK的使用方法。

- 网络类型支持：图像分类，物体检测
- 硬件支持：
  - CPU 基础版: x86\_64
  - NVIDIA GPU: x86\_64 PC
- 操作系统支持：Linux

根据开发者的选择，实际下载的版本可能是以下版本之一：

- EasyDL图像
  - x86 CPU 基础版
  - Nvidia GPU 基础版

#### Release Notes

时间	版本	说明
2021.8.23	1.0.0	第一版！

#### 快速开始

SDK在以下环境中测试通过

- x86\_64, Ubuntu 16.04, gcc 5.4
- x86\_64, Ubuntu 18.04, gcc 7.4
- Tesla P4, Ubuntu 16.04, cuda 9.0, cudnn 7.5
- x86\_64, Ubuntu 16.04, gcc 5.4, XTCL r1.0

#### 依赖包括

- cmake 3+
- gcc 5.4 (需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.11 (可选)
- cuda9.0\_cudnn7 (使用NVIDIA-GPU时必须)
- XTCL 1.0.0.187 (使用昆仑服务器时必须)

#### 1. 安装依赖

以下步骤均可选，请开发者根据实际运行环境选择安装。

#### (可选) 安装cuda& cudnn

##### 在NVIDIA GPU上运行必须

对于GPU基础版，若开发者需求不同的依赖版本，请在[PaddlePaddle官网](#) 下载对应版本的libpaddle\_fluid.so或参考其文档进行编译，覆盖lib文件夹下的相关库文件。

#### (可选) 安装TensorRT

下载包中提供了对应 cuda9.0、cuda10.0、cuda10.2、cuda11.0和cuda11.1 五个版本的 SDK，cuda9.0 和 cuda10.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.0.0.11，cuda10.2 和 cuda11.0 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.1.3.4，cuda11.1 的 SDK 默认依赖的 TensorRT 版本为 TensorRT7.2.3.4，请在[这里](#)下载对应 cuda 版本的 TensorRT，并把其中的lib文件拷贝到系统lib目录，或其他目录并设置环境变量。

#### (可选) 安装XTCL

请安装与1.0.0.187版本兼容的XTCL。必要时，请将运行库路径添加到环境变量。

## 2. 测试Demo

模型资源文件默认已经打包在开发者下载的SDK包中。Demo工程直接编译即可运行。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；在Intel CPU上运行CPU加速版，如果thirdparty里包含openvino文件夹的，必须在编译或运行demo程序前执行以下命令：source \${cpp\_kit位置路径}/thirdparty/openvino/bin/setupvars.sh

部分SDK中已经包含预先编译的二进制，如 bin/easyedge\_demo, bin/easyedge\_serving，配置LD\_LIBRARY\_PATH后，可直接运行：  
LD\_LIBRARY\_PATH=./lib ./bin/easyedge\_serving

编译运行：

```
cd src
mkdir build && cd build
cmake .. && make
./easyedge_image_inference {模型RES文件夹} {测试图片路径}
##### 如果是NNIE引擎，使用sudo运行
sudo ./easyedge_image_inference {模型RES文件夹} {测试图片路径}
```

如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEGE_BUILD_OPENCV=ON
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

demo运行效果：



图片加载失败

```
> ./easyedge_image_inference .././../RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddlev2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddlev2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

## 4. 测试Demo HTTP 服务

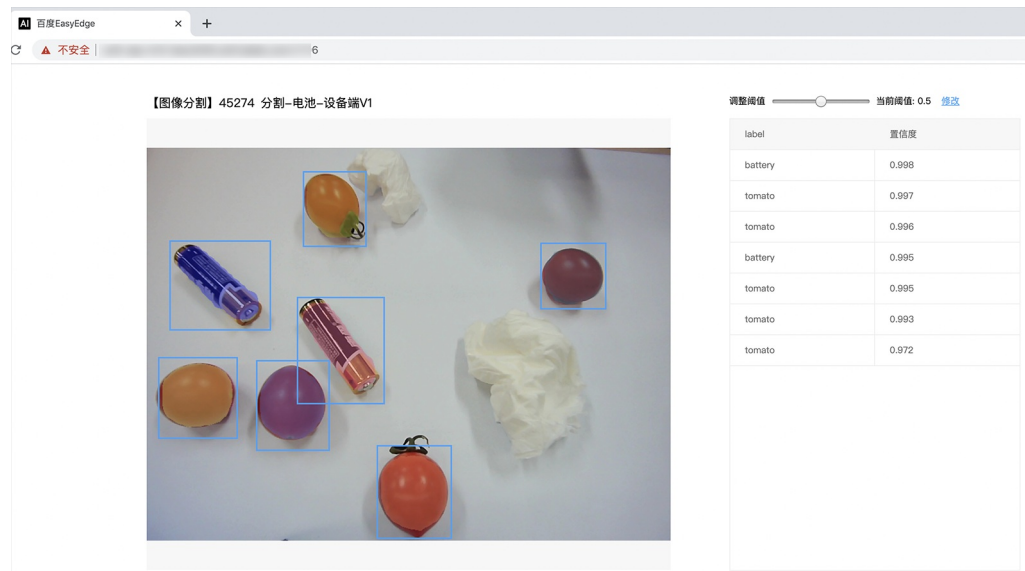
编译demo完成之后，会同时生成一个http服务 运行

```
##### ./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
./easyedge_serving ../../RES "1111-1111-1111-1111" 0.0.0.0 24401
```

后，日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，`http://{设备ip}:24401`，



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

### 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

### 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置模型资源目录
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor ; 在这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

输入图片不限制大小

**SDK参数配置** SDK的参数通过 EdgePredictorConfig::set\_config和global\_controller()->set\_config配置。set\_config的所有key在easyedge\_xxxx\_config.h中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过EdgePredictorConfig::set\_config设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过global\_controller()->set\_config设置

以序列号为例，KEY的说明如下：

```

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

使用方法如下：

```

EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");

```

具体支持的运行参数配置列表可以参考开发工具包中的头文件的详细说明。

相关配置均可以通过环境变量的方法来设置，对应的key名称加上前缀EDGE\_即为环境变量的key。如序列号配置的环境变量key为EDGE\_PREDICTOR\_KEY\_SERIAL\_NUM，如指定CPU线程数的环境变量key为EDGE\_PREDICTOR\_KEY\_CPU\_THREADS\_NUM。注意：通过代码设置的配置会覆盖通过环境变量设置的值。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;

/**
 * @brief
 * 批量图片推理接口
 * @param image: must be BGR , HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    std::vector<cv::Mat>& image, std::vector<std::vector<EdgeResultData>>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测、图像分割时才有意义
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割的模型, 该字段才有意义
    // 请注意: 图像分割时, 以下两个字段会比较大, 使用完成之后请及时释放EdgeResultData
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask

    // 目标追踪模型, 该字段才有意义
    int trackid; // 轨迹id
    int frame; // 处于视频中的第几帧
    EdgeTrackStat track_stat; // 跟踪状态
};
```

## 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标



$y_2 * \text{图片高度} = \text{检测框的右下角的纵坐标}$

### 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域，0代表非目标区域
```

### 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](#)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

### 日志配置

设置 `EdgeLogConfig` 的相关参数。具体含义参考文件中的注释说明。

```
EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);
```

### http服务

1. 开启http服务 http服务的启动可以参考demo\_serving.cpp文件。

```
/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);
```

### 2. 请求http服务

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片或视频来进行测试。

#### http 请求方式一：无额外编码

- 图片测试：不使用图片base64格式

URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

## Python请求示例

```
import requests

with open('./1.jpg', 'rb') as f:
    img_data = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img_data).json()
```

## Java请求示例

## http 请求方法二：图片使用base64格式

HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 分类网络：body中请求示例

```
{
  "image": "<base64数据>"
  "top_num": 5
}
```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 检测和分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

## http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms，不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

## 其他配置

### 1. 日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::CONTROLLER_KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



### 2. CPU线程数设置

CPU线程数可通过 `EdgePredictorConfig::set_config`配置

```
EdgePredictorConfig config;
config.set_config(easyedge::params::PREDICTOR_KEY_CPU_THREADS_NUM, 4);
```

### 3. 批量预测设置

```
EdgePredictorConfig config;
config.set_config(params::PREDICTOR_KEY_VINO_MAX_BATCH_SIZE, 4); //CPU加速版特殊设置

int batch_size = 2; // 使用前修改batch_size再编译、执行
while (get_next_batch(imgs, img_files, batch_size, start_index)) {
  ...
}
```

**CPU加速版设置** `PREDICTOR_KEY_VINO_MAX_BATCH_SIZE`含义：此值用来控制批量图片预测可以支持的最大图片数，实际预测的时候单次预测图片数不可大于此值。实际预测图片数为`std::min(PREDICTOR_KEY_VINO_MAX_BATCH_SIZE, batch_size)`。

## FAQ

### 1. 如何处理一些 undefined reference?

如：undefined reference to `curl\_easy\_setopt@CURL\_OPENSSL\_3'

- 方案1：通过安装libcurl3 libcurl-openssl1.0-dev来解决。
- 方案2：如果开发者想不想使用低版本的openssl（如Ubuntu 18.04），可以link静态库easyedge\_static.a，自己指定需要的Library的版本：

示例：修改CMakeList.txt

```
find_package(CURL REQUIRED)
target_link_libraries(easyedge_demo ${OpenCV_LIBS} easyedge_static pthread ${CURL_LIBRARIES} verify_static ${其他需要的库})
```

其中，其他需要的库视具体sdk中包含的库而定。

## 2. NVIDIA GPU预测时，报错显存不足

如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请根据显存大小和模型配置。调整合适的初始 fraction\_of\_gpu\_memory。参数的含义参考[这里](#)。

## 3. 如何将我的模型运行为一个http服务？

目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

## 4. 运行NNIE引擎报permission denied

日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

## 5. 运行SDK报错 Authorization failed

情况一：日志显示 Http perform failed: null respond

在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

情况二：日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx)

此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 ~/.baidu/easyedge 目录，再重新激活。

## 6. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

### 7. 运行二进制时，提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath, 可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后, 再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:../lib ./easyedge_demo
```

### 8. 运行二进制时提示 libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory

同上面8的问题类似, 没有正确设置动态库的查找路径, 可通过设置LD\_LIBRARY\_PATH为sdk的thirdparty/opencv/lib文件夹解决

```
export LD_LIBRARY_PATH=${LD_LIBRARY_PATH}:../thirdparty/opencv/lib
(tips: 上面冒号后面接的thirdparty/opencv/lib路径以实际项目中路径为准, 比如也可能是../../thirdparty/opencv/lib)
```

### 9. 编译时报错：file format not recognized

可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中, 再解压缩、编译

### 10. 进行视频解码时, 报错符号未找到、格式不支持、解析出的图片为空、无法设置抽帧

请确保安装OpenCV时, 添加了-DWITH\_FFMPEG=ON选项 (或者GStream选项), 并且检查OpenCV的安装日志中, 关于Video I/O段落的说明是否为YES。

```
-- Video I/O:
-- DC1394:      YES (ver 2.2.4)
-- FFmpeg:     YES
-- avcodec:    YES (ver 56.60.100)
-- avformat:   YES (ver 56.40.101)
-- avutil:     YES (ver 54.31.100)
-- swscale:    YES (ver 3.1.101)
-- avresample: NO
-- libv4l/libv4l2: NO
-- v4l/v4l2:   linux/videodev2.h
```

如果为NO, 请搜索相关解决方案, 一般为依赖没有安装, 以apt为例：

```
apt-get install yasm libjpeg-dev libjasper-dev libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev libgstreamer0.10-dev
libgstreamer-plugins-base0.10-dev libv4l-dev python-dev python-numpy libtbb-dev libqt4-dev libgtk2.0-dev libfaac-dev libmp3lame-dev
libopencore-amrnb-dev libopencore-amrwb-dev libtheora-dev libvorbis-dev libxvidcore-dev x264 v4l-utils ffmpeg
```

## Linux集成文档-Python

### 简介

本文档介绍Linux Python SDK 的使用方法。

- 网络类型支持：图像分类, 物体检测
- 硬件支持：
  - Linux x86\_64 CPU
  - Linux x86\_64 Nvidia GPU
- 语言支持：Python 3.5, 3.6, 3.7

### Release Notes

时间	版本	说明
2021.8.23	1.0.0	第一版！

### 快速开始

#### 1. 安装依赖

- 根据引擎的不同，SDK 依赖了不同的底层引擎。根据所需自行安装。

#### 安装 paddlepaddle

- 使用x86\_64 CPU预测时必须安装：

```
pip3 install -U paddlepaddle
```

若 CPU 为特殊型号，如赛扬处理器（一般用于深度定制的硬件中），请关注 CPU 是否支持 avx 指令集。如果不支持，请在[paddle官网](#)安装 noavx 版本

- 使用NVIDIA GPU预测时必须安装：

```
pip3 install -U paddlepaddle-gpu
```

如果环境非 cuda9 cudnn7，请参考[paddle文档](#)安装合适的 paddle 版本。不被 paddle 支持的 cuda 和 cudnn 版本，EasyEdge 暂不支持

## 2. 安装 easyedge python wheel 包

```
pip3 install -U BaiduAI_EasyEdge_SDK-{版本号}-cp36-cp36m-linux_x86_64.whl
```

具体名称以 SDK 包中的 whl 为准。

## 3. 测试 Demo

### 图片预测

输入对应的模型文件夹（默认为RES）和测试图片路径，运行：

```
python3 demo.py {model_dir} {image_name.jpg}
```

测试效果：



4. 测试Demo HTTP 服务 输入对应的模型文件夹（默认为RES）、序列号、设备ip和指定端口号，运行：

```
python3 demo_serving.py {model_dir} "" {host, default 0.0.0.0} {port, default 24401}
```

后，会显示：

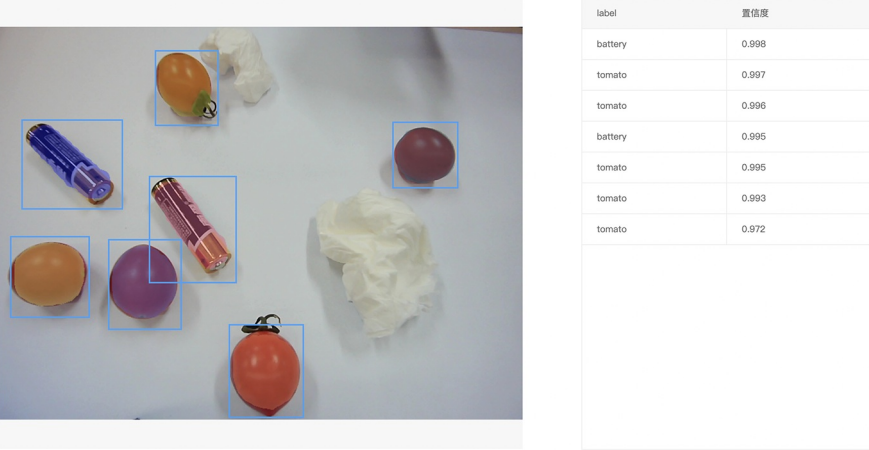
```
Running on http://0.0.0.0:24401/
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片或者视频来进行测试。也可以参考`demo\_serving.py`里 `http_client_test()`函数请求http服务进行推理。

百度EasyEdge

不安全 | 6

【图像分割】45274 分割-电池-设备端V1



调整阈值  当前阈值: 0.5 [修改](#)

label	置信度
battery	0.998
tomato	0.997
tomato	0.996
battery	0.995
tomato	0.995
tomato	0.993
tomato	0.972



图片加载失败

## 使用说明

使用流程 demo.py

```
import BaiduAI.EasyEdge as edge
```

```
pred = edge.Program()
pred.init(model_dir={RES文件夹路径}, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
pred.infer_image((numpy.ndarray的图片))
pred.close()
```

demo\_serving.py

```
import BaiduAI.EasyEdge as edge
from BaiduAI.EasyEdge.serving import Serving

server = Serving(model_dir={RES文件夹路径})
##### 请参考同级目录下demo.py里:
##### pred.init(model_dir=xx, device=xx, engine=xx, device_id=xx)
##### 对以下参数device\device_id和engine进行修改
server.run(host=host, port=port, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID)
```

## 初始化

- 接口

```
def init(self,
        model_dir,
        device=Device.LOCAL,
        engine=Engine.PADDLE_FLUID,
        config_file='conf.json',
        preprocess_file='preprocess_args.json',
        model_file='model',
        params_file='params',
        graph_file='graph.ncsmodel',
        label_file='label_list.txt',
        device_id=0
    ):
    """
    Args:
        device: Device.CPU
        engine: Engine.PADDLE_FLUID
        model_dir: str
            model dir
        preprocess_file: str
        model_file: str
        params_file: str
        graph_file: str
        label_file: str
        device_id: int

    Raises:
        RuntimeError, IOError

    Returns:
        bool: True if success

    """
```

使用 NVIDIA GPU 预测时，必须满足：

- 机器已安装 cuda, cudnn
- 已正确安装对应 cuda 版本的 paddle 版本
- 通过设置环境变量 `FLAGS_fraction_of_gpu_memory_to_use` 设置合理的初始内存使用比例

使用 CPU 预测时，可以通过在 `init` 中设置 `thread_num` 使用多线程预测。如：

```
pred.init(model_dir=_model_dir, device=edge.Device.CPU, engine=edge.Engine.PADDLE_FLUID, thread_num=1)
```

### 预测图像

- 接口



```
def infer_image(self, img,
                threshold=0.3,
                channel_order='HWC',
                color_format='BGR',
                data_type='numpy'):
    """
    Args:
        img: np.ndarray or bytes
        threshold: float
            only return result with confidence larger than threshold
        channel_order: string
            channel order HWC or CHW
        color_format: string
            color format order RGB or BGR
        data_type: string
            image data type

    Returns:
        list

    """
```

- 返回格式: [dict1, dict2, ...]

字段	类型	取值	说明
confidence	float	0~1	分类或检测的置信度
label	string		分类或检测的类别
index	number		分类或检测的类别
x1, y1	float	0~1	物体检测，矩形的左上角坐标（相对长宽的比例值）
x2, y2	float	0~1	物体检测，矩形的右下角坐标（相对长宽的比例值）
mask	string/numpy.ndarray	图像分割的mask	

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

可以参考 demo 文件中使用 opencv 绘制矩形的逻辑。

#### 结果示例

- i) 图像分类

```
{
  "index": 736,
  "label": "table",
  "confidence": 0.9
}
```

- ii) 物体检测

```
{
  "y2": 0.91211,
  "label": "cat",
  "confidence": 1.0,
  "x2": 0.91504,
  "index": 8,
  "y1": 0.12671,
  "x1": 0.21289
}
```

## FAQ

### 1. 运行时报错 "非法指令" 或 "illegal instruction"

可能是 CPU 缺少 avx 指令集支持，请在[paddle官网](#) 下载 noavx 版本覆盖安装

### 2. NVIDIA GPU预测时，报错显存不：

如以下错误字样：

```
paddle.fluid.core.EnforceNotMet: Enforce failed. Expected allocating <= available, but received allocating:20998686233 >
available:19587333888.
Insufficient GPU memory to allocation. at [/paddle/paddle/fluid/platform/gpu_info.cc:170]
```

请在运行 Python 前设置环境变量，通过 `export FLAGS_fraction_of_gpu_memory_to_use=0.3` 来限制SDK初始使用的显存量，0.3表示初始使用30%的显存。如果设置的初始显存较小，SDK 会自动尝试 allocate 更多的显存。

### 3. 我想使用多线程预测，怎么做？

如果需要多线程预测，可以每个线程启动一个Program实例，进行预测。demo.py文件中有相关示例代码。

注意：对于CPU预测，SDK内部是可以使用多线程，最大化硬件利用率。参考init的thread\_num参数。

## 纯离线SDK简介

本文档主要说明定制化模型发布后获得的服务器端SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 进入[EasyDL社区交流](#)，与其他开发者进行互动

## SDK说明

图像分类服务器端SDK支持Linux、Windows两种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux		Intel CPU: x86_64 NVIDIA GPU: x86_64
Windows	64位 Windows7 及以上	NVIDIA GPU: x86_64  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015  GPU依赖： CUDA 9.x + cuDNN 7.x

## 激活&使用步骤

离线SDK的激活与使用分以下步骤：

## 1. 本地运行SDK，并完成首次联网激活

通过左侧导航栏查看不同操作系统SDK的开发文档

## 2. 正式使用

### SDK常见问题

通过左侧导航栏查看不同操作系统SDK的FAQ

以下是通用FAQ，如您的问题仍未解决，请在百度智能云控制台内[提交工单](#)反馈

### 1、SDK如何激活？有效期是多少？

在免费试用期范围内，SDK有效期飞桨EasyDL-桌面版软件的试用期保持一致，通过飞桨EasyDL-高级版发布的SDK，可永久使用。

如有其他异常请在百度智能云控制台内[提交工单](#)反馈

## Windows集成文档

### 简介

本文档介绍Windows GPU SDK的使用方法。

- 网络类型支持：图像分类，物体检测
- 硬件支持：
  - NVIDIA GPU
- 操作系统支持
  - 64位 Windows 7 及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015
- GPU 版依赖（必须安装以下版本） \* CUDA 9.0.x + cuDNN 7.6.x 或者 CUDA 10.0.x + cuDNN 7.6.x
- 协议
  - HTTP

Release Notes | 时间 | 版本 | 说明 | |-----|-----|-----| | 2021.8.23 | 1.0.0 | 第一版！ |

### 快速开始

#### 1. 安装依赖

##### 安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

##### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

##### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

如果使用GPU版SDK，请安装CUDA + cuDNN

https://developer.nvidia.com/cuda  
https://developer.nvidia.com/cudnn

注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验“，点击安装，安装之后重启即可。

2. 运行离线SDK

解压下载好的SDK，SDK默认使用cuda9版本，如果需要cuda10请运行EasyEdge CUDA10.0.bat切换到cuda10版本，



点击"启动服务"，等待数秒即可启动成功，本地服务默认运行在~~

http://127.0.0.1:24401/

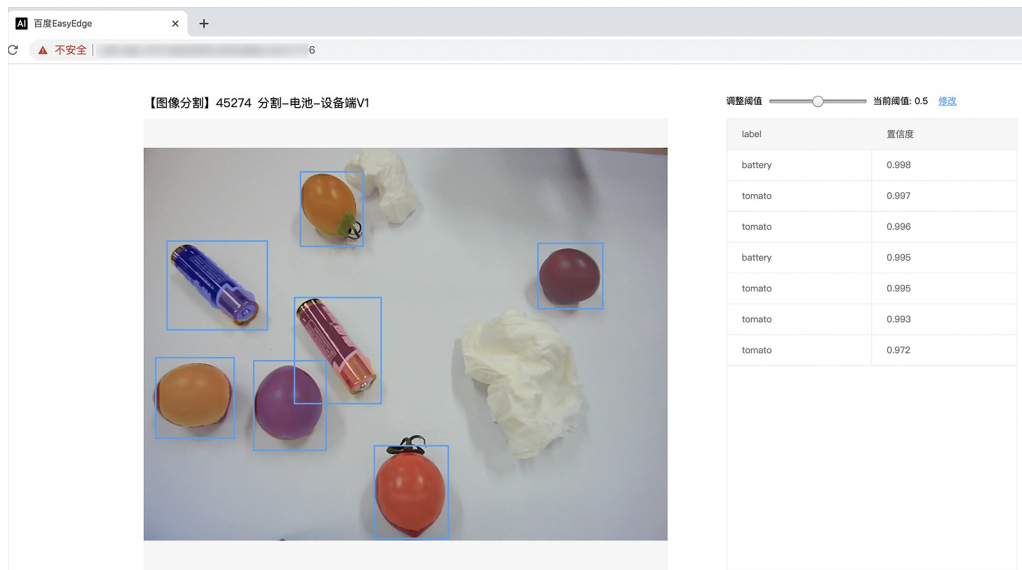
其他任何语言只需通过HTTP调用即可。

如启动失败，可参考如下步骤排查：



Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入http://127.0.0.1:24401，在h5中测试模型效果。



使用说明

图像服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={"threshold": 0.1},
                       data=img.json())
```

C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpWebRequest request = (HttpWebRequest)HttpWebRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

C++ 使用示例代码如下，需要安装curl

```

##### include <sys/stat.h>
##### include <curl/curl.h>

##### define S_ISREG(m) (((m) & 0170000) == (0100000))
##### define S_ISDIR(m) (((m) & 0170000) == (0040000))

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";

    FILE *fp = NULL;
    struct stat stbuf = { 0, };

    fp = fopen(post_data_filename, "rb");

    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }

    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }

    CURL *curl;
    CURLcode res;

    curl_global_init(CURL_GLOBAL_ALL);

    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);

        res = curl_easy_perform(curl);
        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);

    return 0;
}

```

### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|-----| | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 | | x1, y1 | float | 0~1 | 物体检测，矩形的左上角坐标（相对长宽的比例值） | | x2, y2 | float | 0~1 | 物体检测，矩形的右下角坐标（相对长宽的比例值） |

### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

## FAQ

## 1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下： *.NET Framework 4.5* Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

GPU依赖，版本必须如下： \* CUDA 9.0.x + cuDNN 7.6.x 或者 CUDA 10.0.x + cuDNN 7.6.x

GPU加速版（EasyEdge-win-x86-nvidia-gpu-tensorrt）依赖，版本必须如下： *CUDA 9.0.x + cuDNN 7.6.x 或者 CUDA 10.0.x + cuDNN 7.6.x* TensorRT 7.x 必须和CUDA版本对应

GPU加速版（EasyEdge-win-x86-nvidia-gpu-paddletrt）依赖，版本必须如下： *CUDA 11.0.x + cuDNN 8.0.x* TensorRT 7.1.3.4 必须和CUDA版本对应

## 2. 服务调用时返回为空，怎么处理？

调用输入的图片必须是RGB格式，请确认是否有alpha通道。

## 3. 多个模型怎么同时使用？

SDK设置运行不同的端口，点击运行即可。

## 4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

## 5. 启动失败，缺失DLL？

打开EasyEdge.log，查看日志错误，根据提示处理 缺失DLL，请使用 <https://www.dependencywalker.com/> 查看相应模块依赖DLL缺失哪些，请自行下载安装

## 6. 启动失败，报错NotDecrypted？

Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

## 7. 其他问题

如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

## 通用小型设备 离线部署

## Linux集成文档-C++

## 简介

本文档介绍EasyEdge/EasyDL的Linux CPP SDK的使用方法。

- 网络类型支持： - 图像分类 - 物体检测
- 硬件支持：
  - CPU: aarch64 armv7hf
- 操作系统支持：
  - Linux (Ubuntu, Centos, Debian等)
  - 海思HiLinux
  - 树莓派Raspbian/Debian
  - 瑞芯微Firefly

Release Notes | 时间 | 版本 | 说明 | |---| |---| |---| | 2021.8.23 | 1.0.0 | 第一版！ |

## 快速开始

## 安装依赖

## 依赖包括

- cmake 3+
- gcc 5.4 以上(需包含 GLIBCXX\_3.4.22) , gcc / glibc版本请以实际SDK ldd的结果为准
- opencv3.4.5 (可选)

#### 依赖说明：树莓派

树莓派Raspberry默认为armv7hf系统，使用SDK包中名称中包含 armv7hf\_ARM\_的tar包。如果是aarch64系统，使用SDK包中名称中包含 aarch64\_ARM\_的tar包。

在安装前可通过以下命令查看是32位还是64位：

```
getconf LONG_BIT
32
```

#### 测试Demo

模型资源文件默认已经打包在开发者下载的SDK包中。

Demo工程直接编译即可运行。

请先将tar包整体拷贝到具体运行的设备中，再解压缩编译；

对于硬件使用为：-Intel Movidius Myriad2 / Myriad X on Linux x86\_64 / armv7hf / aarch64，在编译或运行demo程序前执行以下命令：

```
source ${cpp_kit位置路径}/thirdparty/opencvino/bin/setupvars.sh
```

部分SDK中已经包含预先编译的二进制，bin/easyedge\_demo, bin/easyedge\_serving，配置LD\_LIBRARY\_PATH后，可直接运行：

```
LD_LIBRARY_PATH=./lib ./bin/easyedge_serving
```

编译运行：

```
cd src
mkdir build && cd build
cmake .. && make
./easyedge_image_inference {模型RES文件夹} {测试图片路径}
```

如果希望SDK自动编译安装所需要的OpenCV库，修改cmake的optionEDGE\_BUILD\_OPENCV为ON即可。SDK会自动从网络下载opencv源码，并编译需要的module、链接。注意，此功能必须需联网。

```
cmake -DEGE_BUILD_OPENCV=ON .. && make -j16
```

若需自定义library search path或者gcc路径，修改CMakeList.txt即可。

对于硬件使用为Intel Movidius Myriad2 / Myriad X 的，如果宿主机找不到神经计算棒Intel® Neural Compute Stick，需要执行以下命令添加USB Rules：

```
cp ${cpp_kit位置路径}/thirdparty/opencvino/deployment_tools/inference_engine/external/97-myriad-usbboot.rules /etc/udev/rules.d/
sudo udevadm control --reload-rules
sudo udevadm trigger
sudo ldconfig
```

demo运行效果：



图片加载失败



```
> ./easyedge_image_inference ../.././RES 2.jpeg
2019-02-13 16:46:12,659 INFO [EasyEdge] [easyedge.cpp:34] 140606189016192 Baidu EasyEdge Linux Development Kit
0.2.1(20190213)
2019-02-13 16:46:14,083 INFO [EasyEdge] [paddle2_edge_predictor.cpp:60] 140606189016192 Allocate graph success.
2019-02-13 16:46:14,326 DEBUG [EasyEdge] [paddle2_edge_predictor.cpp:143] 140606189016192 Inference costs 168 ms
1, 1:txt_frame, p:0.994905 loc: 0.168161, 0.153654, 0.920856, 0.779621
Done
```

## 测试Demo HTTP 服务

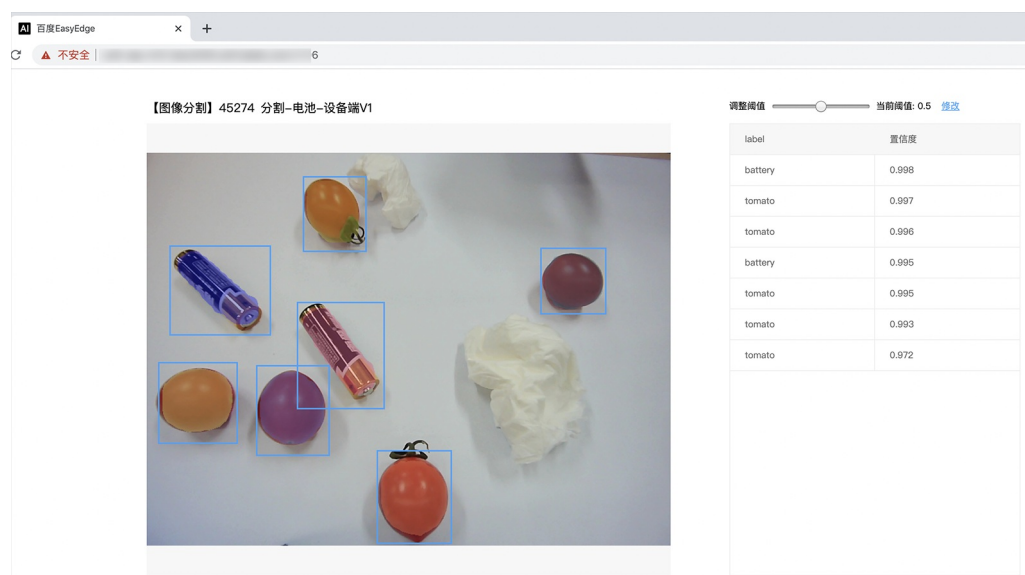
编译demo完成之后，会同时生成一个http服务 运行

```
##### ./easyedge_serving {res_dir} {serial_key} {host, default 0.0.0.0} {port, default 24401}
./easyedge_serving ../.././RES "1111-1111-1111-1111" 0.0.0.0 24401
```

后，日志中会显示

```
HTTP is now serving at 0.0.0.0:24401
```

字样，此时，开发者可以打开浏览器，<http://{设备ip}:24401>，选择图片来进行测试。



同时，可以调用HTTP接口来访问服务，具体参考下文接口说明。

## 使用说明

使用该方式，将运行库嵌入到开发者的程序当中。

## 使用流程

请优先参考Demo的使用流程。遇到错误，请优先参考文件中的注释解释，以及日志说明。

```

// step 1: 配置运行参数
EdgePredictorConfig config;
config.model_dir = {模型文件目录};

// step 2: 创建并初始化Predictor; 这里选择合适的引擎
auto predictor = global_controller()->CreateEdgePredictor(config);

// step 3-1: 预测图像
auto img = cv::imread(图片路径);
std::vector<EdgeResultData> results;
predictor->infer(img, results);

// step 3-2: 预测视频
std::vector<EdgeResultData> results;
FrameTensor frame_tensor;
VideoConfig video_config;
video_config.source_type = static_cast<SourceType>(video_type); // source_type 定义参考头文件 easyedge_video.h
video_config.source_value = video_src;
/*
... more video_configs, 根据需要配置video_config的各选项
*/
auto video_decoding = CreateVideoDecoding(video_config);
while (video_decoding->next(frame_tensor) == EDGE_OK) {
    results.clear();
    if (frame_tensor.is_needed) {
        predictor->infer(frame_tensor.frame, results);
        render(frame_tensor.frame, results, predictor->model_info().kind);
    }
    //video_decoding->display(frame_tensor); // 显示当前frame, 需在video_config中开启配置
    //video_decoding->save(frame_tensor); // 存储当前frame到视频, 需在video_config中开启配置
}

```

对于口罩检测模型，将 `EdgePredictorConfig config`修改为`PaddleMultiStageConfig config`即可。

口罩检测模型请注意输入图片中人脸大小建议保持在 88到9696像素之间，可根据场景远近程度缩放图片后再传入SDK。

**SDK参数配置** SDK的参数通过`EdgePredictorConfig::set_config`和`global_controller()->set_config`配置。`set_config`的所有key在`easyedge_xxxx_config.h`中。其中

- PREDICTOR前缀的key是不同模型相关的配置，通过`EdgePredictorConfig::set_config`设置
- CONTROLLER前缀的key是整个SDK的全局配置，通过`global_controller()->set_config`设置

以序列号为例，KEY的说明如下：

```

/**
 * @brief 序列号设置；序列号不设置留空时，SDK将会自动尝试使用本地已经激活成功的有效期内的序列号
 * 值类型：string
 * 默认值：空
 */
static constexpr auto PREDICTOR_KEY_SERIAL_NUM = "PREDICTOR_KEY_SERIAL_NUM";

```

使用方法如下：

```

EdgePredictorConfig config;
config.model_dir = ...;
config.set_config(params::PREDICTOR_KEY_SERIAL_NUM, "1DB7-1111-1111-D27D");

```

具体支持的运行参数可以参考开发工具包中的头文件的详细说明。

## 初始化

- 接口

```
auto predictor = global_controller()->CreateEdgePredictor(config);
predictor->init();
```

若返回非0，请查看输出日志排查错误原因。

## 预测图像

- 接口

```
/**
 * @brief
 * 通用接口
 * @param image: must be BGR, HWC format (opencv default)
 * @param result
 * @return
 */
virtual int infer(
    cv::Mat& image, std::vector<EdgeResultData>& result
) = 0;
```

图片的格式务必为opencv默认的BGR, HWC格式。

- 返回格式

EdgeResultData中可以获取对应的分类信息、位置信息。

```
struct EdgeResultData {
    int index; // 分类结果的index
    std::string label; // 分类结果的label
    float prob; // 置信度

    // 物体检测活图像分割时才有
    float x1, y1, x2, y2; // (x1, y1): 左上角, (x2, y2): 右下角; 均为0~1的长宽比例值。

    // 图像分割时才有
    cv::Mat mask; // 0, 1 的mask
    std::string mask_rle; // Run Length Encoding, 游程编码的mask
};
```

## 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

## 关于图像分割mask

```
cv::Mat mask为图像掩码的二维数组
{
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 1, 1, 1, 0, 0, 0, 0},
    {0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
}
其中1代表为目标区域，0代表非目标区域
```

## 关于图像分割mask\_rle

该字段返回了mask的游程编码，解析方式可参考 [http demo](http://demo)

以上字段可以参考demo文件中使用opencv绘制的逻辑进行解析

## 预测视频

SDK 提供了支持摄像头读取、视频文件和网络视频流的解析工具类VideoDecoding，此类提供了获取视频帧数据的便利函数。通过VideoConfig结构体可以控制视频/摄像头的解析策略、抽帧策略、分辨率调整、结果视频存储等功能。对于抽取到的视频帧可以直接作为SDK infer 接口的参数进行预测。

- 接口

class VideoDecoding :

```
/**
 * @brief 获取输入源的下一帧
 * @param frame_tensor
 * @return
 */
virtual int next(FrameTensor &frame_tensor) = 0;

/**
 * @brief 显示当前frame_tensor中的视频帧
 * @param frame_tensor
 * @return
 */
virtual int display(const FrameTensor &frame_tensor) = 0;

/**
 * @brief 将当前frame_tensor中的视频帧写为本地视频文件
 * @param frame_tensor
 * @return
 */
virtual int save(FrameTensor &frame_tensor) = 0;

/**
 * @brief 获取视频的fps属性
 * @return
 */
virtual int get_fps() = 0;

/**
 * @brief 获取视频的width属性
 * @return
 */
virtual int get_width() = 0;

/**
 * @brief 获取视频的height属性
 * @return
 */
virtual int get_height() = 0;
```

struct VideoConfig

```

/**
 * @brief 视频源、抽帧策略、存储策略的设置选项
 */
struct VideoConfig {
    SourceType source_type;        // 输入源类型
    std::string source_value;      // 输入源地址，如视频文件路径、摄像头index、网络流地址
    int skip_frames{0};           // 设置跳帧，每隔skip_frames帧抽取一帧，并把该抽取帧的is_needed置为true
    int retrieve_all{false};       // 是否抽取所有frame以便于作为显示和存储，对于不满足skip_frames策略的frame，把所抽取帧的
is_needed置为false
    int input_fps{0};             // 在采取抽帧之前设置视频的fps
    Resolution resolution{Resolution::kAuto}; // 采样分辨率，只对camera有效

    bool enable_display{false};
    std::string window_name{"EasyEdge"};
    bool display_all{false};      // 是否显示所有frame，若为false，仅显示根据skip_frames抽取的frame

    bool enable_save{false};
    std::string save_path;        // frame存储为视频文件的路径
    bool save_all{false};        // 是否存储所有frame，若为false，仅存储根据skip_frames抽取的frame

    std::map<std::string, std::string> conf;
};

```

- `source_type` : 输入源类型，支持视频文件、摄像头、网络视频流三种，值分别为1、2、3。
- `source_value` : 若`source_type`为视频文件，该值为指向视频文件的完整路径；若`source_type`为摄像头，该值为摄像头的index，如对于`/dev/video0`的摄像头，则index为0；若`source_type`为网络视频流，则为该视频流的完整地址。
- `skip_frames` : 设置跳帧，每隔`skip_frames`帧抽取一帧，并把该抽取帧的`is_needed`置为true，标记为`is_needed`的帧是用来做预测的帧。反之，直接跳过该帧，不经过预测。
- `retrieve_all` : 若置该项为true，则无论是否设置跳帧，所有的帧都会被抽取返回，以作为显示或存储用。
- `input_fps` : 用于抽帧前设置fps。
- `resolution` : 设置摄像头采样的分辨率，其值请参考`easyedge_video.h`中的定义，注意该分辨率调整仅对输入源为摄像头时有效。
- `conf` : 高级选项。部分配置会通过该map来设置。

### 注意

- 1.如果使用VideoConfig的display功能，需要自行编译带有GTK选项的opencv，默认打包的opencv不包含此项。
- 2.使用摄像头抽帧时，如果通过`resolution`设置了分辨率调整，但是不起作用，请添加如下选项：

```
video_config.conf["backend"] = "2";
```

- 3.部分设备上的CSI摄像头尚未兼容，如遇到问题，可以通过工单、QQ交流群或微信交流群反馈。  
具体接口调用流程，可以参考SDK中的`demo_video_inference`。

### 日志配置

设置 EdgeLogConfig 的相关参数。具体含义参考文件中的注释说明。

```

EdgeLogConfig log_config;
log_config.enable_debug = true;
global_controller()->set_log_config(log_config);

```

### http服务

1. 开启http服务 http服务的启动参考`demo_serving.cpp`文件。

```

/**
 * @brief 开启一个简单的demo http服务。
 * 该方法会block直到收到sigint/sigterm。
 * http服务里，图片的解码运行在cpu之上，可能会降低推理速度。
 * @tparam ConfigT
 * @param config
 * @param host
 * @param port
 * @param service_id service_id user parameter, uri '/get/service_id' will respond this value with 'text/plain'
 * @param instance_num 实例数量，根据内存/显存/时延要求调整
 * @return
 */
template<typename ConfigT>
int start_http_server(
    const ConfigT &config,
    const std::string &host,
    int port,
    const std::string &service_id,
    int instance_num = 1);

```

## 2. 请求http服务

开发者可以打开浏览器，`http://{设备ip}:24401`，选择图片来进行测试。

http 请求方式一：不使用图片base64格式 URL中的get参数：

参数	说明	默认值
threshold	阈值过滤，0~1	如不提供，则会使用模型的推荐阈值

HTTP POST Body即为图片的二进制内容(无需base64, 无需json)

Python请求示例

```

import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()
    result = requests.post(
        'http://127.0.0.1:24401/',
        params={'threshold': 0.1},
        data=img).json()

```

Java请求示例

http 请求方法二：使用图片base64格式 HTTP方法：POST Header如下：

参数	值
Content-Type	application/json

Body请求填写：

- 分类网络：body 中请求示例

```

{
  "image": "<base64数据>"
  "top_num": 5
}

```

body中参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
top_num	否	number	-	返回分类数量，不填该参数，则默认返回全部分类结果

- 检测和分割网络：Body请求示例：

```
{
  "image": "<base64数据>"
}
```

body中参数详情：

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码，要求base64图片编码后大小不超过4M,最短边至少15px，最长边最大4096px，支持jpg/png/bmp格式 注意去掉头部
threshold	否	number	-	默认为推荐阈值，也可自行根据需要进行设置

http 返回数据

字段	类型说明	其他
error_code	Number	0为成功,非0参考message获得具体错误信息
results	Array	内容为具体的识别结果。其中字段的具体含义请参考预测图像-返回格式一节
cost_ms	Number	预测耗时ms，不含网络交互时间

返回示例

```
{
  "cost_ms": 52,
  "error_code": 0,
  "results": [
    {
      "confidence": 0.94482421875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.059185408055782318,
      "x2": 0.18795496225357056,
      "y1": 0.14762254059314728,
      "y2": 0.52510076761245728
    },
    {
      "confidence": 0.94091796875,
      "index": 1,
      "label": "IronMan",
      "x1": 0.79151463508605957,
      "x2": 0.92310667037963867,
      "y1": 0.045728668570518494,
      "y2": 0.42920106649398804
    }
  ]
}
```

其他配置

日志名称、HTTP 网页标题设置

通过global\_controller的set\_config方法设置：

```
global_controller()->set_config(easyedge::params::KEY_LOG_BRAND, "MY_BRAND");
```

效果如下：



## FAQ

### 1. 如何处理一些 undefined reference / error while loading shared libraries?

如：./easyedge\_demo: error while loading shared libraries: libeasyedge.so.1: cannot open shared object file: No such file or directory 这是因为二进制运行时ld无法找到依赖的库。如果是正确cmake && make 的程序，会自动处理好链接，一般不会出现此类问题。

遇到该问题时，请找到具体的库的位置，设置LD\_LIBRARY\_PATH。

示例一：libverify.so.1: cannot open shared object file: No such file or directory 链接找不到libveirfy.so文件，一般可通过 export LD\_LIBRARY\_PATH=\${LD\_LIBRARY\_PATH}:/lib 解决(实际冒号后面添加的路径以libverify.so文件所在的路径为准)

示例二：libopencv\_videoio.so.4.5: cannot open shared object file: No such file or directory 链接找不到libopencv\_videoio.so文件，一般可通过 export LD\_LIBRARY\_PATH=\${LD\_LIBRARY\_PATH}:/thirdparty/opencv/lib 解决(实际冒号后面添加的路径以libopencv\_videoio.so所在路径为准)

### 2. 如何将我的模型运行为一个http服务？

目前cpp sdk暂未集成http运行方式；0.4.7版本之后，可以通过start\_http\_server方法开启http服务。

### 3. 运行NNIE引擎报permission denied

日志显示：

```
open sys: Permission denied
open err
: Permission denied
open err
: Permission denied
```

请使用sudo在root下运行。

### 4. 运行SDK报错 Authorization failed

1. 情况一：日志显示 Http perform failed: null respond 在新的硬件上首次运行，必须联网激活。

SDK 能够接受HTTP\_PROXY 的环境变量通过代理处理自己的网络请求。如

```
export HTTP_PROXY="http://192.168.1.100:8888"
./easyedge_demo ...
```

2. 情况二：日志显示failed to get/check device id(xxx)或者Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- MAC地址变化
- 磁盘变更
- BIOS重刷

以及系统相关信息。



遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 `~/./baidu/easyedge` 目录，再重新激活。

### 5. 使用libcurl请求http服务时，速度明显变慢

这是因为libcurl请求continue导致server等待数据的问题，添加空的header即可

```
headers = curl_slist_append(headers, "Expect:");
```

### 6. 运行NNIE引擎报错 std::bad\_alloc

检查开发板可用内存，一些比较大的网络占用内存较多，推荐内存500M以上

### 7. 运行二进制时，提示 libverify.so cannot open shared object file

可能cmake没有正确设置rpath，可以设置LD\_LIBRARY\_PATH为sdk的lib文件夹后，再运行：

```
LD_LIBRARY_PATH=$LD_LIBRARY_PATH:~/lib ./easyedge_demo
```

### 8. 编译时报错：file format not recognized

可能是因为在复制SDK时文件信息丢失。请将整个压缩包复制到目标设备中，再解压缩、编译

## 纯离线SDK简介

本文档主要说明定制化模型发布后获得的SDK如何使用，如有疑问可以通过以下方式联系我们：

- 在百度智能云控制台内[提交工单](#)
- 前往[官方论坛](#)交流，与其他开发者进行互动

## SDK说明

SDK支持iOS、Android、Linux、Windows四种操作系统。以下为具体的系统、硬件环境支持：

操作系统	系统支持	硬件环境要求
Linux C++		CPU: AArch64 ARMv7I
Windows	64位 Windows7 及以上	Intel CPU x86_64  环境依赖： .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 Visual C++ Redistributable Packages for Visual Studio 2015

## 激活&使用SDK

SDK的激活与使用分以下步骤：

1. 本地运行SDK，并完成首次联网激活
2. 正式使用

## Windows集成文档

### 简介

本文档介绍Windows CPU SDK的使用方法。

- 网络类型支持：图像分类，物体检测
- 硬件支持：
  - Intel CPU \* x86\_64
- 操作系统支持

- 64位 Windows 7 及以上
- 环境依赖（必须安装以下版本）
  - .NET Framework 4.5
  - Visual C++ Redistributable Packages for Visual Studio 2013
  - Visual C++ Redistributable Packages for Visual Studio 2015

- 协议

- HTTP

Release Notes | 时间 | 版本 | 说明 | |-----|-----|-----| | 2021.8.23 | 1.0.0 | 第一版！ |

## 快速开始

### 1. 安装依赖

#### 安装.NET Framework4.5

<https://www.microsoft.com/zh-CN/download/details.aspx?id=42642>

#### Visual C++ Redistributable Packages for Visual Studio 2013

<https://www.microsoft.com/zh-cn/download/details.aspx?id=40784>

#### Visual C++ Redistributable Packages for Visual Studio 2015

<https://www.microsoft.com/zh-cn/download/details.aspx?id=48145>

### 注意事项

1. 安装目录不能包含中文
2. Windows Server 请自行开启，选择“我的电脑”——“属性”——“管理”——“添加角色和功能”——勾选“桌面体验”，点击安装，安装之后重启即可。

### 2. 运行离线SDK

解压下载好的SDK，打开EasyEdge.exe。

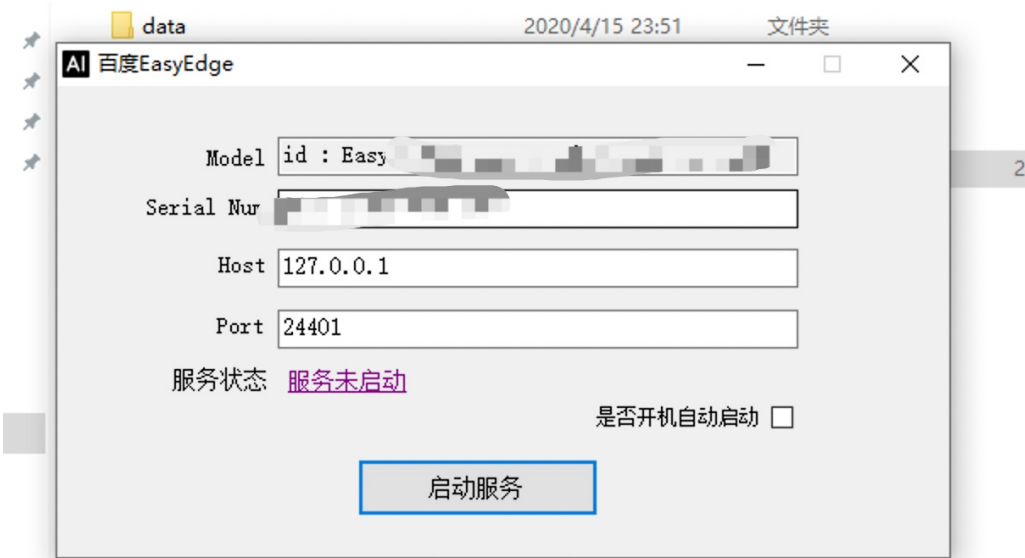


图片加载失败

点击“启动服务”，等待数秒即可启动成功，本地服务默认运行在

<http://127.0.0.1:24401/>

其他任何语言只需通过HTTP调用即可。

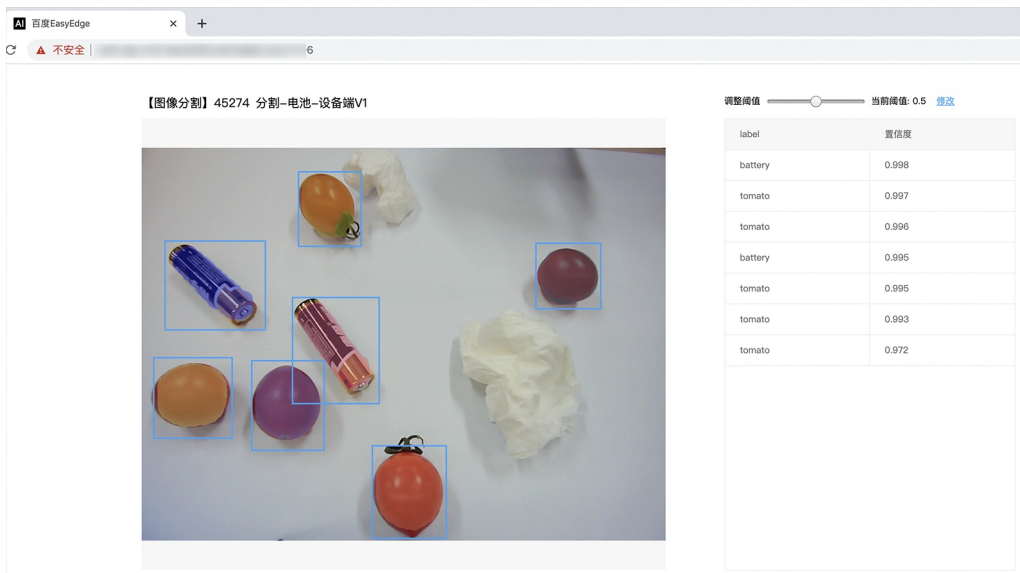


如启动失败，可参考如下步骤排查：



Demo示例(以图像服务为例)

服务运行成功，此时可直接在浏览器中输入http://127.0.0.1:24401，在h5中测试模型效果。



使用说明

图像服务调用说明

Python 使用示例代码如下

```
import requests

with open('./1.jpg', 'rb') as f:
    img = f.read()

**params 为GET参数 data 为POST Body**
result = requests.post('http://127.0.0.1:24401/', params={'threshold': 0.1},
                       data=img.json())
```

#### C# 使用示例代码如下

```
FileStream fs = new FileStream("./img.jpg", FileMode.Open);
BinaryReader br = new BinaryReader(fs);
byte[] img = br.ReadBytes((int)fs.Length);
br.Close();
fs.Close();
string url = "http://127.0.0.1:8402?threshold=0.1";
HttpRequest request = (HttpRequest)HttpRequest.Create(url);
request.Method = "POST";
Stream stream = request.GetRequestStream();
stream.Write(img, 0, img.Length);
stream.Close();

WebResponse response = request.GetResponse();
StreamReader sr = new StreamReader(response.GetResponseStream());
Console.WriteLine(sr.ReadToEnd());
sr.Close();
response.Close();
```

#### C++ 使用示例代码如下，需要安装curl

```

##### include <sys/stat.h>
##### include <curl/curl.h>
##### include <iostream>
##### include <string>
##### define S_ISREG(m) (((m) & 0170000) == (0100000))
##### define S_ISDIR(m) (((m) & 0170000) == (0040000))

size_t write_callback(void *ptr, size_t size, size_t num, void *data) {
    std::string *str = dynamic_cast<std::string*>((std::string *)data);
    str->append((char *)ptr, size*num);
    return size*num;
}

int main(int argc, char *argv[]) {
    const char *post_data_filename = "./img.jpg";
    FILE *fp = NULL;
    std::string response;
    struct stat stbuf = { 0, };
    fp = fopen(post_data_filename, "rb");
    if (!fp) {
        fprintf(stderr, "Error: failed to open file \"%s\"\n", post_data_filename);
        return -1;
    }
    if (fstat(fileno(fp), &stbuf) || !S_ISREG(stbuf.st_mode)) {
        fprintf(stderr, "Error: unknown file size \"%s\"\n", post_data_filename);
        return -1;
    }
    CURL *curl;
    CURLcode res;
    curl_global_init(CURL_GLOBAL_ALL);
    curl = curl_easy_init();
    if (curl != NULL) {
        curl_easy_setopt(curl, CURLOPT_URL, "http://127.0.0.1:24401?threshold=0.1");
        curl_easy_setopt(curl, CURLOPT_POST, 1L);
        curl_easy_setopt(curl, CURLOPT_POSTFIELDSIZE_LARGE, (curl_off_t)stbuf.st_size);
        curl_easy_setopt(curl, CURLOPT_READDATA, (void *)fp);
        curl_easy_setopt(curl, CURLOPT_WRITEFUNCTION, write_callback);
        curl_easy_setopt(curl, CURLOPT_WRITEDATA, &response);
        res = curl_easy_perform(curl);

        if (res != CURLE_OK) {
            fprintf(stderr, "curl_easy_perform() failed: %s\n", curl_easy_strerror(res));
        }
        std::cout << response << std::endl; // response即为返回的json数据
        curl_easy_cleanup(curl);
    }
    curl_global_cleanup();
    fclose(fp);
    return 0;
}

```

结果 获取的结果存储在response字符串中。

#### 请求参数

字段	类型	取值	说明
threshold	float	0 ~ 1	置信度阈值

HTTP POST Body直接发送图片二进制。

返回参数 | 字段 | 类型 | 取值 | 说明 | |-----|-----|----|-----| | confidence | float | 0~1 | 分类或检测的置信度 | | label | string | | 分类或检测的类别 | | index | number | | 分类或检测的类别 | | x1, y1 | float | 0~1 | 物体检测，矩形的左上角坐标（相对长宽的比例值） | | x2, y2 | float | 0~1 | 物体检测，矩形的右下角坐标（相对长宽的比例值） |

#### 关于矩形坐标

x1 \* 图片宽度 = 检测框的左上角的横坐标

y1 \* 图片高度 = 检测框的左上角的纵坐标

x2 \* 图片宽度 = 检测框的右下角的横坐标

y2 \* 图片高度 = 检测框的右下角的纵坐标

FAQ

1. 服务启动失败，怎么处理？

请确保相关依赖都安装正确，版本必须如下：  
 .NET Framework 4.5 Visual C++ Redistributable Packages for Visual Studio 2013 \* Visual C++ Redistributable Packages for Visual Studio 2015

2. 服务调用时返回为空，怎么处理？

调用输入的图片必须是RGB格式，请确认是否有alpha通道。

3. 多个模型怎么同时使用？

SDK设置运行不同的端口，点击运行即可。

4. JAVA、C#等其他语言怎么调用SDK？

参考 <https://ai.baidu.com/forum/topic/show/943765>

5. 启动失败，报错NotDecrypted？

Windows下使用，当前用户名不能为中文，否则无法正确加载模型。

6. 启动失败，报错 SerialNum无效

日志显示failed to get/check device id(xxx)或者 Device fingerprint mismatch(xxx) 此类情况一般是设备指纹发生了变更，包括（但不局限于）以下可能的情况：

- mac 地址变化
- 磁盘变更
- bios重刷

以及系统相关信息。

遇到这类情况，请确保硬件无变更，如果想更换序列号，请先删除 C:\Users\\${用户名}\.baidu\easyedge 目录，再重新激活。

7. 其他问题

如果无法解决，可到论坛发帖：<https://ai.baidu.com/forum/topic/list/199> 描述使用遇到的问题，我们将及时回复您的问题。

AutoDL模式算法适配硬件

🔗 图像分类

服务器

算法类型	Linux										Windows
	通用x86 CPU	英伟达GPU	比特大陆SC计算卡	飞腾CPU	海光DCU	飞腾+华为Atlas300I	x86+寒武纪MLU270	飞腾+百度昆仑XPU-K200	x86+百度昆仑XPU-K200	x86+opervino	英伟达GPU
超高性能	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
高性能	✓	✓	✗	✓	✗	✗	✓	✓	✗	✓	✓
高精度	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

通用小型设备

算法类型	Linux									Windows	
	通用ARM	通用ARM GPU	通用ARM+华为Atlas 310	华为海思N910E	比特大陆SE5计算盒	瑞芯微NPU RK3399Pro	瑞芯微NPU RV1109/RV1126	瑞芯微NPU RK3588	飞腾+比特大陆SC	通用x86 CPU	英特尔Movidius VPU
超高性能	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓
高性能	✓	✓	✓	✓	✗	✓	✓	✓	✗	✓	✗
高精度	✓	✓	✓	✗	✓	✓	✓	✓	✓	✓	✓

专项硬件设备

算法类型	Linux
	Jetson(TX2/Nano/Xavier)
高性能	✔

🔗 物体检测

服务器

算法类型	Linux										Windows
	通用x86 CPU	英伟达GPU	比特大陆SC计算卡	飞腾CPU	海光DCU	飞腾+寒武纪MLU270	x86+寒武纪MLU270	飞腾+百度昆仑XPU-K200	飞腾+百度昆仑XPU-K200	x86+opervino	英伟达GPU
超高性能	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
高性能	✔	✔	✘	✔	✘	✔	✔	✔	✘	✔	✔
高精度	✔	✔	✘	✘	✘	✘	✘	✘	✘	✘	✔

通用小型设备

算法类型	Linux									Windows	
	通用ARM	通用ARM GPU	通用ARM+华为Atlas 310	通用ARM+寒武纪MLU220	比特大陆SE5计算盒	瑞芯微NPU RK3399Pro	瑞芯微NPU RV1109/RV1126	瑞芯微NPU RK3588	飞腾+比特大陆SC	通用x86 CPU	英特尔Movidius VPU
高性能	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
高精度	✔	✔	✔	✘	✘	✔	✔	✔	✘	✔	✘

专项适配硬件

算法类型	Linux
	Jetson(TX2/Nano/Xavier)
高性能	✔

🔗 实例分割

服务器

算法类型	Linux	Linux	Windows
	通用x86 CPU	英伟达GPU	通用x86+ GPU
默认	✔	✔	✔

通用小型设备

算法类型	Windows
	通用x86
默认	✔

🔗 语义分割

服务器

算法类型	Linux	Linux	Windows
	通用x86 CPU	英伟达GPU	通用x86+ GPU
高性能	✔	✔	✔
高精度	✔	✔	✔

通用小型设备

算法类型	Windows
	通用x86
高精度	✔
高性能	✔

高级调参模式算法适配硬件

🔗 图像分类

算法类型	Linux													Windows
	通用x86 CPU	x86+英伟达GPU	通用ARM CPU	通用ARM GPU	x86+寒武纪MLU270	通用ARM+华为昇腾310	飞腾+百度昆仑XPU-K200	飞腾+寒武纪MLU270	比特大陆SC计算卡	瑞芯微NPU RK3399 Pro	瑞芯微NPU RV1109/RV1126	瑞芯微NPU RK3588	英伟达Jetson	x86+英伟达GPU
MobileNet	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
ResNet	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
SENet	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔	✔
EfficientNet	✔	✔	✔	✔	✘	✘	✘	✘	✔	✔	✔	✔	✘	✔
PPCLNet	✔	✔	✔	✔	✘	✘	✘	✘	✔	✔	✔	✔	✔	✔
ResNeXt	✔	✔	✔	✔	✘	✔	✔	✘	✔	✘	✔	✔	✘	✔

🔗 物体检测

算法类型	Linux													Windows
	通用x86 CPU	x86+英伟达GPU	通用ARM CPU	通用ARM GPU	x86+寒武纪MLU270	通用ARM+华为昇腾310	飞腾+百度昆仑XPU-K200	飞腾+寒武纪MLU270	比特大陆SC计算卡	瑞芯微NPU RK3399 Pro	瑞芯微NPU RV1109/RV1126	瑞芯微NPU RK3588	英伟达Jetson	x86+英伟达GPU
CaradeRCNN	✔	✔	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
FastRCNN	✔	✔	✔	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘	✘
YOLOv3	✔	✔	✔	✔	✔	✔	✔	✘	✔	✔	✔	✔	✔	✔
PicoDet	✔	✔	✔	✔	✘	✔	✔	✘	✔	✘	✘	✔	✔	✔

🔗 实例分割

算法类型	Linux		Windows	
	通用x86芯片	x86+英伟达GPU	通用ARM芯片	x86+英伟达GPU
MaskRCNN	✓	✓	✓	✓

## 语义分割

算法类型	Linux		Windows	
	通用x86芯片	x86+英伟达GPU	通用ARM芯片	x86+英伟达GPU
DeepLabV3	✓	✓	✓	✓
BiSeNetV2	✓	✓	✓	✓
DeepLabV3p	✓	✓	✓	✓
OCRNet	✓	✓	✓	✓
Fast-SCNN	✓	✓	✓	✓
PP-LiteSeg	✓	✓	✓	✓
STDCSeg	✓	✓	✓	✓

## 常见问题

### 训练相关问题

#### 如果使用GPU环境进行训练？

如果您的计算机有NVIDIA® GPU，且需要使用GPU环境进行训练，请确保满足以下条件：

- Windows 10/11：需安装 CUDA 12.0 与 cuDNN v8.9.0
- Ubuntu 18.4/20.4：需 CUDA 12.0 与 cuDNN v8.9.0

您可参考NVIDIA官方文档了解CUDA和CUDNN的安装流程和配置方法，详见：

- CUDA 安装指南：<https://docs.nvidia.com/cuda/>
- cuDNN安装指南：<https://docs.nvidia.com/deeplearning/cudnn/install-guide/index.html>

#### 训练失败怎么办？

若训练失败，训练任务列表中会展示失败原因及解决办法，如有其他问题请[提交工单](#)联系我们。

### 模型效果相关问题

#### 如何提升模型效果？

一个模型很难一次性就训练到最佳的效果，可能需要结合模型评估报告和校验结果不断扩充数据和调优。为此我们设计了模型迭代功能，即当模型训练完毕后，会生成一个最新的版本号，首次V1、之后V2……以此类推。可以通过调整训练数据和算法，多次训练，获得更好的模型效果。

若想要提升模型效果，可以尝试以下两种方法：

#### 一、检查并优化训练数据

##### 图像分类

1. 检查是否存在训练数据过少的情况，建议每个类别的图片量不少于100个，如果低于这个量级建议扩充。
2. 检查不同类别的数据量是否均衡，建议不同分类的数据量级相同，并尽量接近，如果有的类别数据量很高，有的类别数据量较低，会影响模型整体的识别效果。
3. 通过模型效果评估报告中的错误识别示例，有针对性地扩充训练数据。
4. 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

##### 物体检测

1. 检查是否存在训练数据过少的情况，建议每个标签标注50个框以上，如果低于这个量级建议扩充。
2. 检查不同标签的标注框数量是否均衡，建议不同标签的标注框数数据量级相同，并尽量接近，如果有的标签框数很多，有的标签框数很少，会影响模型整体的识别效果。
3. 通过模型效果评估报告中的错误识别示例，有针对性地扩充训练数据。
4. 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测



试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

### 实例分割

1. 检查是否存在训练数据过少的情况，建议每个标签标注50个目标以上，如果低于这个量级建议扩充。
2. 检查不同标签的标注目标数是否均衡，建议不同标签的标注目标数数据量级相同，并尽量接近，如果有的标签标注的很多，有的标签标注的很少，会影响模型整体的识别效果。
3. 通过模型效果评估报告中的错误识别示例，有针对性地扩充训练数据。
4. 检查测试模型的数据与训练数据的采集来源是否一致，如果设备不一致、或者采集的环境不一致，那么很可能会存在模型效果不错但实际测试效果较差的情况。针对这种情况建议重新调整训练集，将训练数据与实际业务场景数据尽可能一致。

## 二、尝试不同的训练配置

可前往训练配置页面尝试不同的配置组合，因不同数据集在不同的算法上可能表现不一致，所以建议您多尝试不同的算法选型后综合挑选精度最高的模型使用，你可以选择如下的配置项：

- 增量训练
- 自定义验证集
- 数据增强策略
- 在高级训练配置中增加输入图片分辨率

# 分享我的模型

## 分享EasyDL定制化模型

本文将引导您将定制化模型售卖至AI市场中，具体的介绍与操作流程，您可以详细查看如下流程。

1. [售卖市场介绍](#)
2. [模型售卖的准备工作](#)
3. [模型售卖的具体流程](#)

### 🔗 售卖市场介绍

您在EasyDL定制化训练出的模型可以在百度AI市场 ([aim.baidu.com](http://aim.baidu.com)) 进行售卖。AI市场作为国内首家专注于服务AI产业链的商业平台，集结AI服务商所提供的AI软硬件、解决方案、数据服务、模型算法等产品与服务，为需方提供一站式AI采购平台，帮助供需双方在市场内建立精准的对接与交易通道。

如果您是**企业**，您可以获取模型销售收入，价格由您自主决定；如果您是**个人**，您也可通过发布0元模型商品获得社区积分奖励，商品购买一次将获得1分，50积分即可兑换大礼 ([社区积分获取兑换详情](#))。

您将模型售卖至百度AI市场后，对您模型感兴趣的买家可以在市场中一键下单购买，并对模型进行再训练或是部署为服务集成在自己的应用中。模型在AI市场上架、购买以及买家再训练，均不影响您对原始模型的操作，您可以放心售卖。

### 🔗 模型售卖的准备工作

您的模型在AI市场出售需要具备以下条件

- 满足AI市场服务商入驻标准：
  1. 符合国家相关法律、法规规定
  2. 完成百度云账号个人或企业实名认证
  3. 接受《[百度AI市场入驻协议](#)》及相关协议及管理规范
- 定制化模型服务满足售卖要求：
  1. 模型分类属于图像检测、图像分类或情感倾向分析才可予以售卖。并且图像分类、情感倾向分析类只可售卖高精度算法模型。
  2. 模型最新训练时间在2020.09.25后，且距今不超过1年（模型商品超过训练时间1年，在AI市场将自动下架）。

3. 模型完全由您自主训练，基于已购模型或预置模型再次训练的模型不可售卖。

## 🔗 模型售卖的具体流程

您确认符合AI市场服务商入驻标准，且有符合售卖条件的模型，可按如下步骤完成售卖。期间有任何问题，可发邮件至Almarket@baidu.com咨询，我们会尽快与您联系，解答您的疑问，帮您顺利完成。

售卖具体流程如下：

### 🔗 1、入驻AI市场并成功开店

售卖模型需要在AI市场开店成为服务商，请您按照如下流程操作：

1. 选择身份：根据您的情况，选择以企业或个人身份入驻市场。注：一经选择后不可修改
2. 实名认证：请以模型训练的账号完成认证。企业认证需提交营业执照照片，个人认证需准备身份证照片，请提前准备。
3. 开通店铺：您需提交店铺logo、店铺简介、联系方式等基本信息。通常审核需要1-2个工作日，完成审核后可进入下一步。

### 🔗 2、选择模型出售

您开店成功后，有两种方案选择模型予以出售。您可以根据您的需要，选择任一种方案予以出售。

(1) 在EasyDL的AI市场-售卖模型中，选择可售模型列表中的模型，点击“出售模型”后将会跳转至AI市场的商品列表中，商品列表中将会出现一个已经创建好的草稿，该草稿部分信息已自动填充完成，您需要点击编辑对其余信息予以补充。

**模型中心**

我的模型

创建模型

训练模型

校验模型

发布模型

EasyData数据服务

数据总览

在线标注

云服务数据回流

AI市场

我的已购模型

**售卖模型**

可售模型 (1) 在售模型 (0)

模型选择: [wtb]美女和野兽 (ID: 34660) 可售模型标准

部署方式: 公有云API

训练版本: V5

出售模型 商品管理

**售卖模型收益是什么?**

您可将已完成训练的模型上架到AI市场 (aim.baidu.com) 作为服务商售卖您的模型。

如果您是个人，您可以通过发布0元模型商品获得社区积分奖励，商品购买一次将获得1分，50积分即可兑换大礼包 [社区积分获取兑换详情](#)

多渠道免费曝光，有机会作为优质案例获得专属推广

**售卖模型前需要做哪些准备?**

售卖模型需要在AI市场开店成为服务商，请您按照如下流程操作：

1. 选择身份：根据您的情况，选择以企业或个人身份入驻市场。注：一经选择后不可修改
2. 实名认证：请以模型训练的账号完成认证。企业认证需提交营业执照照片，个人认证需准备身份证照片，请提前准备。
3. 开通店铺：您需提交店铺logo、店铺简介、联系方式等基本信息。

(2) 在AI市场服务商后台的商品列表中选择“商品发布”，在商品分类中选择模型算法以及对应的模型分类，右侧出现从“从现有模型发布”按钮，点击后选中您需要发布的模型。

< 返回商品列表 发布商品

基本信息

商品名称: 名称不可重复使用，要求与服务紧密相关，建议以公司简称作为前缀名，2-32个字

商品分类: 模型算法 / 图像分类 从现有模型发布

AI能力:

应用行业:

商品摘要:

商品推荐语:

**快速发布**

您的账号在EasyDL经典版中有如下模型可售卖，一键勾选，快速发布。模型训练完成1年内均可在AI市场售卖，到期将会自动下架，重训后即可重新发布。

模型名称	部署方式	训练版本	训练完成时间
[wtb]美女和野兽	公有云API	V5	2020-09-27 20:45:14

发布 取消

注意：如果您有多个模型但无可售模型，可能原因是您的模型最新训练时间在2020.9.25之前或是模型训练时间超过1年，建议您重新训练模型再予以售卖。

### 🔗 3、填写模型商品信息

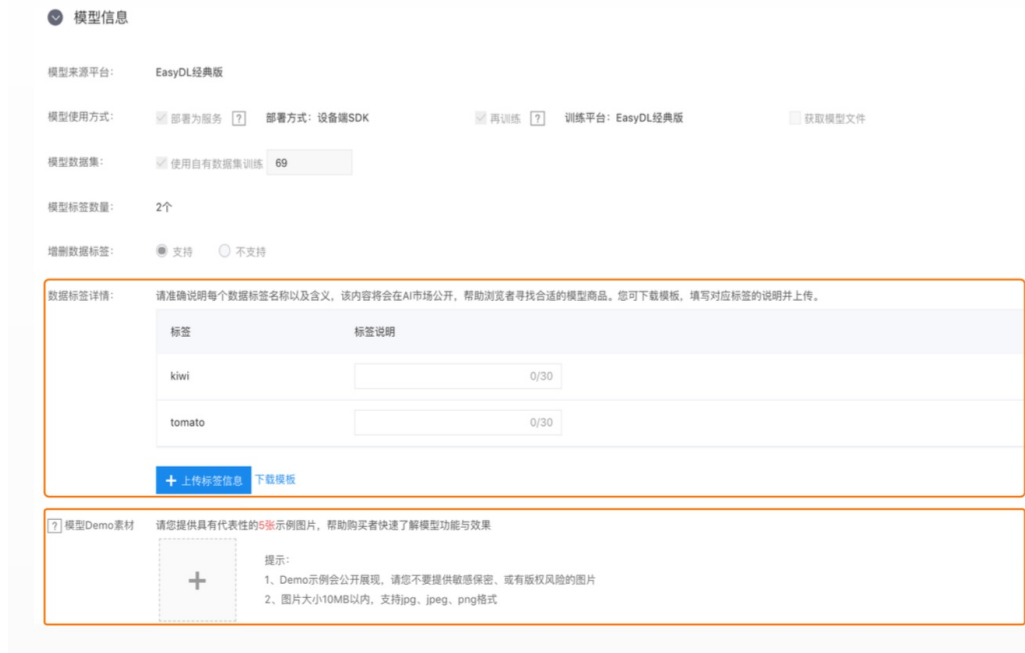
模型商品售卖需要填写四个部分信息：

- **基本信息**：部分信息已为您默认填充，您还需完成应用行业、商品摘要、商品推荐语、商品图片、产品亮点、产品说明、商品协议等内容予以填写，图片内容越丰富越直观表达模型应用场景，越容易获得买家的青睐。其中商品协议为您与用户之间的协议，AI市场只提供协议模板，您在此基础上修改完成上传PDF，买家需要同意后才可下单购买。



- **模型信息**：模型信息包括模型使用方式、数据标签信息、模型demo素材等信息，您需要重点关注数据标签信息与模型demo素材信息的填写。
  - 数据标签：需要您准确说明每个标签的名称与含义，这将帮助买家理解模型用途并基于此进行模型再训练。
  - 模型demo素材：为了促进模型售卖，AI市场为当前上架的模型免费提供Demo试用服务。因此您需要为Demo提供5张示例图片与文本，方便

买家试用模型效果。



- **交付信息**：AI市场已为您默认填写好规格信息。对于企业服务商，您的规格信息包括：10天免费试用版与正式版，试用版价格为0，正式版您可以自主定价；对于个人服务商，您只有一个正式版模型规格，价格为0，但模型一旦售卖将会获得社区积分，可兑换精美礼品。

**交付信息**

**规格信息:**

10天免费试... 正式版模型

规格名称:

规格说明: 

自模型交付日起10天内有效, 仅可支持模型再训练查看效果, 不可部署为服务, 到期后模型失效。

交易方式: 线上交易

交付类型: 模型交付

交付时间: 即时交付

定价方式: 枚举定价

商品单位: 个

含税价格: 免费

有效期: 10天

**规格信息:**

10天免费试... 正式版模型

规格名称:

交易方式: 线上交易

交付类型: 模型交付

交付时间: 即时交付

定价方式: 枚举定价

商品单位: 个

含税价格:  元/个

有效期: 1年

设置展示原价, 商品发布后的价格展示效果

- **售后信息:** 售后信息包括发票与服务保障。对于企业服务商, 您需要为买家提供增值税普票或专票; 对于个人服务商, 由于商品为零元, 您不需要提供发票。服务保障说明已默认为您填写好, 无需修改。

**售后服务信息**

发票服务:  仅支持开具增值税专用发票(可抵扣)  仅支持开具增值税普通发票(不可抵扣)  前两项都支持

服务保障:  质保期  年  无质保  终身质保

保障说明: (选填)

1、模型类商品不支持退款, 请购买前充分了解模型效果。

2、10天免费试用版在购买成功时生效, 生效期内可再训练或将模型部署为服务。

3、正式版模型购买后可在EasyDL经典版平台训练与部署。模型有效期为一年, 一年内您可以对购买原始模型进行再训练(但不保证EasyDL新功能的可用性)。有效期结束后, 您已部署的服务不会强制下线, 您可继续使用。

4、模型购买后不会自动升级或更新, 如需使用升级版模型, 请重新购买。

203/300

商品信息填写完整后即可提交审核。提交后, 将会由专人进行质量审核, 约一个工作日给您结果反馈,

#### 4、售卖并获得收入

您的模型服务审核通过并成功上架后, AI市场的买家将可在市场中购买您的模型服务。成功付款后, 收入将由AI市场代收, 您可在月底发起结算申请, AI市场将销售收入打到您的账户中。

## 分享EasyDL定制化API

本文将引导您将定制化API服务售卖至AI市场中, 从而快速获得品牌曝光, 并获得丰厚收入。接下来, 您将了解如下内容:

1. [售卖市场介绍](#)
2. [API服务售卖的准备工作](#)
3. [API服务售卖的具体流程](#)

#### 售卖市场介绍

您在EasyDL定制化训练出的模型可以在百度AI市场 ([aim.baidu.com](http://aim.baidu.com)) 进行售卖。AI市场作为国内首家专注于服务AI产业链的商业平台, 集结AI服务商所提供的AI软硬件、解决方案、数据服务、模型算法等产品与服务, 为需方提供一站式AI采购平台, 帮助供需双方在市场上建立精准的对接与交易通道。

您将训练完成的API售卖至AI市场中, 不仅会获得大量浏览用户关注, 增强您企业的品牌曝光。同时, 您所训练的服务还将发挥巨大的商业价值, 对您服务感兴趣的买家可以在市场中一键下单购买, 并集成到业务中使用, 您也将获得相应的收入。

#### API服务售卖的准备工作

您的服务在AI市场出售定制化API服务需要具备以下条件

- 满足AI市场服务商入驻标准：
  1. 符合国家相关法律、法规规定，拥有正规的公司资质
  2. 完成百度云账号企业实名认证
  3. 接受《[百度AI市场入驻协议](#)》及相关协议及管理规范
- 定制化API服务满足售卖要求：
  1. 定制化识别接口至少被成功调用1次及以上
  2. 图像分类/声音分类模型准确率>80%，且总训练数据量>50
  3. 物体检测模型mAP>60%，且总训练数据量>20

## 🔗 API服务售卖的具体流程

您确认您的企业/单位符合AI市场服务商入驻标准，且有符合售卖条件的API服务，可按如下步骤完成售卖。期间有任何问题，可发邮件至 [Almarket@baidu.com](mailto:Almarket@baidu.com) 咨询，我们会尽快与您联系，解答您的疑问，帮您顺利完成。售卖具体流程如下：

### 🔗 1、入驻AI市场并成功开店

售卖模型需要在[AI市场开店](#)成为服务商，请您按照如下流程操作：

1. 实名认证：请以EasyDL训练者的账号完成企业实名认证。企业认证需提交营业执照照片，请提前予以准备。
2. 开通店铺：您需提交店铺logo、店铺简介、联系方式等基本信息。通常审核需要1-2个工作日，完成审核后可进入下一步。

### 🔗 2、选择API出售

您开店成功后，有两种方案选择API予以出售。您可以根据您的需要，选择任一种方案予以出售。

(1) 在EasyDL控制台-公有云服务-售卖服务中，选择可售API列表中的API，点击“发布商品”后将会跳转至AI市场商品发布页面，部分API信息将会予以填充。

产品服务 / EasyDL定制训练平台 - 售卖已发布服务

经典版

公有云服务

- 应用列表
- 权限管理
- 用量统计
- 监控报表
- 技术文档
- 售卖服务**

私有服务器部署服务

通用设备请购服务

专项硬件适配服务

售卖已发布服务

已发布服务售卖条件及售卖流程说明 [收起教程](#)

售卖位置：模型发布成功后，将已上线的定制化识别接口发布至 [百度AI市场](#) 进行售卖。

售卖条件：

- 定制化识别接口成功调用次数≥1；
- 图像分类/声音分类模型准确率>80%，总训练数据量>50；
- 物体检测模型mAP>60%，总训练数据量>20；
- 文本分类模型满足top1准确率>95%，文本数据量超过2万条。

售卖流程：

入驻AI市场 → 创建店铺 签订协议 → 发布商品 → 售卖获得收入

序号	接口名称	应用类型	模型ID	模型名称	模型类型	售卖状态	操作
1	邦邦得劲检测		8160	吴拓邦的物体检测2	物体检测	未发布	<a href="#">发布商品</a>
2	邦邦的分类		8111	吴拓邦的图像分类2	图像分类	未发布	<a href="#">发布商品</a>

(2) 在AI市场服务商后台的商品列表中选择“商品发布”，在商品分类中选择软件服务-API接口，右侧出现从“从现有API发布”按钮，点击后选中您需要发布的API。

### 🔗 3、发布API商品

商品发布需要填写四个部分信息：

- **商品基本信息**：注意：由于EasyDL服务的限制，买家使用的最大并发限制为2QPS，您可在商品信息中提醒买家，API服务暂无法支持高并发的需求。
- **API生产信息**：API生产信息主要包括请求参数、请求示例、返回示例以及错误码。由于本部分信息对于买家调用API非常重要，请谨慎填写。具体内容您可按照训练模型的类别，参考EasyDL官方提供的[定制化声音识别API文档](#)进行填写。
- **交付信息**：选择线上交易，创建规格。在线售卖API已默认添加0元/20次(买家只能购买一次)的体验版规格，方便买家进行简单测试，您可以根据需要自行删除或添加套餐规格，例如配置套餐包：配额100次调用，价格10元。
- **售后信息**：售后信息包括发票与服务保障。对于企业服务商，您需要为买家提供增值税普票或专票，服务保障请您按照实际情况填写。

商品信息填写完整后即可提交审核。提交后，将会由专人进行质量审核，约一个工作日给您结果反馈，审核期间，AI市场将会测试调用一次，来确定您的API是否符合售卖标准。

#### 4、售卖并获得收入

您的API服务审核通过并成功上架后，AI市场的买家将可在市场中购买您的服务。成功付款后，收入将由AI市场代收，您可在月底发起结算申请，AI市场将销售收入打到您的账户中。

## 版本更新记录

### 2022年12月

序号	功能模块	功能描述
1	模型训练	EasyDL图像-物体检测模型训练新增「半监督学习」功能，支持添加已标注数据和未标注数据共同参与训练，从而获得一个泛化效果更好的模型

### 2022年09月

序号	功能模块	功能描述
1	模型评估	EasyDL图像-图像分类、物体检测模型评估功能新增「模型调优建议」功能,建议将从「基于整个模型」和「基于单个标签」两个角度，给出数据中各因素（对比度、标注框面积等）对模型准确率、漏检率、误检率三个方面造成的影响程度，并提出修改策略建议。
2	模型训练	EasyDL图像-实例分割新增支持「高性能」算法,性能相较高精度算法提升一倍。
3	模型训练	EasyDL文本-文本创作后端升级文心大模型，从原来的仅提供10B模型调整为高精度（10B）、高性能（1.5B）两类算法，10B模型在效果方面具有2%领先优势，1.5B模型在性能方面具有较大优势，预计1.5B模型训练资源减少到原有的1/8（10B模型训练需要8张v100卡加载，1.5B模型只需要1张v100），预测资源预估约减少为1/3。
4	模型服务	EasyDL结构化数据-表格数据预测新增支持「本地服务器/通用小型设备x86 CPU部署」，满足用户对离线部署的场景需求。
5	产品优化	EasyDL我的模型页面支持模型搜索、模型置顶功能，进一步优化用户模型管理使用体验。

### 2022年07月

序号	功能模块	功能描述
1	模型服务	EasyDL图像-图像分类、物体检测新增「辨影软硬一体方案」，针对图像分类、物体检测操作台在训练模型时支持选择辨影作为专项适配硬件的选项之一，通过发挥辨影在工业质检场景灵活部署、自带显示屏等业务优势，提高用户本地部署的易用性。
2	模型训练	EasyDL图像-图像分类、物体检测新增支持「超高性能」算法，性能提升1.2倍。
3	模型服务	EasyDL图像-公有云服务支持接口输入为图片URL地址，优化用户服务调用使用体验。
4	模型服务	EasyDL文本-文本创作支持在线H5体验模型推理，优化用户在模型体验阶段的使用体验。
5	产品优化	创建模型数量上调至30，进一步满足用户创建多个模型的开发诉求。

### 2022年06月

序号	功能模块	功能描述
1	场景应用	EasyDL新增支持「场景范例」，包含工业制造、智慧城市、电商等多行业场景算法，辅助用户快速上手体验AI开发流程。

### 2022年05月

序号	功能模块	功能描述
1	模型种类	EasyDL新增跨模态技术方向上线「图文匹配」任务类型，满足用户在图文匹配相关场景的应用诉求，例如，知识文档、图谱构建等。
2	场景应用	EasyDL图像-物体检测新增人脸检测、行人检测、车辆识别3类场景算法能力，进一步满足用户在细分场景对高精度模型的诉求。
3	模型训练	EasyDL图像-物体检测高级训练配置新增支持「数据不平衡优化」策略，解决用户在数据收集阶段因为样本分布数量不均导致模型精度下降的问题。

🕒 2022年04月

序号	功能模块	功能描述
1	模型种类	EasyDL图像-图像分割新增模「语义分割」模型类型，充分解决用户在识别外观形状有细分差异场景下的识别难题。

🕒 2022年03月

序号	功能模块	功能描述
1	模型种类	EasyDL文本新增「评论观点抽取」任务类型，多维度收集用户的文本评论观点，常应用于电商购物、政府信箱等业务场景。

🕒 2022年01月

序号	功能模块	功能描述
1	模型服务	EasyDL图像-图像分类&物体检测支持批量预测，支持用户异步获取预测结果的诉求，同时帮助用户进一步利用资源空闲期降低预测成本。

🕒 2021年10月

序号	功能模块	功能描述
1	模型训练	EasyDL图像-图像分类支持免训练迭代模式，当用户的训练数据存在不断更新的情况，可开启该模式，快速添加新数据至数据库直接迭代模型，不必重新训练，降低训练成本。
2	模型训练	EasyDL图像-物体检测支持自定义四边形标注和训练，充分满足用户在复杂标注场景下的物体检测标注诉求，提高模型效果精度。

🕒 2021年09月

序号	功能模块	功能描述
1	模型部署	EasyDL图像-上线智能边缘控制台，充分满足用户在本地管理预测设备、实现服务包统一更新迭代的诉求。

🕒 2021年08月

序号	功能模块	功能描述
1	模型种类	EasyDL图像-物体检测支持小目标检测，解决当目标物极小时识别率不佳的复杂场景问题。
2	模型训练	EasyDL图像-支持增量训练任务
3	数据服务	EasyDL图像-图像分割支持自动识别标注

🕒 2021年07月

序号	功能模块	功能描述
1	模型训练	EasyDL图像支持自动超参搜索

🕒 2021年06月

序号	功能模块	功能描述
1	模型种类	EasyDL视频-目标跟踪支持多标签模型
2	数据服务	EasyDL视频-目标跟踪支持在线标注
3	数据服务	EasyDL文本-实体抽取支持智能标注
4	数据服务	EasyDL图像上线噪声样本挖掘

#### 🕒 2021年05月

序号	功能模块	功能描述
1	模型种类	EasyDL文本技术方向新增多语种文本分类模型
2	模型种类	EasyDL的本地服务器API已支持线上购买
3	模型训练	EasyDL图像分类高级训练配置支持数据不平衡优化
4	模型训练	EasyDL图像支持精度提升配置包
5	模型训练	EasyDL图像支持自定义验证集&自定义测试集

#### 🕒 2021年03月

序号	功能模块	功能描述
1	模型种类	EasyDL视频上线目标跟踪
2	模型种类	EaysDL OCR全新上线

### 2021年01月

序号	功能模块	功能描述
1	模型种类	EasyDL结构化数据上线时序预测模型
2	数据服务	图像数据导入新增支持COCO格式、导出支持VOC、COCO格式
3	数据服务	在大图标注模式下，提供无损压缩的快速浏览模式
4	数据服务	图像分类批量标注
5	数据服务	图像分割支持导入已标数据
6	模型训练	模型训练支持配置epoch、输入图片分辨率等高级参数
7	模型评估	图像分类模型的混淆矩阵分析，支持查看热力图
8	模型部署	支持端云协同服务
9	模型部署	设备端SDK新增支持Android平台高通骁龙GPU、Linux平台瑞芯微NPU
10	模型部署	软硬一体方案新增均衡算法，提供在精度和性能上更加平衡的算法选择

#### 🕒 2020年12月

序号	功能模块	功能描述
1	数据服务	校验模型页面查看检测框，支持按置信度排序,在物体检测框数量较多时查看结果体验更佳
2	模型管理	支持模型名称修改

#### 🕒 2020年11月

EasyDL平台全新升级，包含图像、文本、语音、OCR、视频、结构化数据6大方向，及零售行业版，覆盖更多应用场景

序号	功能模块	功能描述
1	模型种类	上线短文本相似度任务
2	数据服务	标签格式支持中文
3	数据服务	标注框支持自定义颜色，优化图像分割/物体检测标注模板下用户的标注框浏览体验

#### 🕒 2020年9月



序号	功能模块	功能描述
1	模型种类	支持更丰富的技术方向/任务类型：文本实体抽取、语音识别、结构化数据分析，开放目标跟踪邀测
2	模型训练	接入AI市场，支持用户交易模型，并基于购买的模型再训练
3	模型部署	图像分类模型，支持适配比特大陆的设备端SDK
4	模型训练	物体检测模型，新增「超高精度」、「均衡」两种算法
5	模型性能	图像分类、检测、分割模型，支持在训练页面查看算法的适配硬件及性能，方便选择算法
6	模型效果	文本分类单标签模型，后端框架接入文心，支持高精度与高性能两个算法
7	模型部署	文本单标签、多标签、情感倾向分析模型，支持私有API部署
8	模型部署	支持在线购买软硬一体方案专用SDK、按产品线鉴权设备端SDK授权

## ☞ 2020年6月

序号	功能模块	功能描述
1	模型训练	图像分类、物体检测模型训练时，支持配置数据增强算子
2	模型部署	物体检测模型支持Atlas系列硬件，包括：设备端华为Atlas 200开发板、服务器端Atlas 300加速卡

## ☞ 2020年5月

序号	功能模块	功能描述
1	模型部署	JetsonNano软硬一体方案上线

## ☞ 2020年4月

序号	功能模块	功能描述
1	数据服务	图像分类支持在线标注
2	模型部署	图像分类支持量化加速，提高端部署性能

## ☞ 2020年3月

序号	功能模块	功能描述
1	模型部署	EdgeBoard(VMX)软硬一体方案上线
2	模型部署	新增声音分类服务器端SDK
3	模型部署	图像分类设备端基础版SDK，支持Linux系统Atlas200开发板

## ☞ 2019年12月

序号	功能模块	功能描述
1	模型效果	物体检测设备端SDK部署高精度算法精度进一步提升，平均精度提升5%
2	数据服务	nlp-序列标注功能上线
3	模型效果	nlp方向支持“文心2.0”预训练模型

## ☞ 2019年8月

序号	功能模块	功能描述
1	模型种类	图像分类本地部署训练新增高精度算法
2	模型效果	物体检测模型效果优化
3	模型种类	EasyDL新增图像分割模型

### 🔗 2019年7月

序号	功能模块	功能描述
1	模型效果	物体检测高性能模型平均准确率提升20%
2	模型性能	物体检测高性能模型后端时延降低90%，约500ms

### 🔗 2019年6月

序号	功能模块	功能描述
1	数据服务	EasyDL智能标注功能

### 🔗 2019年5月

序号	功能模块	功能描述
1	模型种类	EasyDL定制视频分类上线

### 🔗 2019年4月

EasyDL零售行业版上线

序号	功能模块	功能描述
1	数据服务	物体检测支持多人同时标注数据集
2	模型部署	物体检测离线SDK新增支持windows及linux操作系统

## 常见问题

### 常见问题

#### 🔗 价格常见问题

模型训练如何计费？

- EasyDL提供各个技术方向均提供免费训练的模式，如需使用更高级付费的资源，请开通付费后使用

图像分类、物体检测、图像分割API如何收费？调用量不够怎么办？

- 每个公有云API有累计10000点的免费调用额度，如需付费使用，请在[控制台](#)进行线上购买

#### 🔗 数据相关问题

需要上传多少张图片才能训练出效果较好的模型？

- 每个分类至少需要准备20张以上。如果想要较好的效果，建议每个分类准备不少于100张图片。

上传图片的总量有限制吗？

- 每个账号下所有数据集的图片总数不能超过200万张。

智能标注功能目前已对图像分类、物体检测、图像分割模型开放，[了解功能详情](#)

以下为智能标注相关常见问题

“一键标注”和“立即训练”要如何选择？

- 当系统推荐“立即训练”，且系统预标注的框确实已非常精准时，可以不用标注剩余数据，直接开始模型训练。此时，仅用当前已标注图片训练的模型，与标注所有数据后训练的模型相比，效果几乎等同
- 如果系统预标注的框还有些不精准，可以启动一键标注，人工确认系统标注的标注框后，再开始训练

### 选择了“立即训练”之后是否还可以“一键标注”？

- 选择“立即训练”之后，系统默认为您结束此次智能标注
- 再次启动智能标注后，您可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

### 智能标注结束后，又往数据集上传了新图片，是否可以直接“一键标注”新图片？

- 如果您创建了新的标签、或新上传的图片场景和之前的图片场景差异较大，建议不要使用一键标注，而是从头开始智能标注（即再次筛选关键图片）
- 如果不是以上情况，再次启动智能标注后，可以通过以下方式进行一键标注：
  - 根据系统提示，进入一键标注
  - 查看系统对“未标注[优先]”图片的预标注，点击“满意预标注结果”后，进入一键标注

### 智能标注中可以增删标签吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前就创建好所有需要识别的标签
- 如果确实需要增删标签，可以先结束智能标注

### 智能标注中可以增删图片吗？

- 暂不支持。为了保证系统智能标注的效果，建议在启动功能前上传需要标注的所有图片，并删除不相关的图片
- 如果确实需要增删图片，可以先结束智能标注

### 智能标注中可以修改已标注图片的标注框吗？

- 可以。但为了保证智能标注的效果，建议不要大量改动
- 如果确实需要修改大量标注，建议先结束智能标注

### 为什么我已经人工标注了很多图片，但系统预标注依然不准？

- 系统预标注的结果会受以下因素影响：
  - 智能标注期间，对“已标注”图片的标签进行大量改动
  - 曾结束智能标注，并对标签、图片进行增删
- 如果您没有进行以上操作，系统标注结果依然不理想，请在百度智能云控制台内[提交工单](#)反馈

### 多个数据集是否可以同时启动智能标注？

- 目前每个账号同一时间仅支持对一个数据集启动智能标注

### 共享中的数据集是否可以启动智能标注？

- 暂不支持。智能标注中的数据集也暂不支持共享，如有疑问请在百度智能云控制台内[提交工单](#)反馈

### 智能标注失败了怎么办？

- 可以先尝试稍后重新启动
- 若再次遇到问题，请在百度智能云控制台内[提交工单](#)反馈

### 图像分割模型如何正确标注？

- 所有图片中出现的目标物体都需要被标出（标注可以重叠）
- 标注应包含整个物体，且尽可能不要包含多余的背景
- 如果图片中存在很多相同标签的目标物体，可以使用右侧的锁定按钮。锁定标签后，只需要在左侧标注目标物体即可，不用再重复选择标签

### 数据处理失败或者状态异常怎么办？

- 如是是图像分类模型上传处理失败，请先检查已上传的分类命名是否正确，是否存在中文命名、或者增加了空格；然后检查下数据图片量是否超过上限（每个账户下200万张）；再检查图片中是否有损坏。如果自查没有发现问题，请在百度智能云控制台内[提交工单](#)反馈

### 模型训练失败怎么办？

- 如果遇到模型训练失败的情况，请在百度智能云控制台内[提交工单](#)反馈

### 已经上线的模型还可以继续优化吗？

- 已经上线的模型依然可以持续优化，操作上还是按照标准流程在训练模型中-选择要优化的模型和数据完成训练，然后在模型列表中更新线上服务，完成模型的优化

## 🔗 模型效果相关问题

### 如何通过「完整评估结果」里的错误示例优化模型？

- 错误示例中，左侧是正确的结果，右侧是模型的识别结果
- 观察模型识别有误的图片有哪些共同点，并有针对性地补充训练数据。比如：当图片比较亮的时候模型都能识别正确，但比较暗的时候模型就识别错了。这时就需要补充比较暗的图片作为训练数据

### 我的数据有限，如何优化效果？

- 先申请发布模型，并备注说明希望通过云服务数据管理功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

### 实际调用服务时模型效果变差？

- 训练图片和实际场景要识别的图片拍摄环境应一致，举例：如果实际要识别的图片是摄像头俯拍的，那训练图片就不能用网上下载的目标正面图片
- 每个标签的图片需要覆盖实际场景里面的可能性，如拍照角度、光线明暗的变化，训练集覆盖的场景越多，模型的泛化能力越强
- 如果使用的是云服务，可以开通云服务数据管理功能，将实际调用云服务识别的图片加入训练集，不断迭代模型

\*\*如果训练数据已经达到以上要求，且单个分类/标签的图片量超过200张以上，效果仍然不佳，请在百度智能云控制台内[提交工单](#)反馈

## 🔗 模型上线相关问题

### 希望加急上线怎么处理？

- 请在百度智能云控制台内[提交工单](#)反馈

### 每个账号可以上线几个模型？是否可以删除已上线的模型？

- 每个账号最多申请发布十个模型，已上线模型无法删除

### 申请发布模型审核不通过都是什么原因？

- 可能原因有，1、经过电话沟通当前模型存在一些问题或者不再使用，如训练数据异常、数据量不够、不想再继续使用等原因，沟通达成一致拒绝。2、电话未接通且模型效果较差，会直接拒绝。如果需要申诉，请在百度智能云控制台内[提交工单](#)反馈

## 🔗 模型部署相关问题

### 🔗 平台的部署方式支不支持我的硬件？

部署类型	支持的硬件示例
通用ARM	绝大多数安卓、苹果手机；瑞芯微RK32、RK32、RK35系列、树莓派等开发板
英特尔神经计算棒	NCS 1代、NCS 2代
海思NNIE	Hi3559AV100/Hi3559CV100等
华为昇腾Atlas开发板	Atlas200计算盒、Atlas300 计算卡
比特大陆SE计算盒	Bitmain SE5
通用x86CPU	绝大多数英特尔和AMD CPU
通用x86CPU加速版	英特尔志强、酷睿、凌动系列CPU
高通骁龙	骁龙660以后芯片的手机
华为NPU	mate10，mate10pro，P20，mate20，荣耀v20等
华为达芬奇NPU	mate30，p40，nova6，荣耀v30等
英伟达GPU	消费级显卡GeForce系列、RTX系列、TITAN，专业显卡Quadro、Tesla系列
英伟达Jetson	TX2、Nano、Xavier、Xavier NX

## 快速链接

### 公有云部署

[公有云服务API调用常见错误码](#)

### 本地服务器部署

[Windows-GPU部署常见问题](#)

[Linux-C++部署常见问题](#)

[Linux-Python部署常见问题](#)

[Linux-Atlas部署常见问题](#)

### 通用小型设备部署

[Android部署常见问题](#)

[iOS部署常见问题](#)

[Windows-CPU部署常见问题](#)

[Linux-C++部署常见问题](#)

[Linux-Python部署常见问题](#)

### 软硬一体方案部署

[EdgeBoard\(FZ\)部署常见问题](#)

[EdgeBoard\(VMX\)部署常见问题](#)

[Jetson部署常见问题](#)

# 智能边缘控制台

## 智能边缘控制台-单节点版

### EasyEdge 智能边缘控制台-单节点版 IEC

EasyEdge Intelligent Edge Console（以下简称IEC）是EasyEdge推出的边缘设备管理的本地化方案。可以运行于多种架构、多系统、多类型的终端之上。通过IEC，用户可以方便地在本地进行

- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活，服务管理
- 接入本地和远程摄像头，网页中实时预览
- 自动监控和记录相关事件
- 硬件信息的可视化查看

支持的系统+CPU架构包括：

- Windows x86\_64 (Windows 7 ~ Windows 10，暂不支持Windows 11)

- Linux x86\_64 / arm32 / arm64

支持各类常见的AI加速芯片，包括：

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列
- 其他EasyDL/EasyEdge/BML支持的AI芯片
  - 完整列表可参考[这里](#)

## 🔗 Release Note

注意：2.0.0之后，默认以系统服务形式安装iec，无法兼容1.x版本的iec

版本号	发布时间	更新说明
2.2.2.1	2023-08-30	系统稳定性优化
2.2.2.0	2023-03-16	支持多媒体服务器硬件编解码
2.2.1	2022-11-15	修复告警规则删除问题
2.2.0	2022-10-27	新增onvif/gb28181支持；完善端云通信逻辑
2.0.0	2022-03-22	支持连接中心节点IECC；支持以系统服务安装
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	IEC 第一版！

## 🔗 快速开始

从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#)：intel、AMD的64位x86\_64 CPU
- [Linux amd64](#)：intel、AMD的64位x86\_64 CPU
- [Linux arm](#)：树莓派等32位的ARM CPU
- [Linux arm64](#)：RK3399、飞腾等64位的ARM CPU

或者从[纯离线服务管理页](#)可下载智能边缘控制台



您也可以通过先安装多节点版本IECC，通过中心节点来自动连接安装边缘节点。

## 🔗 Linux 安装

解压缩之后，目录结构如下

```
0 EasyEdge-IEC-v2.0.0-linux-amd64-standard > tree .
.
├── easyedge-iec
├── easyedge-iec-setup.sh
├── etc
│   └── easyedge-iec.yml
└── readme.txt
```

以系统服务形式安装 (推荐) 以root用户运行 `bash ./easyedge-iec-setup.sh install` 即可

```
[setup]: sudo could not be found
[setup]: Start to install IEC...
[setup]: + bash -c ". /easyedge-iec --com.role=edge --service=install"
[setup]: + bash -c "cp /code/EasyEdge-IEC-v2.2.2-linux-amd64-standard/easyedge-iec-setup.sh /usr/sbin/easyedge-iec-setup.sh"
[setup]: Install IEC success!
[setup]: + bash -c "/usr/sbin/easyedge-iec --com.role=edge --service=start"
[setup]: Start to check IEC status...
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: + bash -c "curl -s 127.0.0.1:8702 >/dev/null"
[setup]: + bash -c "/usr/sbin/easyedge-iec --com.role=edge --service=status | grep running > /dev/null 2>&1"
[setup]: IEC status: OK!
[easyedge-iec]: default configure file: /etc/easyedge-iec/easyedge-iec.yml
[easyedge-iec]: default log dir: /var/log/easyedge-iec/
[easyedge-iec]: service usage: service easyedge-iec { start | stop }
[setup]: Done!
```

- 日志 : `/var/log/easyedge-iec/easyedge-iec.log`
- 系统配置 : `/etc/easyedge-iec/easyedge-iec.yml`
- 服务启动/停止 : `service easyedge-iec { start | stop }` (不同操作系统内可能不同, 具体命令参考安装日志)

自定义安装 (不推荐) 自定义安装方法仅限于 安装脚本无法识别的情况。

- 拷贝 `./EasyEdge-IEC-v2.0.0-linux-amd64-standard/` 整个目录至自定义文件夹, 如 `/opt/EasyEdge-IEC`
- 进入到 `/opt/EasyEdge-IEC`
- 通过 `nohup` 等方法运行 `./easyedge-iec-linux-{您的系统架构} amd64: intel、AMD的64位x86_84 CPU arm : 树莓派等32位的ARM CPU * arm64 : RK3399、飞腾等64位的ARM CPU`
- 日志 : `./log/easyedge-iec.log`
- 系统配置 : `./easyedge-iec.yml`

## Windows 安装

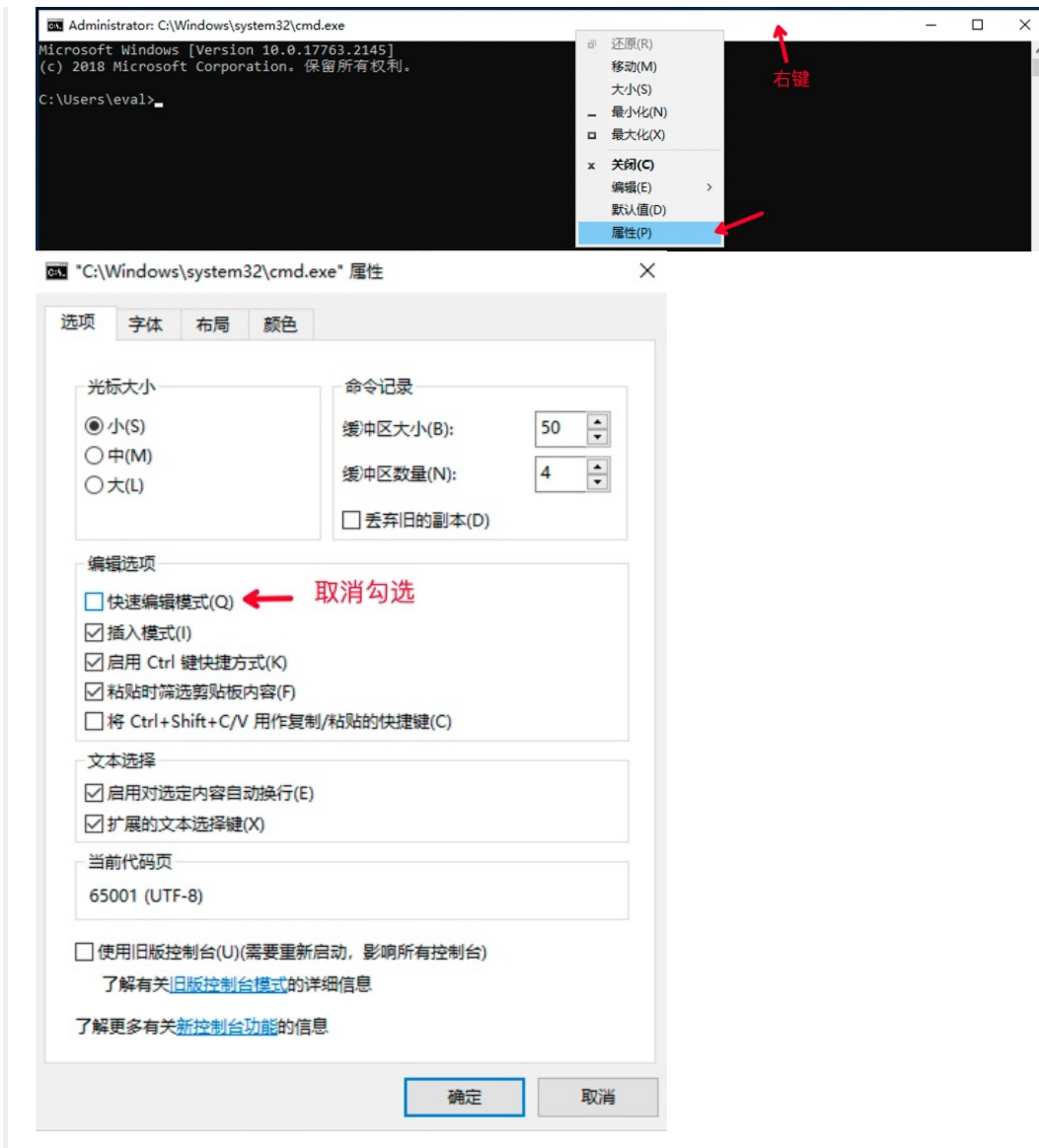
解压缩之后, 安装目录如下所示 :

```
0 tmp2 > tree EasyEdge-IEC-v2.2.2-windows-amd64-standard
EasyEdge-IEC-v2.2.2-windows-amd64-standard
├── easyedge-iec.exe
├── easyedge-iec-setup.bat
├── etc
│   └── easyedge-iec.yml
└── readme.txt

1 directory, 4 files
```

打开命令行 (非powershell) 运行 `easyedge-iec-setup.bat install`。

如果遇到hang住的情况, 可修改命令行配置 启动之后, 打开浏览器, 访问 `http://{设备ip}:8702/easyedge/iec` 即可 :

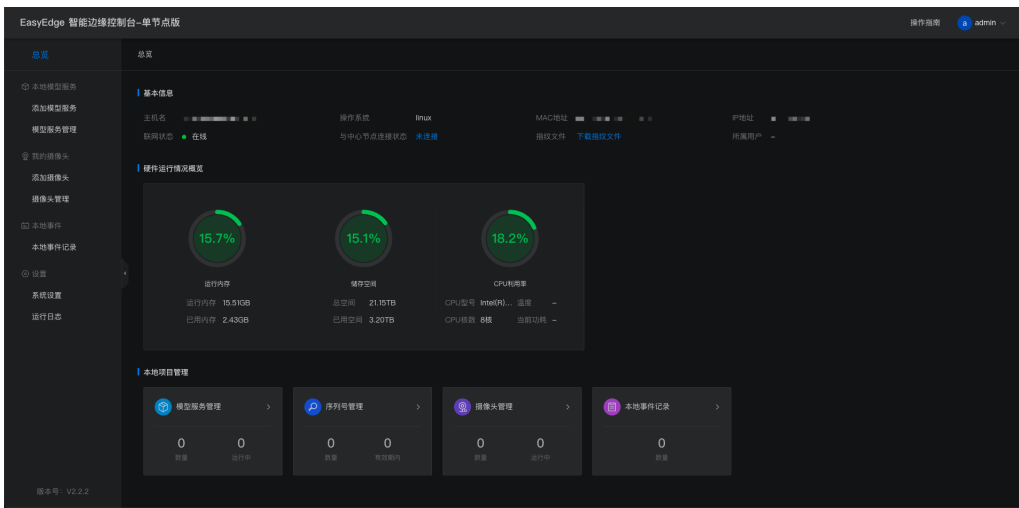


启动之后，打开浏览器，访问 [http://\(设备ip\):8702](http://(设备ip):8702) 即可：



默认用户名密码为 admin / easyedge

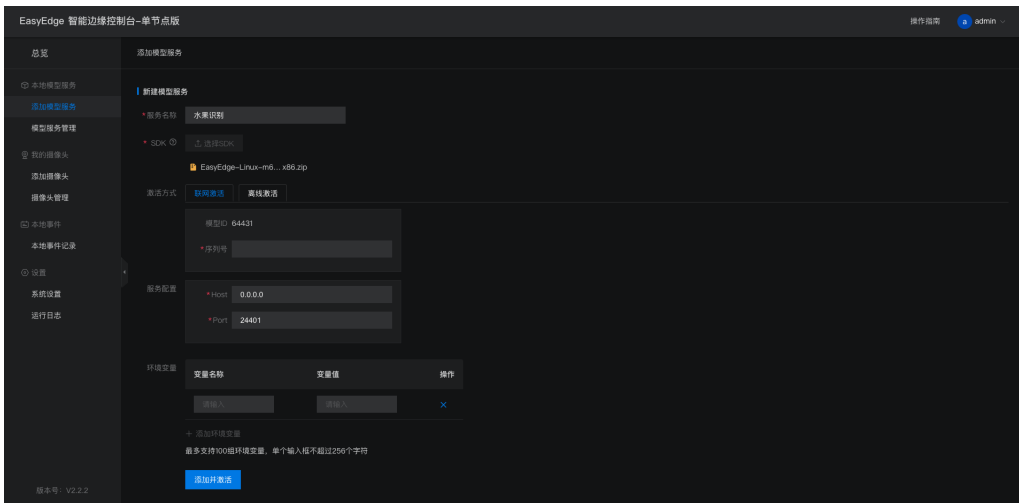




### 功能使用说明

#### ①添加模型服务

首先，点击导航栏的「本地模型服务」-「添加模型服务」。在页面中定义服务名称后，将已经下载好的Linux/Windows版本的SDK与IEC关联。关联完毕后可按两种激活方式，激活使用SDK。



部分SDK需要提前安装系统依赖，如TRT等，具体请参考EasyDL/BML/EasyEdge SDK使用文件中的环境依赖安装说明

### 联网激活

1. 在关联SDK完成后，需要在百度智能云控制台对应部署方式管理页中新增测试序列号或购买正式序列号。（图中以服务器版SDK为例）



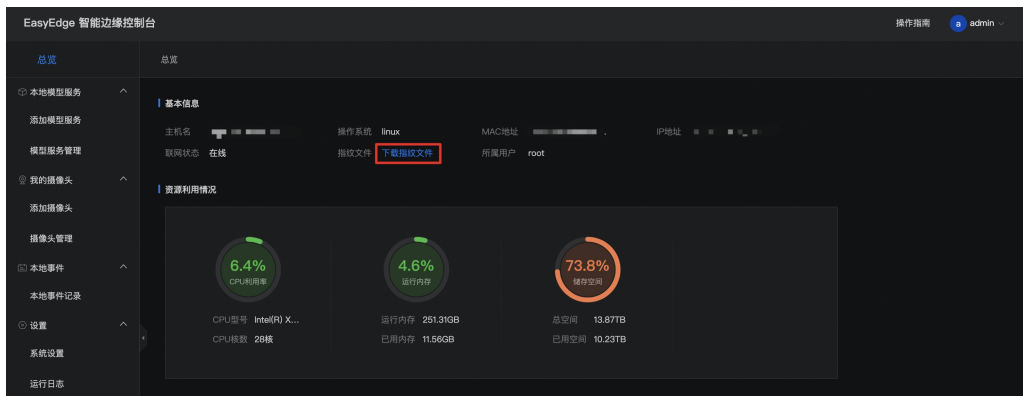
2. 再在IEC中填入所申请的序列号

3. 配置服务，在服务端口不冲突占用的情况下，使用默认即可

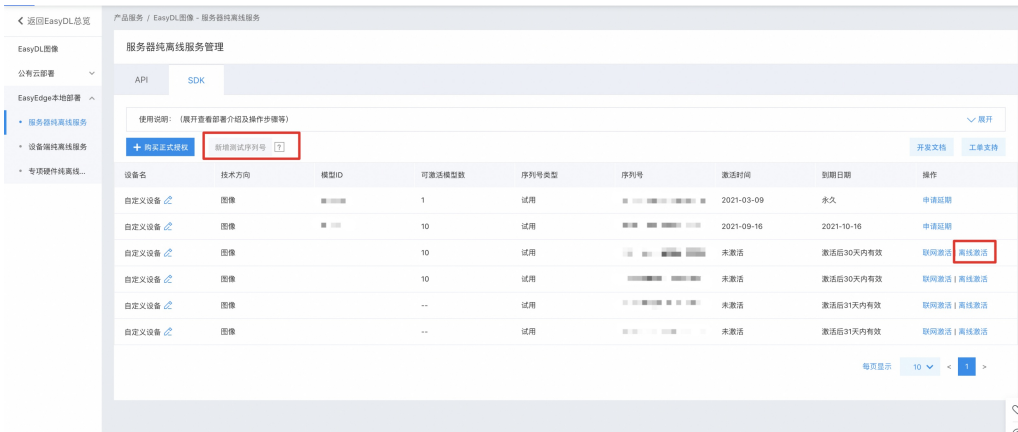
4. 添加并激活

### 离线激活

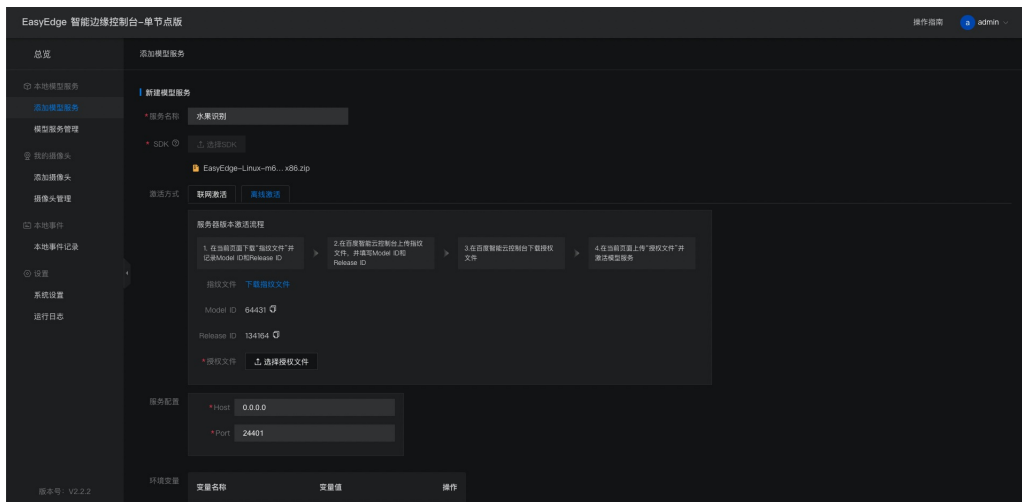
1. 在IEC总览页面下载「指纹文件」



2. 在百度智能云的**控制台**中找到SDK对应的管理列表，图中以服务器SDK为例。申请序列号后，点击对应序列号尾部的「离线激活」操作，按指引激活



3. 在IEC的添加模型服务页面，上传下载好的授权文件，完成激活



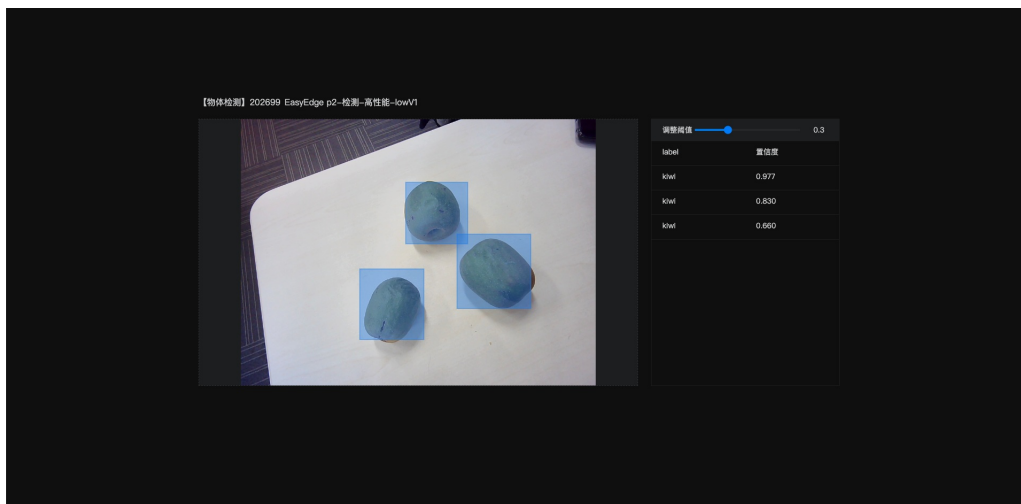
激活完成后即可在「模型服务管理」列表中启动服务，使用后续的操作栏功能。

序列号激活错误码

错误码	文案	描述
4001	parameters missing. 参数缺失	
4002	parameters invalid. 参数不合法	
4003	model invalid. 模型信息不合法	
4004	no more activation left. 该序列号和该设备的激活次数超上限	
4005	the serial key is out of date. 该序列号过期	
4006	the serial key has been activated. 该序列号已被其他设备激活	该序列号已被其他设备激活，不能重复激活。
4007	account invalid. 序列号不能用于其他账号的模型	序列号不能用于其他账号的模型，只能用于绑定账号的模型。
4008	serial key invalid. 序列号不合法	序列号不存在或找不到
4009	bundle id invalid. 包名不合法	
4010	product invalid. 产品不合法	如easydl的SDK使用BML的序列号来激活，会报该错误
4011	platform invalid. 平台不合法	
4012	activate too frequent. 激活太频繁	激活太频繁，请稍后再进行激活。
4013	device type and license type not match. 硬件类型和序列号类型不匹配	如使用加速版序列号激活基础版SDK会报该错误
4014	exceed max activate device num. 超过最大授权数量	
4015	technology invalid. 技术类型不合法	
4016	exceed max activate entity num. 超过最大模型数量	
4017	device invalid. 设备不合法	
4018	model invalid. 模型不合法	

### 体验本地demo

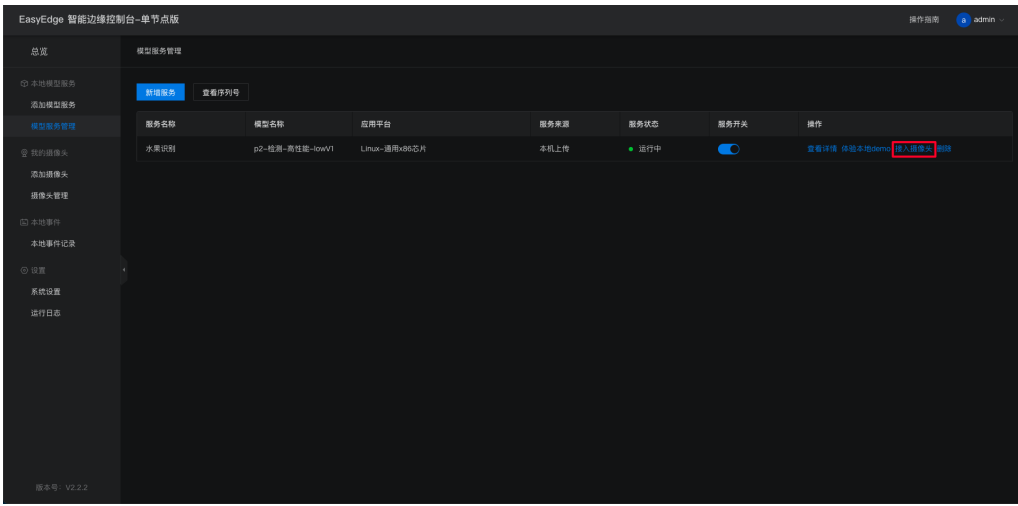
点击「本地demo体验」即可在立即上传图片进行预测



### 接入摄像头

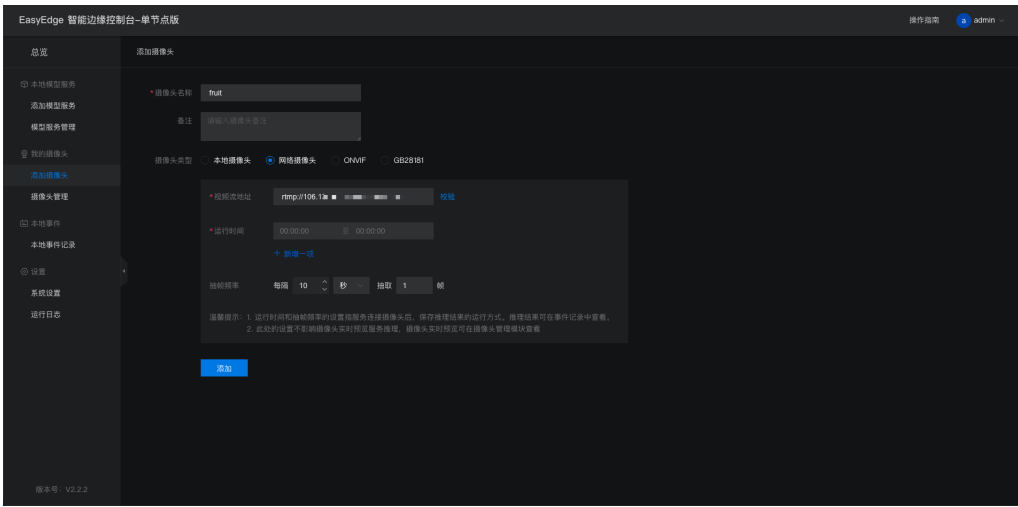
使用接入摄像头功能首先需要添加摄像头并创建告警规则，请参考第②步和第③步，完成后按照第④步操作

注：服务启动后也可参考「模型发布」模块的技术文档进行开发使用，本文档主要介绍IEC使用功能



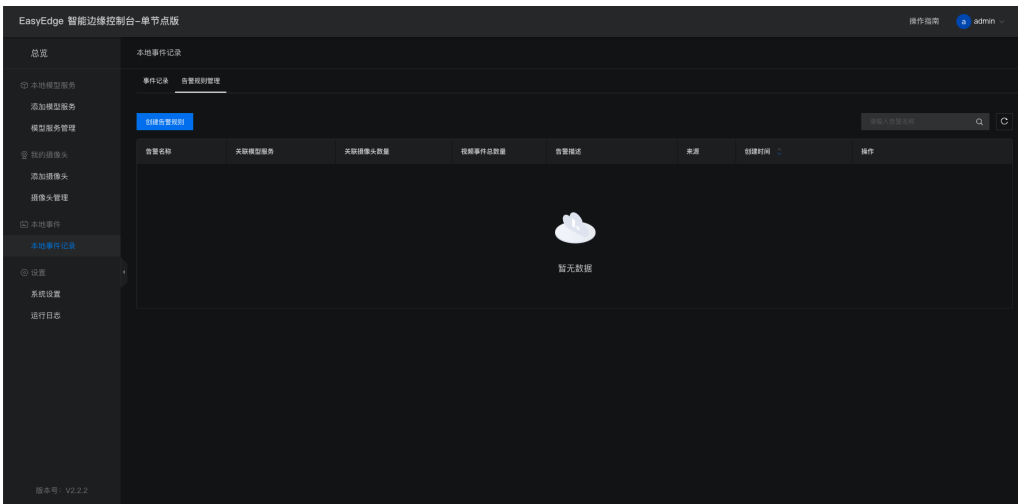
### ②添加摄像头

导航栏点击「我的摄像头」-「添加摄像头」，定义摄像头名称、备注后即可添加摄像头。支持本地摄像头、网络摄像头、ONVIF协议摄像头和GB28181协议摄像头。摄像头添加成功后即可设置摄像头的运行时间和频率

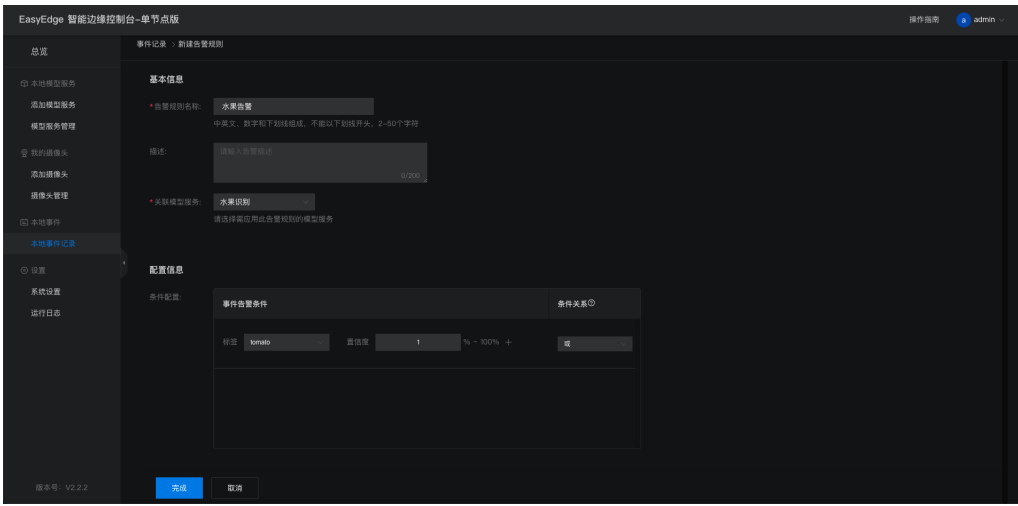


### ③创建告警规则

导航栏点击「本地事件」-「本地事件记录」，切换至「告警规则管理」tab。点进创建告警规则



选择要关联的模型服务，并配置产生告警不同标签需要满足的阈值条件

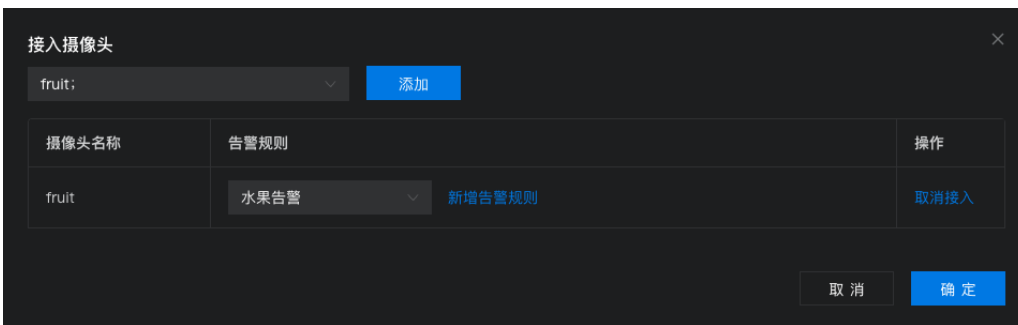


### ④ 摄像头接入模型服务预测

点击「本地模型服务」-「模型服务管理」中，所需接入预测的服务的「接入摄像头」

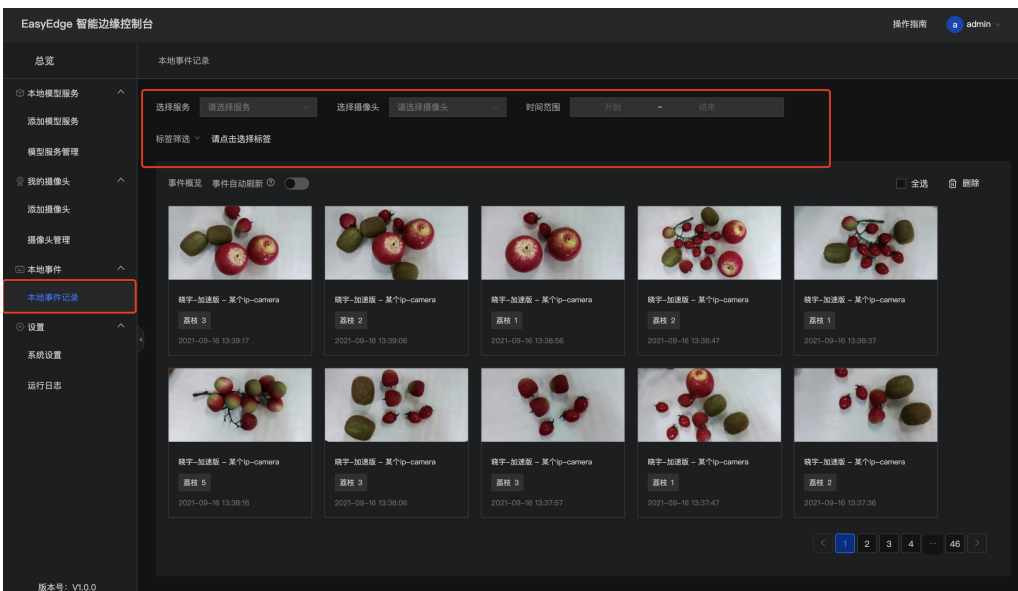


在弹出的弹窗中选择第②步中添加的摄像头，并选择第③步中创建的告警规则，此时点击确认即可在「摄像头管理」中的实时预览功能中查看摄像头预测结果



### ⑤ 本地事件

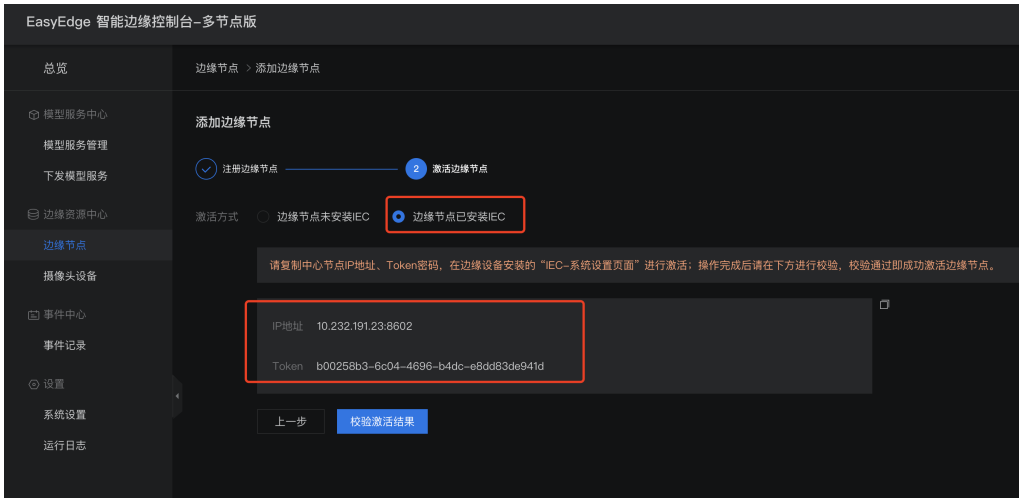
点击导航栏「本地事件记录」，可通过服务名称、摄像头名称、事件记录的时间、标签及置信度来筛选识别结果查看，多个标签及置信度同样也是“或”的逻辑记录。如有想要删除的事件数据可选择后删除，全选为本页全选。



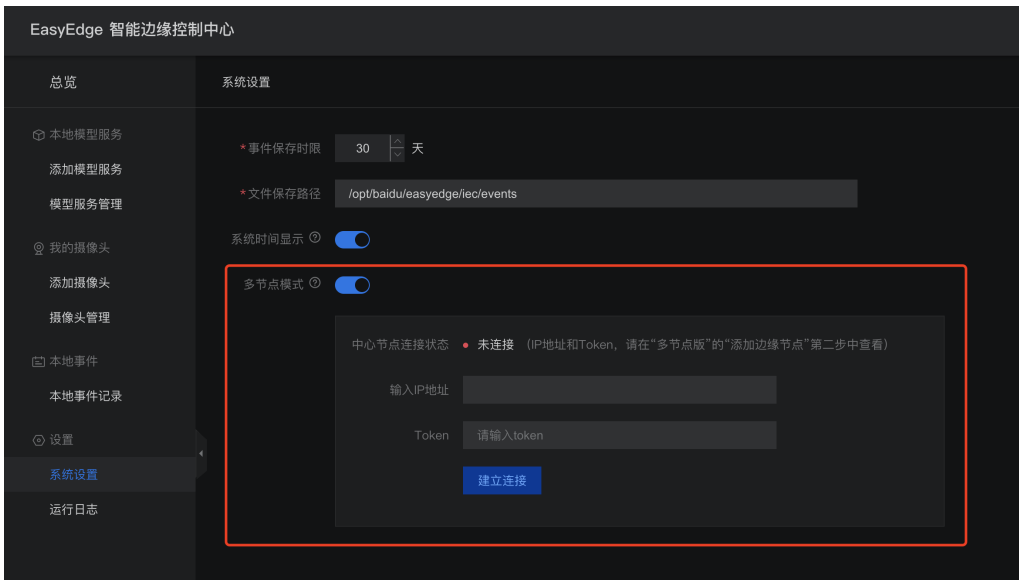
⑥ ⑥连接到智能边缘控制台-多节点版 (IECC)

与中心节点连接之后，边缘节点主程序版本会自动随控制中心版本升级。 (>2.0.0)

- Step 1 在IECC中添加边缘节点，选择「边缘节点已安装IEC」，并记录IP地址与Token



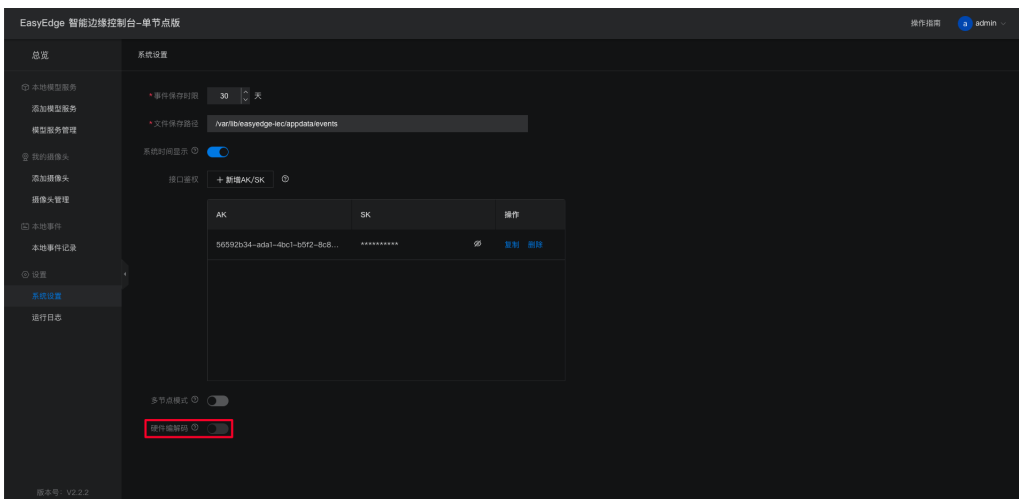
- Step 2 在IEC的系统设置中打开多节点模式，并填入刚才记录的IP地址与Token，点击建立连接



- 连接完成后即可在中心节点IECC去监控/管理/应用在边缘节点上的IEC

⑦ ⑦多媒体服务器使用硬件编解码

点击导航栏「设置」-「系统设置」，会自动检测当前硬件是否支持开启硬件编解码，如支持则可开启



## 配置项\*

配置文件 `easyedge-iec.yml` 中有关于IEC的各项配置说明，一般无需修改，请确保理解配置项含义之后，再做修改。

```
##### IEC系统配置
##### ----- 高级配置一般无需修改 -----
##### !!! 注意!!! 请确保理解配置项含义后再做修改
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
# role: edge
# 硬件利用率刷新时间间隔：过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# 事件监测触发扫描周期
eventTriggerIntervalSecond: 10
# IEC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: /var/lib/easyedge-iec/appdata
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge）
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
toFile: yes
# loggingFile: /var/log/easyedge-iec/easyedge-iec.log
loggingFolder: /var/log/easyedge-iec/
# 0:info; -1:debug; -2:verbose。设置为<-1时，SDK HTTP服务会输出DEBUG日志
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8702
listenHost: 0.0.0.0

sdk:
# GPU SDK所使用的cuda版本：auto(自动检测) / 9 / 10 / 10.2 / 11.0 / 11.1。如果auto检测不正确，可以强制指定版本号。更换版本只对新添加的模型有效
cudaVersion: auto
# AI服务启动时，额外配置的 LD_LIBRARY_PATH(linux) 或者 PATH(windows)
libPath: ./
# AI服务启动时，额外配置的其他环境变量。
ENVs:
  EDGE_CONTROLLER_KEY_LOG_BRAND: EasyEdge
##### EDGE_CONTROLLER_KEY_XXX: XXXX

commu:
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2

##### 数据库相关配置
db:
sqliteDbFile: /var/lib/easyedge-iec/easyedge-iec.db
hubDbFile: /var/lib/easyedge-iec/easyedge-iec.hub.db
eventDbFile: /var/lib/easyedge-iec/easyedge-event.db
fileServerDbFile: /var/lib/easyedge-iec/easyedge-fileserver.hub.db
nodeMonitorDbFile: /var/lib/easyedge-iec/easyedge-nodemonitor.hub.db

##### 推流相关配置
mediaserver:
flvPort: 8715
rtmpPort: 8716

##### 视频流相关配置
edgestream:
```

```

easyedge.yml:
# FATAL 1, ERROR 2, WARNING 3, INFO 4, DEBUG 5, VERBOSE_LEVEL1 6, VERBOSE_LEVEL2 7, VERBOSE_LEVEL3 8
logLevel: 5
listenHost: 0.0.0.0
listenPort: 8710
# 摄像头预览：识别结果绘制延迟消失
renderExtendFrames: 10
# 预测队列大小: 如果设置为60, 当摄像头fps=30时, 视频延迟约为2秒。 降低inferenceQueueSize可以降低预览延迟, 但是根据硬件的算力情况, 可能导致模型推理速度跟不上, 没有识别结果, 不建议设置太低
inferenceQueueSize: 60
videoEncodeBitRate: 400000
# 视频采样 & 视频实时预览分辨率设置
# 0: auto, 1: 1080p, 2: 720p, 3: 480p, 4: 360p, 5: 240p
resolution: 0
# 内置多媒体服务配置
# port设为0表示关闭
mediaServerFlvPort: 8713
mediaServerRtmpPort: 8714
mediaServerRtspPort: 0
mediaServerRtpPort: 8716

#### 信令服务器相关配置
sipserver:
listenHost: 0.0.0.0
listenPort: 8708
# 当前域
region: 3707000008
# 当前服务id
deviceId: 37070000082008000001
# 用户id固定头部
uid: 37070000081118
# 设备id固定头部
did: 37070000081318

loadbalance:
HTTPPort: 8780
TCPPortMin: 30000
TCPPortMax: 31000
UDPPortMin: 30000
UDPPortMax: 31000

```

## FAQ

启动服务后，进程中出现两个 `easyedge-iec` 进程 这是正常现象，IEC通过守护进程的方式来完成更新等操作。

启动服务时，显示端口被占用 `port already been used` 通过修改 `easyedge-iec.yml` 文件的配置后，再重新启动服务。

安装服务时，报错 `permission denied` 请以管理员身份运行安装程序。

中心节点重启后，边缘节点IEC一直离线 中心节点短时间的离线，边缘节点会自动重连。如果中心节点已经恢复在线，边缘节点长时间未自动连接上，可通过边缘节点 `iec` 的方法来重新连接（右上角 admin - 重启系统）

IEC 是否有Android / iOS 版本 我们将会在近期发布对Android操作系统的支持

添加SDK时，报错 `SDK not supported by this device` 一般是因为使用的SDK跟硬件不匹配，如 GPU的SDK，硬件没有GPU卡。对于Jetson，也可能是Jetpack版本不支持，可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表（cpp文件中的文件名来查看）来匹配。

## 智能边缘控制台-多节点版

### EasyEdge 智能边缘控制台——多节点版

#### 整体介绍

智能边缘控制台 - 多节点版（EasyEdge Intelligent EdgeConsole Center 以下简称IECC），是EasyEdge推出的边缘资源管理、服务应用与管理一站式本地化方案。

通过IECC，用户可以方便地在中心节点管理子节点：



- 边缘硬件资源的管理与监控
- EasyDL/BML/EasyEdge的SDK的 离线 / 在线激活, 服务管理
- 视频流解析, 接入本地和远程摄像头, 网页中实时预览
- 自动监控和记录相关视频流推理事件

#### 支持的系统+CPU架构包括：

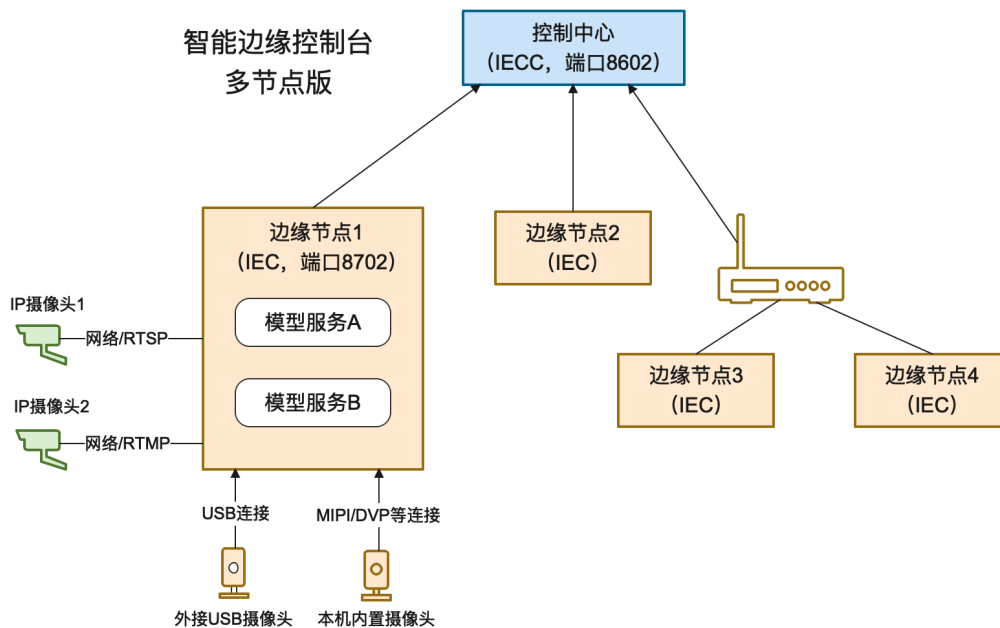
- Windows x86\_64 (Windows 7 ~ Windows 10, 暂不支持Windows 11)
- Linux x86\_64 / arm32 / arm64

#### 支持各类常见的AI加速芯片, 包括：

- NVIDIA GPU / Jetson 系列
- Baidu EdgeBoard FZ系列
- 比特大陆 Bitmain SC / SE 系列
- 华为 Atlas 系列
- 寒武纪 MLU 系列
- 其他EasyDL/EasyEdge/BML支持的AI芯片

#### 连接说明

以下为 中心节点 (控制中心), 边缘节点/子节点, 摄像头的连接示意：



其中：

- 控制中心需要有固定IP, 而边缘节点可以处于多级子网之下, 只需IEC能够主动访问到控制中心节点即可
- 模型服务均运行于各边缘节点之上
- 摄像头均与边缘节点相连

#### Release Note

版本号	发布时间	更新说明
2.2.2.1	2023-08-30	系统稳定性优化
2.2.2.0	2023-03-16	支持边缘节点多媒体服务器硬件编解码
2.2.1	2022-11-15	修复告警规则删除问题
2.2.0	2022-10-27	边缘节点新增Android支持；新增onvif/gb28181支持；优化端云通信通道安全
2.0.0	2022-03-25	多节点版上线！
1.0.2	2021-12-22	更新视频预览推流库；新增若干AI芯片支持；支持多种芯片温度、功耗展示；多项性能优化
1.0.0	2021-09-16	智能边缘控制台 - 单节点版 IEC 第一版！

## 安装

从这里选择您需要的操作系统和CPU架构下载：

- [Windows amd64](#)：intel、AMD的64位x86\_84 CPU
- [Linux amd64](#)：intel、AMD的64位x86\_84 CPU
- [Linux arm](#)：树莓派等32位的ARM CPU
- [Linux arm64](#)：RK3399、飞腾等64位的ARM CPU

或者从纯离线服务管理页可下载智能边缘控制台



以Linux为例，解压缩后目录结构如下所示：

```

./EasyEdge-IECC-v{版本号}-{平台}-{架构}/
├── easyedge-iecc
├── easyedge-iecc-setup.sh
├── etc
├── easyedge-iec.yml
└── readme.txt

```

## Linux 系统

### 通过系统服务形式安装（推荐）

以管理员运行 `bash easyedge-iecc-setup.sh install` 即可。

```
[setup]: sudo could not be found
[setup]: Start to install IECC...
[setup]: + bash -c "./easyedge-iecc --com.role=hub --service=install"
[setup]: Install IECC success!
[setup]: + bash -c "/usr/sbin/easyedge-iecc --com.role=hub --service=start"
[setup]: Start to check IECC status...
[setup]: + bash -c "curl -s 127.0.0.1:8602 >/dev/null"
[setup]: + bash -c "curl -s 127.0.0.1:8602 >/dev/null"
[setup]: + bash -c "curl -s 127.0.0.1:8602 >/dev/null"
[setup]: + bash -c "/usr/sbin/easyedge-iecc --com.role=hub --service=status | grep running > /dev/null 2>&1"
[setup]: IECC status: OK!
[easyedge-iecc]: default configure file: /etc/easyedge-iecc/easyedge-iecc.yml
[easyedge-iecc]: default log dir: /var/log/easyedge-iecc/
[easyedge-iecc]: service usage: service easyedge-iecc { start | stop }
[setup]: Done!
```

出现IECC status: OK!字样，表示安装成功。

- 日志：/var/log/easyedge-iecc/easyedge-iecc.log
- 系统配置：/etc/easyedge-iecc/easyedge-iecc.yml
- 服务启动/停止：service easyedge-iecc { start | stop } (不同操作系统内可能不同，具体命令参考安装日志)
- 配置服务自启动：可根据不同操作系统参考[这里](#)进行对应配置

可通过 `bash easyedge-iecc-setup.sh uninstall` 来卸载，以及`bash easyedge-iecc-setup.sh upgrade`来升级为当前安装包的版本

### 自定义安装（不推荐）

自定义安装仅限于 安装脚本无法识别您的操作系统的情况。

- 拷贝 ./EasyEdge-IECC-v{版本号}-{平台}-{架构}/ 整个目录至自定义文件夹，如/opt/EasyEdge-IECC
- 进入到 /opt/EasyEdge-IECC
- 通过 nohup 等方法运行 ./easyedge-iecc --com.role=hub amd64: intel、AMD的64位x86\_64 CPU arm：树莓派等32位的ARM CPU \* arm64：RK3399、飞腾等64位的ARM CPU
- 日志：./log/easyedge-iecc.log
- 系统配置：./etc/easyedge-iec.yml

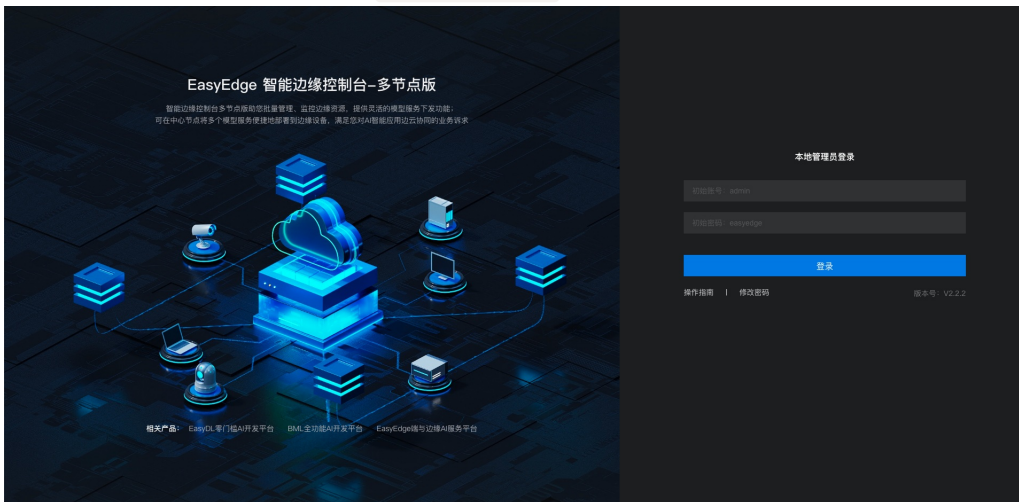
Windows 系统 打开命令行（非powershell）运行 easyedge-iecc-setup.bat install。

注：如果遇到hang住的情况，可修改命令行配置





验证安装：启动之后，打开浏览器，访问 <http://{设备ip}:8602> 即可：



**更新服务：** 停止服务，下载最新的安装包，替换二进制可执行文件，启动服务。

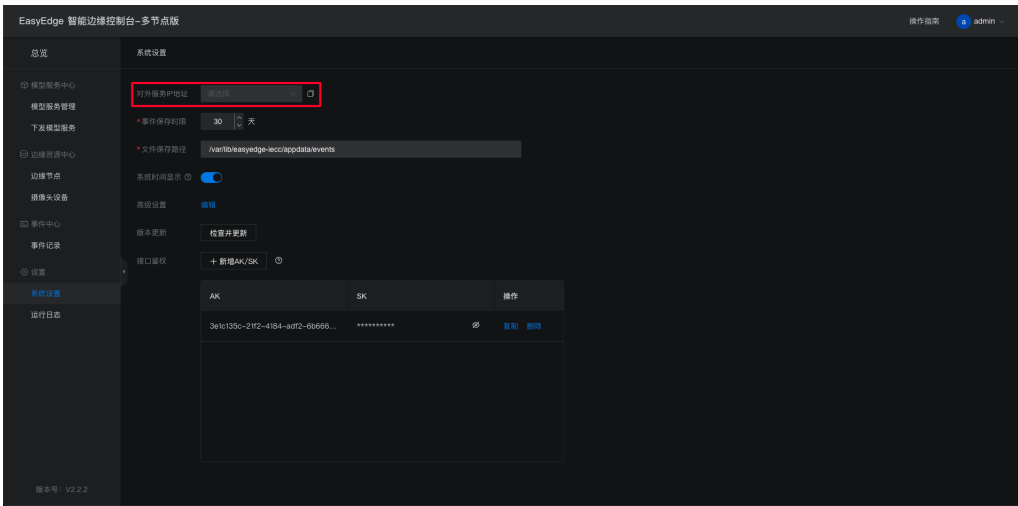
注：

1. 中心节点更新到新版之后，已连接的边缘节点会自动跟随中心节点，自我升级到同样的版本。
2. 报错: Text file busy. 一般是因为服务没有停止。

## 🔗 使用流程

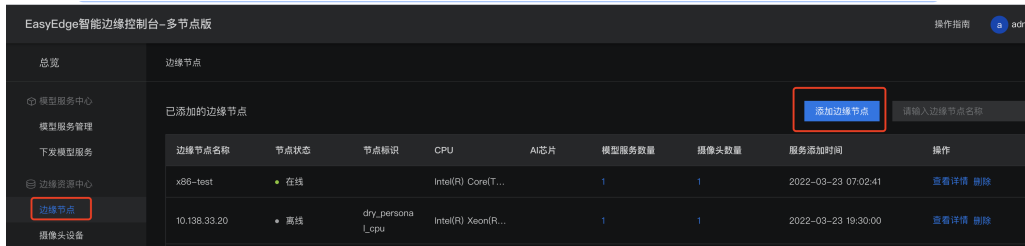
### 🔗 ①配置对外服务IP地址

- 中心节点所在机器可能存在多网卡，由于边缘节点需要连接中心节点，因此需要配置对外服务IP地址，确定唯一的对外IP
- 导航栏点击「设置」-「系统设置」，设置对外服务IP地址，候选列表为扫描到的IP地址

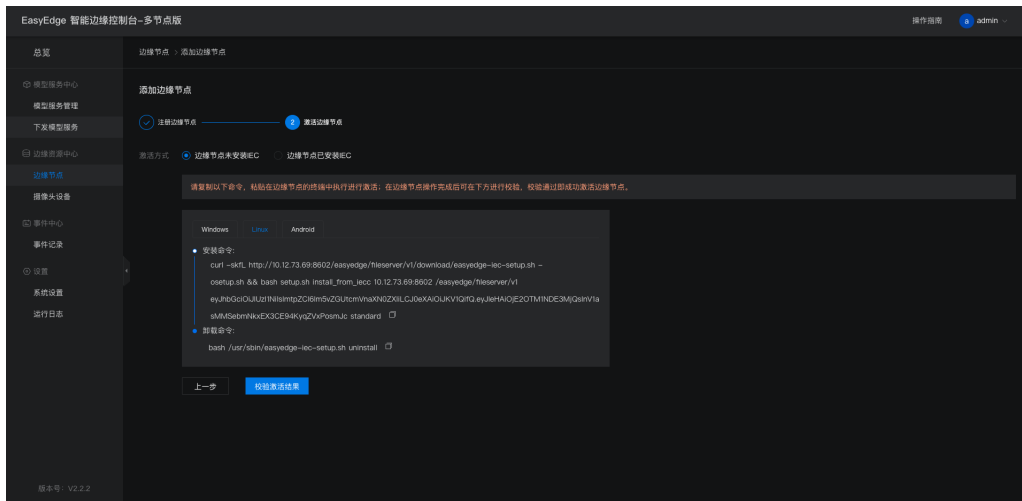


②注册并激活边缘节点

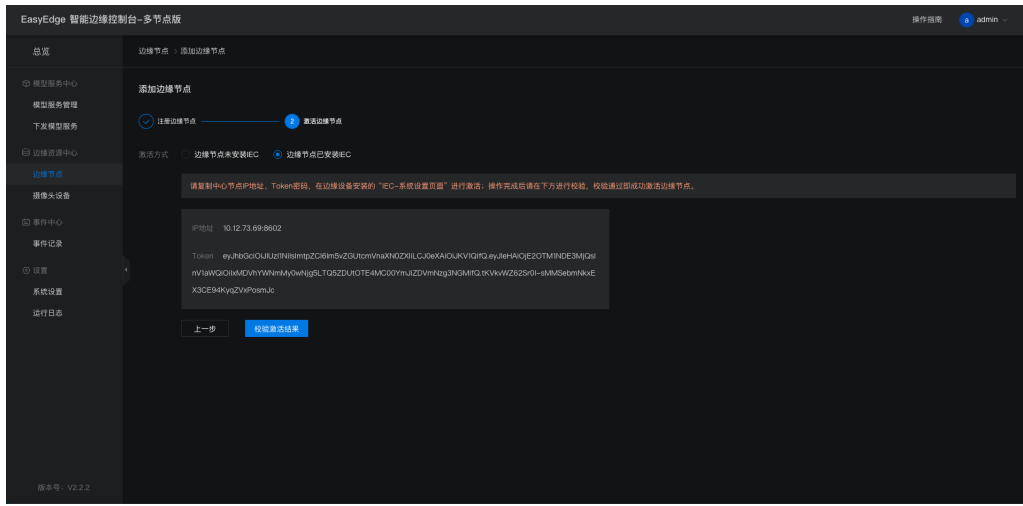
- 导航栏点击「边缘资源中心」-「边缘节点」，点击页面中的「添加边缘节点」按钮



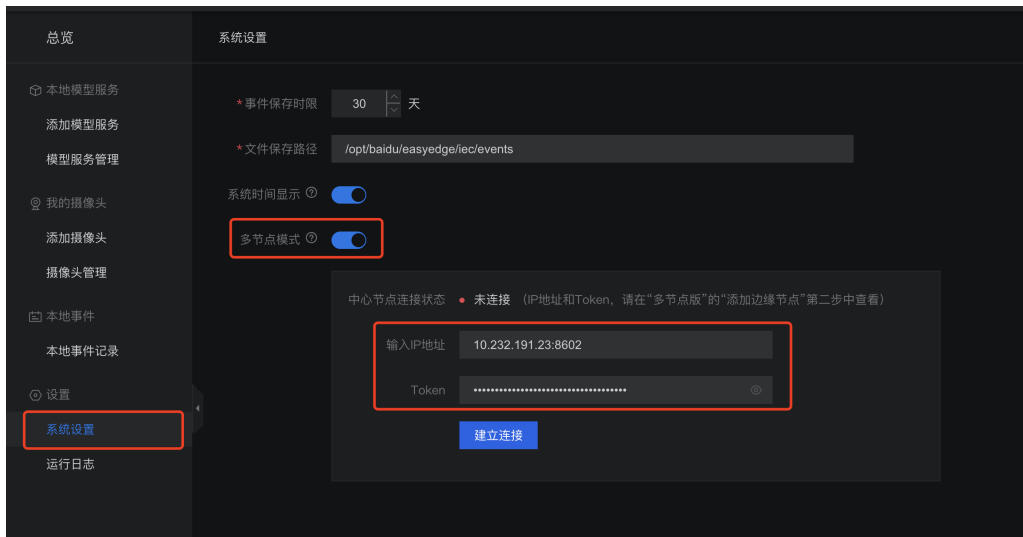
- 注册边缘节点，填写基本信息
- 激活边缘节点，根据边缘节点上是否安装智能边缘控制台-单节点版（IEC）分两种激活方式
  - 边缘节点未安装IEC：复制提供的命令，在边缘节点的终端中输入执行（命令会自动在当前目录，下载单节点版IEC并注册到控制中心）。终端命令执行完成后，在下方校验激活结果，如结果通过即可完成边缘节点的激活



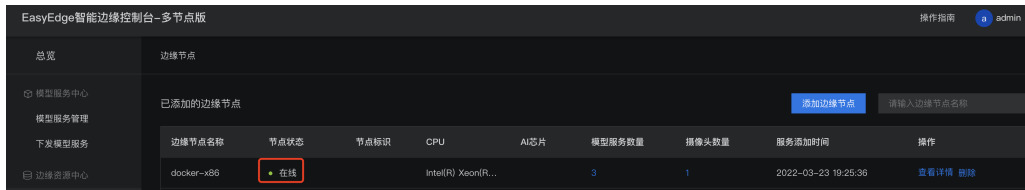
- 边缘节点已安装IEC：记录页面中提供的IP地址和Token



- 在边缘节点的IEC-系统设置中，打开多节点模式开关，将刚才记录的IP地址和Token填入其中，建立连接

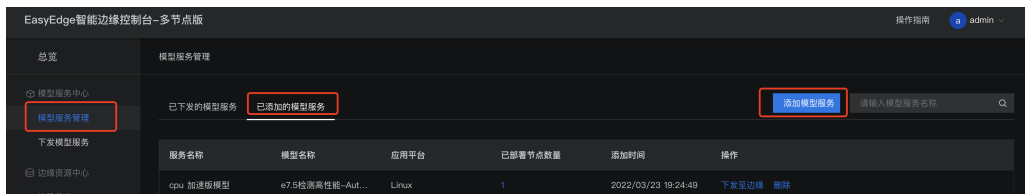


- 成功激活后可在边缘节点页面中看到一行状态为在线的记录

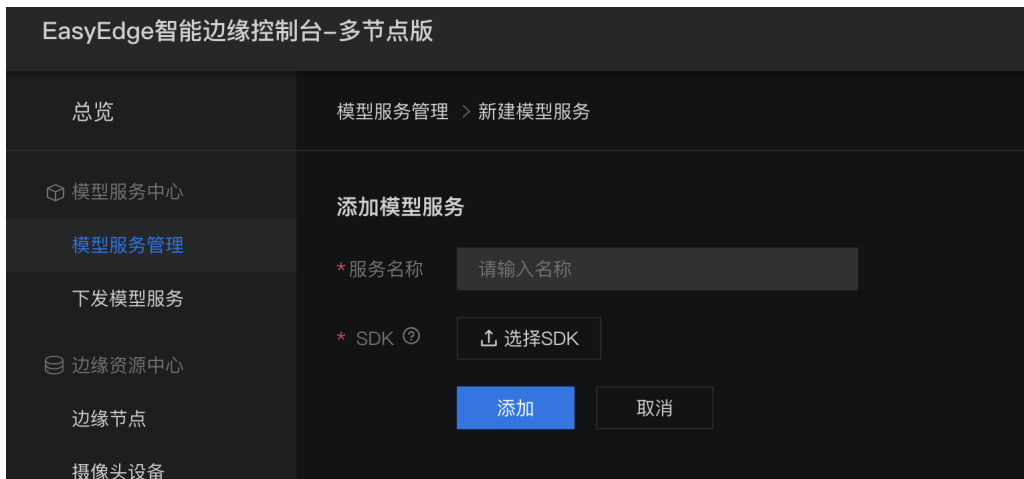


③上传并下发模型服务

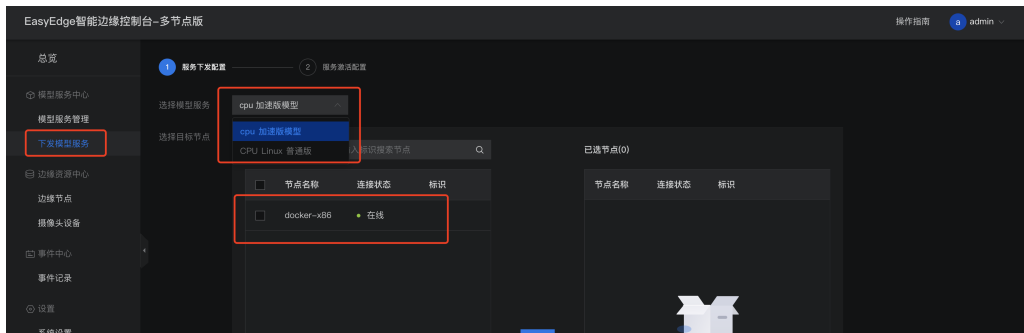
- 导航栏点击「模型服务中心」-「模型服务管理」,已添加的模型服务页面中点击「添加模型服务」



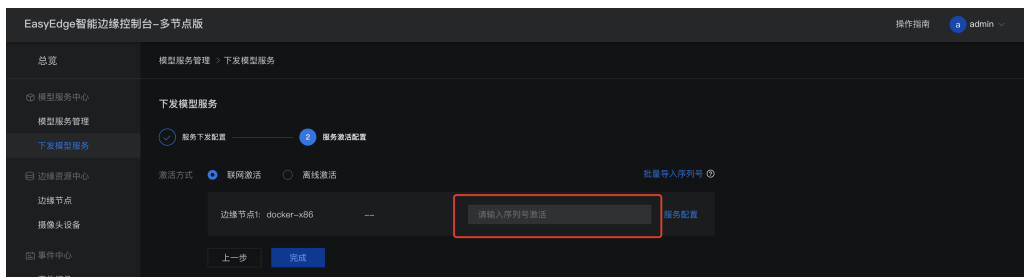
- 上传来自于EasyDL/BML的SDK，目前仅支持Windows/Linux的SDK



- 添加成功后可在已添加的模型服务页面查看添加的模型服务SDK
- 在模型服务SDK上传成功以及边缘节点也添加激活过后，即可将模型服务下发至边缘。点击导航栏-下发模型服务，选择已添加的模型服务，选择下发的目标节点（支持多节点批量下发）进行模型服务下发



- 确定下发配置后，填入模型服务在边缘节点联网激活运行的序列号（支持批量导入）即可完成模型服务下发，序列号可在[智能云控制台](#)获取。离线激活的过程可参考IECC中的具体指引



- 完成上述流程后即可在模型服务管理-已下发的模型服务列表中查看记录，并进行下一步应用功能体验

注：完成此步骤后即可在边缘节点进行二次集成已下发的模型服务，具体的集成方式可在文档-某图像任务类型-模型发布中查找对应的SDK开发文档进行集成开发

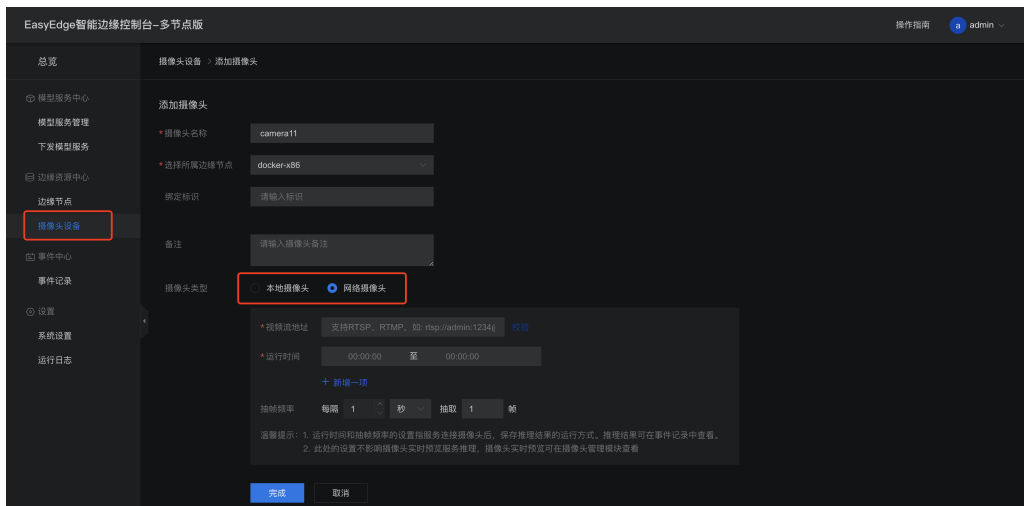


下发时可以通过高级配置设置服务运行的host和port。若不设置，默认host为0.0.0.0，port为系统随机分配的可用端口

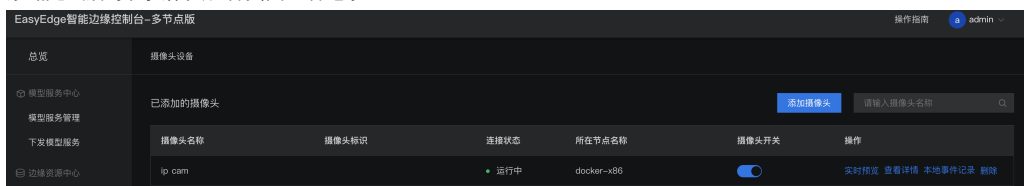
#### ④配置摄像头

Step ④ - ⑦ 描述的是如何使用IECC可视化进行视频流式推理与应用，对此有需求的用户建议详细查看后续步骤内容。如仅需对下发的模型服务进行二次集成的用户无需进行后续操作，参考SDK对应的开发文档进行集成即可

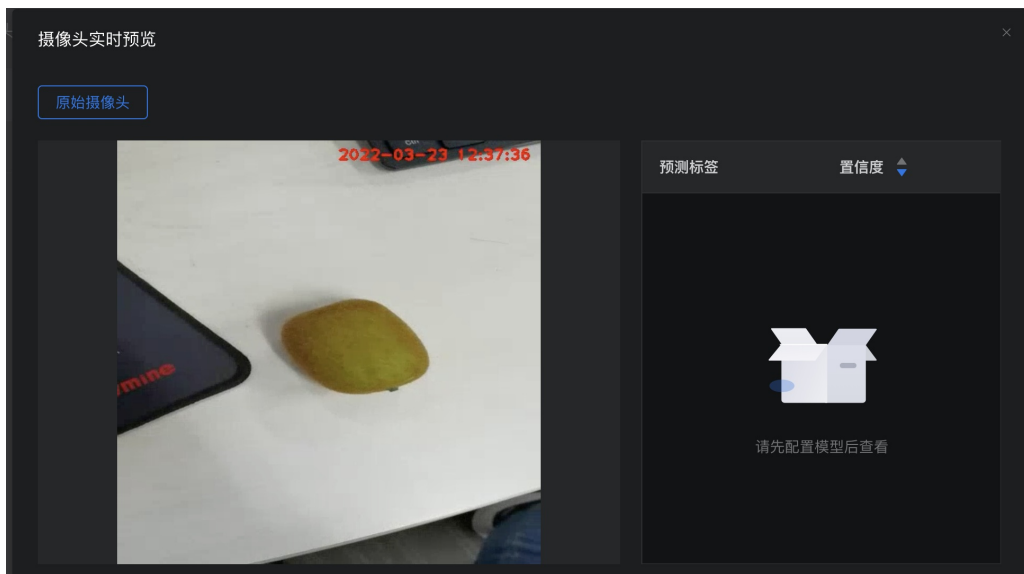
- 首先需要确定边缘节点已经接入物理摄像头，可通过USB接口接入，也可通过RTSP/RTMP流式协议接入。在摄像头设备页面点击添加摄像头按钮，填写对应的信息添加摄像头。支持设置摄像头的运行时间以及摄像头的抽帧频率



- 添加完成后可在摄像头设备页面查看记录

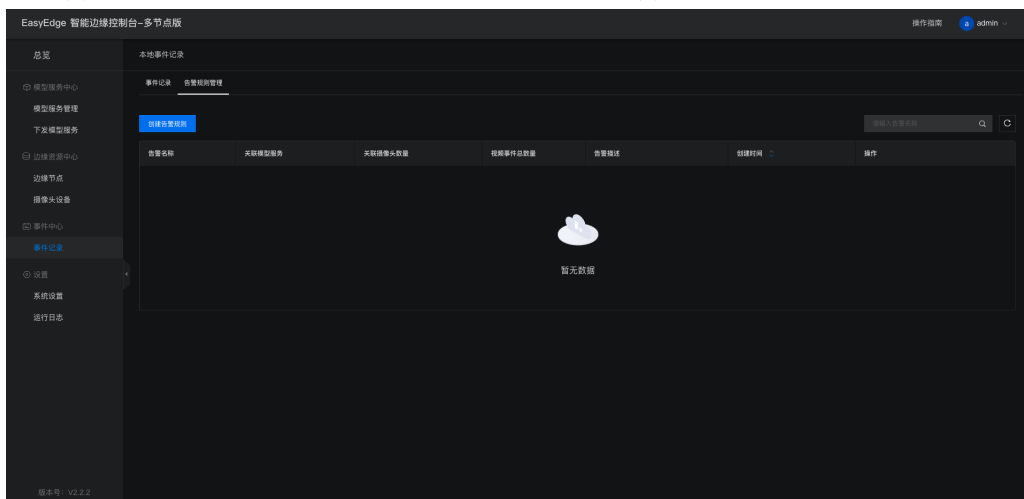


- 点击预览可查看摄像头预览画面



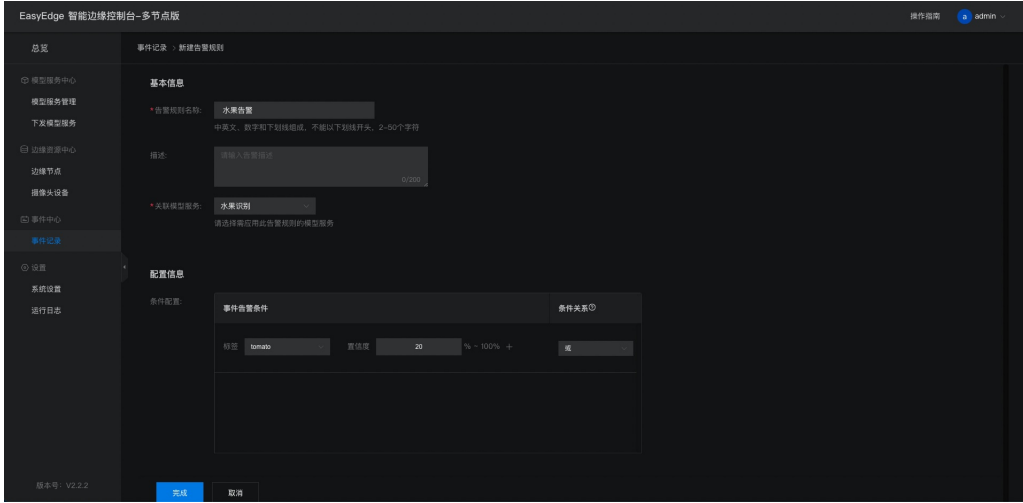
## ⑤配置告警规则

- 导航栏点击「事件中心」-「事件记录」，切换至「告警规则管理」tab。点进创建告警规则



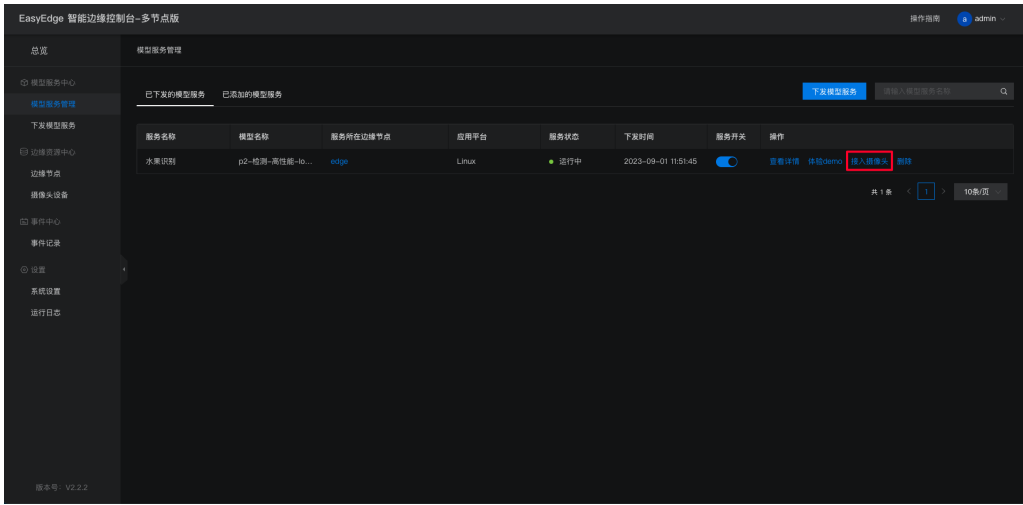


- 选择要关联的模型服务，并配置产生告警不同标签需要满足的阈值条件

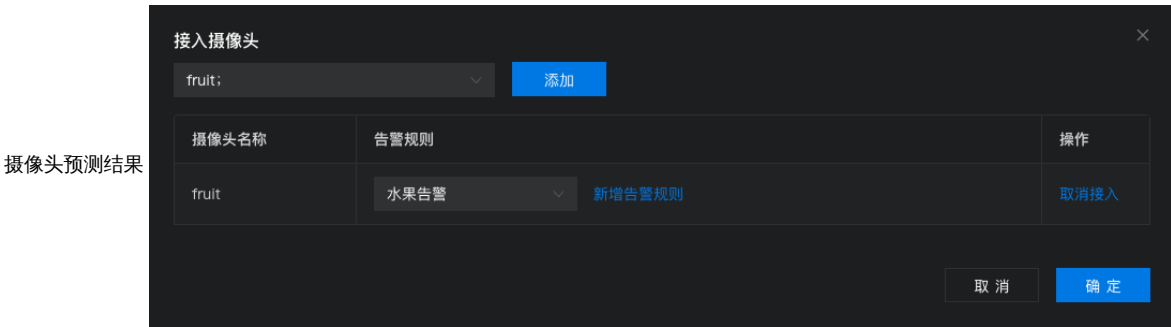


⑥模型服务接入视频流预测

- 点击「本地模型服务」-「模型服务管理」中，所需接入预测的服务的「接入摄像头」

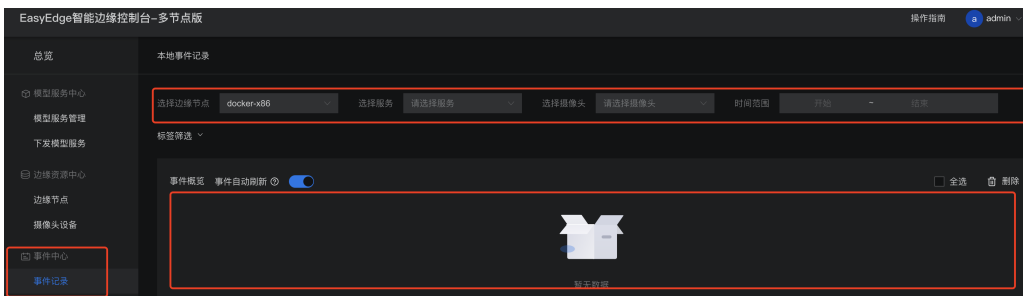


- 在弹出的弹窗中选择第④步中添加的摄像头，并选择第⑤步中创建的告警规则，此时点击确认即可在「摄像头管理」中的实时预览功能中查看



⑦视频事件告警

- 可在事件中心-事件记录中查看满足时间告警条件的图片记录



⑧高级配置说明

在系统设置 - 高级，可以修改控制中心的高级系统配置

```
#### IECC 控制中心系统配置
#### ----- 高级配置一般无需修改 -----
#### !!! 注意!!! 请确保理解配置项含义后再做修改
version: 3

com:
# hub: 作为中心节点模式启动。 edge: 作为子节点启动
role: hub
# 硬件利用率刷新时间间隔： 过低的刷新间隔可能会消耗CPU资源。
hardwareUsageRefreshSecond: 60
# IECC保存SDK等资源的路径：可填写 "default", 也可以直接填写绝对路径
appDataFolder: /var/lib/easyedge-iecc/appdata
# 是否开启DEBUG模式：开启之后，将会打印大量日志，便于追踪系统问题
debug: no
# 是否需要用户名/密码登陆，强烈建议打开！（默认用户名/密码为 admin/easyedge）
loginEnabled: yes
# 登录有效时间，单位秒
sessionMaxAge: 3600

logging:
# 是否把日志打印到控制台
toStd: no
# 是否把日志输出到文件。关闭后，将无法在页面中正确显示系统日志
ToFile: yes
# loggingFile: /var/log/easyedge-iecc/easyedge-iecc.log
loggingFolder: /var/log/easyedge-iecc/
# 0:info; -1:debug; -2:verbose
level: -1

webservice:
# WEB服务的监听端口
listenPort: 8602
listenHost: 0.0.0.0

commu:
mqServer:
  host: 0.0.0.0
  port: 8632
  HTTPPort: 8620
  maxPayload: 8388608
  pingIntervalSecond: 30
# 普通消息等待respond的超时时间
respondWaitTimeoutSecond: 2
nodeRefreshIntervalSecond: 30

#### 数据库相关配置
db:
sqliteDbFile: /var/lib/easyedge-iecc/easyedge-iecc.db
hubDbFile: /var/lib/easyedge-iecc/easyedge-iecc.hub.db
eventDbFile: /var/lib/easyedge-iecc/easyedge-event.db
hubEventDbFile: /var/lib/easyedge-iecc/easyedge-event.hub.db
fileServerDbFile: /var/lib/easyedge-iecc/easyedge-fileserver.hub.db
nodeMonitorDbFile: /var/lib/easyedge-iecc/easyedge-nodemonitor.hub.db

#### 推流相关配置
mediaserver:
flvPort: 8613
rtmpPort: 8614

#### 文件服务器相关配置
fileserver:
root: /var/lib/easyedge-iecc/fs
```

## FAQ

启动服务后，进程中出现两个 `easyedge-iecc` 进程 这是正常现象，IECC通过守护进程的方式来完成更新等操作。

启动服务时，显示端口被占用 `port already been used` 通过修改 `easyedge-iecc.yml` 文件的配置后，再重新启动服务。

安装服务时，报错permission denied 请以管理员身份运行安装程序。

添加SDK时，报错 SDK不支持该硬件。 SDK not supported by this device 一般是因为使用的SDK跟硬件不匹配，如 GPU的SDK，硬件没有GPU卡。对于Jetson，也可能是Jetpack版本不支持，可以通过查看 本机Jetpack版本和SDK支持的Jetpack版本列表（cpp文件中的文件名来查看）来匹配。

## EasyEdge 智能边缘控制台-单节点版 IEC API

### 概述

欢迎使用EasyEdge 智能边缘控制台-单节点版 IEC。

您可以使用本文档介绍的API对单节点版 IEC服务进行操作。

### 接口概览

单节点版 IEC API 提供下列接口类型：

接口类型	描述
AI服务接口	接口包括查询、启动、停止等
摄像头设备接口	接口包括创建、查询、更新、删除、启动、停止等
物联网设备接口	接口包括创建、查询、更新、删除等
其他接口	包括生成鉴权用的Access Token等

### 通用说明

API调用遵循HTTP协议。数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

### API认证机制

所有API的安全认证一律采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串。对于每个HTTP请求，需先调用生成Access Token的接口生成认证字符串。提交认证字符串放在query里。服务端根据生成算法验证认证字符串的正确性。

当服务端接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

### 通信协议

支持HTTP调用方式。

### 请求结构说明

数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

请求参数包括如下4种：

参数类型	说明
URI	通常用于指明操作类型，如:POST /iec/iapi/v{version}/{type}/{op}
Query参数	URL中携带的请求参数，通常用来传入认证字符串
HEADER	通过HTTP头域传入
RequestBody	通过JSON格式组织的请求数据体

### 响应结构说明

响应值分为两种形式：

响应内容	说明
HTTP STATUS CODE	如200,400,403,404等
ResponseBody	JSON格式组织的响应数据体

### API版本号

参数	类型	参数位置	描述	是否必须
version	String	URI参数	API版本号, 当前值为1	必须

## ☞ 日期与时间规范

日期与时间的表示有多种方式。为统一起见, 除非是约定俗成或者有相应规范的, 凡需要日期时间表示的地方一律采用UTC时间, 遵循ISO 8601, 并做以下约束:

- 表示日期一律采用YYYY-MM-DD方式, 例如2014-06-01表示2014年6月1日。
- 表示时间一律采用hh:mm:ss方式, 并在最后加一个大写字母Z表示UTC时间。例如23:00:10Z表示UTC时间23点0分10秒。
- 凡涉及日期和时间合并表示时, 在两者中间加大写字母T, 例如2014-06-01T23:00:10Z表示UTC时间2014年6月1日23点0分10秒。

## ☞ 规范化字符串

通常一个字符串中可以包含任何Unicode字符。在编程中这种灵活性会带来不少困扰。因此引入“规范字符串”的概念。一个规范字符串只包含百分号编码字符以及URI (Uniform Resource Identifier) 非保留字符 (Unreserved Characters)。RFC 3986规定URI非保留字符包括以下字符: 字母 (A-Z, a-z)、数字 (0-9)、连字号 (-)、点号 (.)、下划线 (\_)、波浪线 (~)。

将任意一个字符串转换为规范字符串的方式是:

- 将字符串转换成UTF-8编码的字节流。
- 保留所有URI非保留字符原样不变。
- 对其余字节做一次RFC 3986中规定的百分号编码 (Percent-Encoding), 即一个%后面跟着两个表示该字节值的十六进制字母。字母一律采用大写形式。

示例: 原字符串: this is an example for 测试, 对应的规范字符串: this%20is%20an%20example%20for%20%E6%B5%8B%E8%AF%95。

## ☞ 服务域名

服务端点Endpoint	协议
{单节点版 IEC所在机器IP}:{单节点版 IEC监听端口, 默认8702}	HTTP

## ☞ 错误码

### ☞ 错误码格式

当用户访问API出现错误时, 会返回给用户相应的错误码和错误信息, 便于定位问题, 并做出适当的处理。请求发生错误时通过Response Body返回详细错误信息, 遵循如下格式:

参数名	类型	说明
status	int	表示具体错误类型。
msg	String	有关该错误的详细说明。

例如:

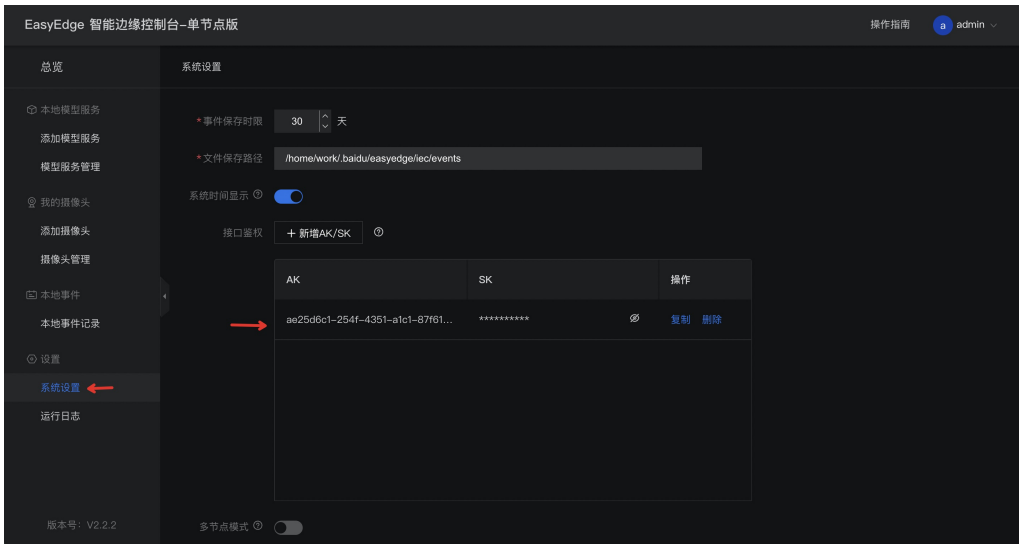
```
{
  "status": 170000,
  "message": "发生错误. Error"
}
```

## ☞ 公共请求参数

### ☞ 公共请求Query参数

当用户访问API时, 需要通过query参数传入access\_token, 如 [http://127.0.0.1:8702/iec/iapi/v1/camera/new?access\\_token={access\\_token}](http://127.0.0.1:8702/iec/iapi/v1/camera/new?access_token={access_token})

### ☞ 获取生成Access Token用的AK、SK



获取Access Token

基本信息

Path : /auth/v1/token

Method : GET

接口描述 :

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

名称	类型	是否必须	默认值	备注	其他信息
grantType	string	必须		固定传入client_credentials	
clientId	string	必须		设置-系统设置-AK	
clientSecret	string	必须		设置-系统设置-SK	

返回数据

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
status	number	必须			
data	object	非必须			
├ expiresIn	number	必须		过期时间戳, 秒级	
├ accessToken	string	必须			

AI服务相关接口

本地服务列表

基本信息

Path : /iec/iapi/v1/aiservice/list

Method : GET

接口描述 :

请求参数

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object []	非必须			item 类型: object
├ id	number	必须			
├ uuid	string	必须			
├ name	string	必须			
├ source	string	必须			
├ activationType	string	必须		ONLINE 在线激活 / OFFLINE 纯离线激活	
├ activationSerialNum	string	必须			
├ host	string	必须			
├ port	number	必须			
├ env	object	必须			
├ LD_LIBRARY_PATH	string	非必须			
├ AAA1	string	非必须			
├ sssss	string	非必须			
├ isServiceOn	boolean	必须			
├ serviceType	string	必须		process 进程 / container 容器	
├ serviceStatus	string	必须			
├ serviceTip	string	必须			
├ createAt	string	必须			
msg	string	非必须			
status	number	非必须			

## 启动服务

## 基本信息

**Path** : /iec/iapi/v1/aiservice/start

**Method** : POST

**接口描述** :

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
serviceId	number	必须		服务id可以从“服务列表”接口获取	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
status	number	非必须			

## 停止服务

### 基本信息

**Path** : /iec/iapi/v1/aiservice/stop

**Method** : POST

**接口描述** :

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

#### Body

名称	类型	是否必须	默认值	备注	其他信息
serviceld	number	必须		服务id可以从“服务列表”接口获取	

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
status	number	非必须			

### 更新服务

#### 基本信息

**Path** : /iec/iapi/v1/aiservice/update

**Method** : POST

**接口描述** : 更新服务后服务会自动重启

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

#### Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	非必须			
name	string	非必须			
host	string	非必须			
port	number	非必须			
env	array []	非必须		环境变量，格式为 [ ["key", "value"] ]	item 类型: array

### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
status	number	非必须			

### 摄像头设备相关接口

#### 校验摄像头

#### 基本信息

**Path** : /iec/iapi/v1/camera/check-validity

**Method** : POST

**接口描述** :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
cameraAddr	string	必须		摄像头地址	
cameraType	string	必须		摄像头类型, IP表示网络摄像头, LOCAL表示本地摄像头	枚举: IP,LOCAL

🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	boolean	必须		摄像头是否合法	

🔗 新增摄像头

🔗 基本信息

**Path** : /iec/iapi/v1/camera/new

**Method** : POST

**接口描述** :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**



名称	类型	是否必须	默认值	备注	其他信息
name	string	必须		摄像头名称	
camAddr	string	非必须		摄像头地址，IP或LOCAL时必须	
camType	string	必须		摄像头类型，IP表示网络摄像头，LOCAL表示本地摄像头，ONVIF表示ONVIF摄像头，GB28181表示国标摄像头	枚举： LOCAL,IP,ONVIF,GB28181
onvifConfig	object	非必须		ONVIF时必须	
├ ip	string	必须		IP	
├ port	number	必须		端口	
├ username	string	必须		用户名	
├ password	string	必须		密码	
gb28181Config	object	非必须		GB28181时必须	
├ sipName	string	必须		信令服务器用户名	
├ sipDeviceId	string	必须		信令服务器设备ID	
├ sipPassword	string	必须		信令服务器密码	
remark	string	必须		摄像头备注	
timeRange	array []	必须		摄像头运行区间	item 类型: array
├		非必须			
├		非必须		秒	
frameExtract	object	必须		抽帧配置	
├ everySecond	number	必须		每多少秒	
├ frames	number	必须		抽多少帧	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	object	必须			
├ id	number	必须		当前节点唯一ID	
├ uuid	string	必须		全局唯一ID	

## 摄像头列表

## 基本信息

**Path** : /iec/iapi/v1/camera/list

**Method** : GET

**接口描述** :

## 请求参数

**Query**

参数名称	是否必须	示例	备注
pageNo	否		页数
pageSize	否		每页数量

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	object []	必须			item 类型: object
├ id	number	必须		摄像头ID	
├ uuid	string	必须		摄像头全局唯一ID	
├ name	string	必须		摄像头名称	
├ remark	string	必须		摄像头备注	
├ source	string	必须		摄像头来源	枚举: SELF,IECC
├ cameraType	string	必须		摄像头类型	枚举: LCOAL,IP,ONVIF,GB28181
├ cameraStatus	string	必须		摄像头状态	枚举: RUNNING,ERROR,STOPPED

## 摄像头详情

## 基本信息

**Path** : /iec/iapi/v1/camera/get

**Method** : GET

**接口描述** :

## 请求参数

**Query**

参数名称	是否必须	示例	备注
id	是		摄像头ID

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object	必须			
├ id	number	必须		摄像头ID	
├ uuid	string	必须		摄像头全局唯一ID	
├ name	string	必须		摄像头名称	
├ remark	string	必须		摄像头备注	
├ source	string	必须		摄像头来源	枚举: SELF,IECC
├ cameraType	string	必须		摄像头类型	枚举: LCOAL,IP,ONVIF,GB28181
├ cameraAddr	string	必须		摄像头地址, IP或LOCAL时有意义	
├ onvifConfig	object	非必须			
├ ip	string	必须		IP	
├ port	number	必须		端口	
├ username	string	必须		用户名	
├ password	string	必须		密码	
├ gb28181Config	object	非必须			
├ sipName	string	必须		信令服务器用户名	
├ sipDeviceId	string	必须		信令服务器设备ID	
├ sipPassword	string	必须		信令服务器密码	
├ frameExtractInterval	number	必须		每多少秒	
├ frameExtractNum	number	必须		抽多少帧	
├ onlineTime	array []	必须		摄像头运行区间	item 类型: array
├		非必须			
├		非必须		秒	
├ cameraStatus	string	必须		摄像头状态	枚举: RUNNING,ERROR,STOPPED

#### 更新摄像头

#### 基本信息

**Path** : /iec/iapi/v1/camera/update

**Method** : POST

接口描述 :

#### 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	
name	string	必须		摄像头名称	
camAddr	string	非必须		摄像头地址，IP或LOCAL时必填	
camType	string	必须		摄像头类型，IP表示网络摄像头，LOCAL表示本地摄像头，ONVIF表示ONVIF摄像头，GB28181表示国标摄像头	枚举: LOCAL,IP,ONVIF,GB28181
onvifConfig	object	非必须		ONVIF时必填	
├─ ip	string	必须		IP	
├─ port	string	必须		端口	
├─ username	string	必须		用户名	
├─ password	string	必须		密码	
gb28181Config	object	非必须		GB28181时必填	
├─ sipName	string	必须		信令服务器用户名	
├─ sipDeviceId	string	必须		信令服务器设备ID	
├─ sipPassword	string	必须		信令服务器密码	
remark	string	必须		摄像头备注	
timeRange	array []	必须		摄像头运行区间	item 类型: array
├─		非必须			
├─		非必须		秒	
frameExtract	object	必须		抽帧配置	
├─ everySecond	number	必须		每多少秒	
├─ frames	number	必须		抽多少帧	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		修改是否成功	

## 启动摄像头

## 基本信息

**Path** : /iec/iapi/v1/camera/start

**Method** : POST

**接口描述** :

## 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		开启摄像头是否成功	

## 停止摄像头

## 基本信息

**Path** : /iec/iapi/v1/camera/stop

**Method** : POST

**接口描述** :

## 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		关闭摄像头是否成功	

## 删除摄像头

## 基本信息

**Path** : /iec/iapi/v1/camera/delete

**Method** : POST

**接口描述** :

## 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		删除摄像头是否成功	

## 物联网设备相关接口

## 新增设备

## 基本信息

**Path** : /iec/iapi/v1/plc/new

**Method** : POST

**接口描述** :

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
name	string	必须		设备名称	
deviceType	string	必须		设备类型	枚举: modbus,opcua
modbus	object	非必须		modbus时必须填	
├─ protocol	string	必须		协议类型	枚举: tcp,rtu
├─ rtu	object	非必须		rtu时必须填	
├─ port	string	必须		端口	
├─ baudrate	number	必须		波特率	
├─ databit	number	必须		数据位	枚举: 5,6,7,8
├─ stopbit	number	必须		停止位	枚举: 1,2
├─ parity	string	必须		校验位	枚举: N,E,O
├─ tcp	object	非必须		tcp时必须填	
├─ address	string	必须		连接地址	
├─ port	number	必须		端口	
├─ slaved	number	必须		从站号	
├─ interval	number	必须		采样间隔	

opcua	object	非必须		opcua时必须填	
└─ endpoint	string	必须		通道地址	
└─ securityPolicy	string	必须		安全策略	枚举: None,Basic256Sha256,Aes128Sha256RsaOaep,Aes256Sha256RsaPss
└─ securityMode	string	必须		安全模式	枚举: None,Sign,SignAndEncrypt
└─ certificate	string	非必须		数字证书	
└─ privateKey	string	非必须		密钥证书	
└─ username	string	非必须		用户名	
└─ password	string	非必须		密码	
└─ timeout	number	必须		连接超时时间	
└─ interval	number	必须		采样间隔	
remark	string	必须		设备备注	
attributes	object []	必须		属性	item 类型: object
└─ id	string	必须		标识符	
└─ name	string	必须		属性名	
└─ type	string	必须		类型	枚举: bool,int16,int32,int64,float32,float64,string
└─ defaultValue	string	非必须		默认值	
└─ unit	string	必须		单位	
└─ required	boolean	必须		是否必填	
properties	object []	必须		测点	item 类型: object
└─ id	string	必须		标识符	
└─ name	string	必须		测点名	
└─ type	string	必须		类型	枚举: bool,int16,int32,int64,float32,float64,string
└─ mode	string	必须		读写类型	枚举: ro,rw
└─ unit	string	必须		单位	
└─ modbus	object	非必须		modbus时必须填	
└─ func	number	必须		寄存器类型 : 1.线圈寄存器 2.离散输入寄存器 3.保持寄存器 4.输入寄存器	枚举: 1,2,3,4
└─ address	string	必须		寄存器地址	
└─ quantity	number	必须		寄存器数量	
└─ opcua	object	非必须		opcua时必须填	
└─ nodeid	string	必须		节点ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object	必须			
├ id	number	必须		当前节点唯一ID	
├ uuid	string	必须		全局唯一ID	

## 设备列表

## 基本信息

**Path** : /iec/iapi/v1/plc/list

**Method** : GET

**接口描述** :

## 请求参数

## Query

参数名称	是否必须	示例	备注
pageNo	否		页数
pageSize	否		每页数量

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object []	必须			item 类型: object
├ id	number	必须		设备ID	
├ uuid	string	必须		设备全局唯一ID	
├ name	string	必须		设备名称	
├ remark	string	必须		设备备注	
├ deviceType	string	必须		设备类型	
├ source	string	必须		设备来源	枚举: SELF,IECC
├ status	string	必须		设备状态	枚举: RUNNING,ERROR,STOPPED

## 设备详情

## 基本信息

**Path** : /iec/iapi/v1/plc/get

**Method** : GET

**接口描述** :

## 请求参数

## Query

参数名称	是否必须	示例	备注
id	是		设备ID



返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object	必须			
├ id	string	必须		设备ID	
├ uuid	string	必须		设备全局唯一ID	
├ name	string	必须		设备名称	
├ deviceType	string	必须		设备类型	枚举: modbus,opcua
├ modbus	object	非必须		modbus时必填	
├─ protocol	string	必须		协议类型	枚举: tcp,rtu
├─ rtu	object	非必须		rtu时必填	
├─ port	string	必须		端口	
├─ baudrate	number	必须		波特率	
├─ databit	number	必须		数据位	枚举: 5,6,7,8
├─ stopbit	number	必须		停止位	枚举: 1,2
├─ parity	string	必须		校验位	枚举: N,E,O
├─ tcp	object	非必须		tcp时必填	
├─ address	string	必须		连接地址	
├─ port	number	必须		端口	
├─ slaved	number	必须		从站号	
├─ interval	number	必须		采样间隔	
├─ opcua	object	非必须		opcua时必填	
├─ endpoint	string	必须		通道地址	
├─ securityPolicy	string	必须		安全策略	
├─ securityMode	string	必须		安全模式	
├─ certificate	string	非必须		数字证书	
├─ privateKey	string	非必须		密钥证书	
├─ username	string	非必须		用户名	
├─ password	string	非必须		密码	
├─ timeout	number	必须		连接超时时间	
├─ interval	number	必须		采样间隔	
├─ remark	string	必须		设备备注	
├─ source	string	必须		设备来源	枚举: SELF,IECC
├─ attributes	object []	必须		属性	item 类型: object
├─ id	string	必须		标识符	

name	string	必须	属性名	
type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,string
defaultValue	string	非必须	默认值	
unit	string	必须	单位	
required	boolean	必须	是否必填	
properties	object []	必须	测点	item 类型: object
id	string	必须	标识符	
name	string	必须	测点名	
type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,string
mode	string	必须	读写类型	枚举: ro,rw
unit	string	必须	单位	
modbus	object	非必须	modbus时必填	
func	number	必须	寄存器类型: 1.线圈寄存器 2.离散输入寄存器 3.保持寄存器 4.输入寄存器	枚举: 1,2,3,4
address	string	必须	寄存器地址	
quantity	number	必须	寄存器数量	
opcua	object	非必须	opcua时必填	
nodeid	string	必须	节点ID	
status	string	必须	设备状态	枚举: RUNNING,ERROR,STOPPED
url	string	非必须	测点数据获取websocket连接地址	
realData	object	非必须	当前测点数据	

更新设备

基本信息

Path : /iec/iapi/v1/plc/update

Method : POST

接口描述 :

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		设备ID	
name	string	必须		设备名称	

deviceType	string	必须		设备类型	枚举: modbus,opcua
modbus	object	非必须		modbus时必须填	
├─ protocol	string	必须		协议类型	枚举: tcp,rtu
├─ rtu	object	非必须		rtu时必须填	
├─ port	string	必须		端口	
├─ baudrate	number	必须		波特率	
├─ databit	number	必须		数据位	枚举: 5,6,7,8
├─ stopbit	number	必须		停止位	枚举: 1,2
├─ parity	string	必须		校验位	枚举: N,E,O
├─ tcp	object	非必须		tcp时必须填	
├─ address	string	必须		连接地址	
├─ port	number	必须		端口	
├─ slaveld	number	必须		从站号	
├─ interval	number	必须		采样间隔	
opcua	object	非必须		opcua时必须填	
├─ endpoint	string	必须		通道地址	
├─ securityPolicy	string	必须		安全策略	枚举: None,Basic256Sha256,Aes128Sha256RsaOaep,Aes256Sha256RsaPss
├─ securityMode	string	必须		安全模式	枚举: None,Sign,SignAndEncrypt
├─ certificate	string	非必须		数字证书	
├─ privateKey	string	非必须		密钥证书	
├─ username	string	非必须		用户名	
├─ password	string	非必须		密码	
├─ timeout	number	必须		连接超时时间	
├─ interval	number	必须		采样间隔	
remark	string	必须		设备备注	
attributes	object []	必须		属性	item 类型: object
├─ id	string	必须		标识符	
├─ name	string	必须		属性名	
├─ type	string	必须		类型	枚举: bool,int16,int32,int64,float32,float64,string
├─					

defaultValue	string	非必须		默认值	
unit	string	必须		单位	
required	boolean	必须		是否必填	
properties	object	必须		测点	item 类型: object
id	string	必须		标识符	
name	string	必须		测点名	
type	string	必须		类型	枚举: bool,int16,int32,int64,float32,float64,string
mode	string	必须		读写类型	枚举: ro,rw
unit	string	必须		单位	
modbus	object	非必须		modbus时必填	
func	number	必须		寄存器类型: 1.线圈寄存器 2.离散输入寄存器 3.保持寄存器 4.输入寄存器	枚举: 1,2,3,4
address	string	必须		寄存器地址	
quantity	number	必须		寄存器数量	
opcua	object	非必须		opcua时必填	
nodeid	string	必须		节点ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	boolean	必须		修改是否成功	

## 删除设备

## 基本信息

Path: /iec/iapi/v1/plc/delete

Method: POST

接口描述:

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		设备ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		删除设备是否成功	

#### 写测点

#### 基本信息

**Path** : /iec/iapi/v1/plc/property/write

**Method** : POST

**接口描述** :

#### 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	
propertyId	string	必须		测点标识符	
propertyValue	any	必须		测点值	

#### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		修改测点值是否成功	

#### 功能更新记录

时间	版本	说明
2023-05-25	1.0.0	第一版

## EasyEdge 智能边缘控制台-多节点版 IECC API

#### 概述

欢迎使用EasyEdge 智能边缘控制台-多节点版 IEC。您可以使用本文档介绍的API对多节点版 IEC服务进行操作。

#### 接口概览

多节点版 IEC API 提供下列接口类型：

接口类型	描述
AI服务接口	接口包括查询、启动、停止等
摄像头设备接口	接口包括创建、查询、更新、删除、启动、停止等
物联网设备接口	接口包括创建、查询、更新、删除等
其他接口	包括生成鉴权用的Access Token等

#### 通用说明

API调用遵循HTTP协议。数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。

## API认证机制

所有API的安全认证一律采用Access Key与请求签名机制。Access Key由Access Key ID和Secret Access Key组成，均为字符串。对于每个HTTP请求，需先调用生成Access Token的接口生成认证字符串。提交认证字符串放在query里。服务端根据生成算法验证认证字符串的正确性。当服务端接收到用户的请求后，系统将使用相同的SK和同样的认证机制生成认证字符串，并与用户请求中包含的认证字符串进行比对。如果认证字符串相同，系统认为用户拥有指定的操作权限，并执行相关操作；如果认证字符串不同，系统将忽略该操作并返回错误码。

## 通信协议

支持HTTP调用方式。

## 请求结构说明

数据交换格式为JSON，所有request/response body内容均采用UTF-8编码。请求参数包括如下4种：

参数类型	说明
URI	通常用于指明操作类型，如:POST /iec/iapi/v{version}/{type}/{op}
Query参数	URL中携带的请求参数，通常用来传入认证字符串
HEADER	通过HTTP头域传入
RequestBody	通过JSON格式组织的请求数据体

## 响应结构说明

响应值分为两种形式：| 响应内容 | 说明 | |--|--| HTTP STATUS CODE | 如200,400,403,404等 || ResponseBody | JSON格式组织的响应数据体 |

## API版本号

参数	类型	参数位置	描述	是否必须
version	String	URI参数	API版本号，当前值为1	必须

## 日期与时间规范

日期与时间的表示有多种方式。为统一起见，除非是约定俗成或者有相应规范的，凡需要日期时间表示的地方一律采用UTC时间，遵循ISO 8601，并做以下约束：

- 表示日期一律采用YYYY-MM-DD方式，例如2014-06-01表示2014年6月1日。
- 表示时间一律采用hh:mm:ss方式，并在最后加一个大写字母Z表示UTC时间。例如 23:00:10Z表示UTC时间23点0分10秒。
- 凡涉及日期和时间合并表示时，在两者中间加大写字母T，例如2014-06-01T23:00:10Z表示UTC时间2014年6月1日23点0分10秒。
- 

## 规范化字符串

通常一个字符串中可以包含任何Unicode字符。在编程中这种灵活性会带来不少困扰。因此引入“规范字符串”的概念。一个规范字符串只包含百分号编码字符以及URI (Uniform Resource Identifier) 非保留字符 (Unreserved Characters)。RFC 3986规定URI非保留字符包括以下字符：字母 (A-Z, a-z)、数字 (0-9)、连字号 (-)、点号 (.)、下划线 (\_)、波浪线 (~)。将任意一个字符串转换为规范字符串的方式是：

- 将字符串转换成UTF-8编码的字节流。
- 保留所有URI非保留字符原样不变。
- 对其余字节做一次RFC 3986中规定的百分号编码 (Percent-Encoding)，即一个%后面跟着两个表示该字节值的十六进制字母。字母一律采用大写形式。示例：原字符串：this is an example for 测试，对应的规范字符串：this%20is%20an%20example%20for%E6%B5%8B%E8%AF%95。

## 服务域名

服务端点Endpoint	协议
{多节点版 IEC所在机器IP}:{多节点版 IEC监听端口，默认8702}	HTTP

## 错误码

## 错误码格式

当用户访问API出现错误时，会返回给用户相应的错误码和错误信息，便于定位问题，并做出适当的处理。请求发生错误时通过Response Body返回详细错误信息，遵循如下格式：

参数名	类型	说明
status	int	表示具体错误类型。
msg	String	有关该错误的详细说明。

例如：

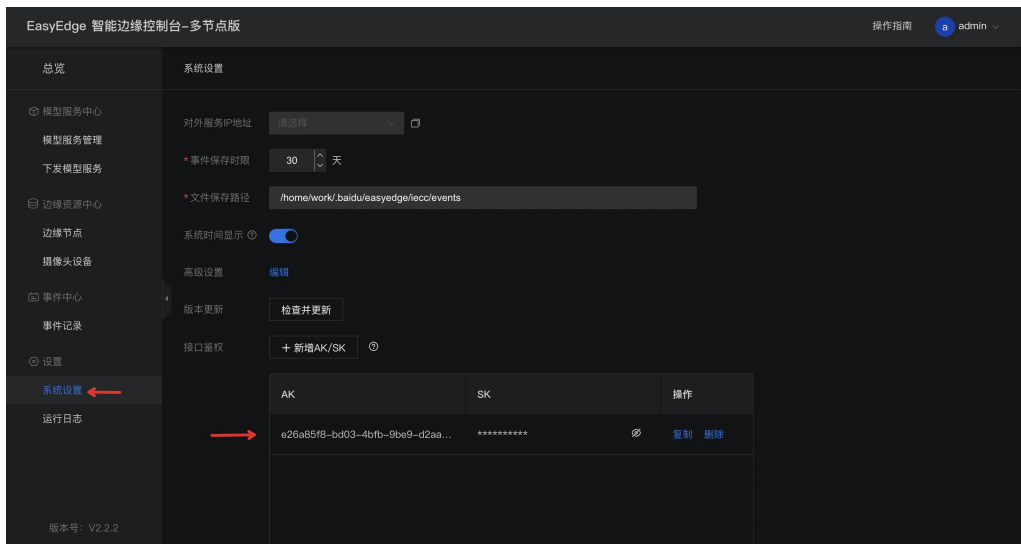
```
{
  "status": 170000,
  "message": "发生错误. Error"
}
```

## 公共请求参数

### 公共请求Query参数

当用户访问API时，需要通过query参数传入access\_token，如 [http://127.0.0.1:8702/iec/iapi/v1/camera/new?access\\_token={access\\_token}](http://127.0.0.1:8702/iec/iapi/v1/camera/new?access_token={access_token})

### 获取生成Access Token用的AK、SK



## 获取Access Token

### 基本信息

**Path**： /auth/v1/token

**Method**： GET

**接口描述**：

### 请求参数

#### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

#### Body

名称	类型	是否必须	默认值	备注	其他信息
grantType	string	必须		固定传入client_credentials	
clientId	string	必须		设置-系统设置-AK	
clientSecret	string	必须		设置-系统设置-SK	

#### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
msg	string	非必须			
status	number	必须			
data	object	非必须			
├ expiresIn	number	必须		过期时间戳，秒级	
├ accessToken	string	必须			

#### 节点相关接口

##### 节点列表

##### 基本信息

**Path** : /iec/iapi/v1/node/list

**Method** : POST

##### 接口描述 :

节点的新增涉及到激活流程，需要在IEC控制中心页面操作。

##### 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
filters	object	非必须			
├ name	string	非必须		搜索包含子串的节点	

##### 返回数据



名称	类型	是否必须	默认值	备注	其他信息
data	object []	非必须			item 类型: object
├ id	number	必须			
├ uuid	string	必须			
├ name	string	非必须			
├ tag	object	非必须			
├ remark	string	非必须			
├ hostname	string	非必须			
├ platform	string	非必须		操作系统	
├ MACAddr	string	非必须		节点的mac地址, 逗号分割	
├ IPAddr	string	非必须		节点的ip地址, 逗号分割	
├ CPUArch	string	非必须		CPU架构	
├ CPUModel	string	非必须		CPU型号	
├ ASICModel	string	非必须		AI芯片的型号, 逗号分割	
├ isActivated	boolean	非必须		是否已经激活	
├ createAt	string	非必须			
msg	string	非必须			
status	number	非必须			

#### 已添加的模型列表

#### 基本信息

**Path** : /iec/iapi/v1/aimodel/list

**Method** : GET

**接口描述** :

#### 请求参数

#### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object []	非必须			item 类型: object
├ id	number	必须			
├ name	string	必须			
├ modelProduct	string	必须			
├ modelName	string	必须			
├ modelType	number	必须		1-分类, 2-检测, 14-语义分割, 6-实例分割	
├ modelSoc	string	必须			
├ modelThresholdRec	number	必须			
├ platform	string	必须			
├ form	string	必须			
├ isEdgeKit	boolean	必须			
├ createAt	string	必须			
msg	string	非必须			
status	number	非必须			

🔗 下发模型为服务

🔗 基本信息

**Path** : /iec/iapi/v1/aimodel/deploy

**Method** : POST

**接口描述** :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		模型id, 可以从aimode/list接口获得	
targetNodes	object []	必须			item 类型: object
├─ id	number	必须		节点id, 可以从node/list接口获得	
├─ activationType	string	必须		ONLINE / OFFLINE	
├─ serialNum	string	非必须		ONLINE 激活时填入序列号	
├─ licenseFileCont	string	非必须		OFFLINE 激活时填入离线激活的license内容	
├─ host	string	非必须		默认为0.0.0.0	
├─ port	number	非必须		默认为0, 也即系统自动选择	
├─ env	array []	非必须			item 类型: array

🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object []	非必须			item 类型: object
├─ id	number	非必须			
├─ uuid	string	非必须			
msg	string	非必须			
status	number	非必须			

🔗 已下发的服务详情

🔗 基本信息

**Path** : /iec/iapi/v1/aiservice/detail

**Method** : POST

**接口描述** :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	非必须			

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object	非必须			
├ id	number	必须			
├ name	string	必须			
├ description	string	必须			
├ uuid	string	必须			
├ platform	string	必须			
├ modelType	number	必须			
├ isEdgeKit	boolean	非必须			
├ soc	string	非必须		支持的AI芯片	
├ modelName	string	非必须			
├ nodeId	number	必须			
├ nodeName	string	非必须			
├ CPUArch	string	非必须			
├ host	string	必须			
├ port	number	必须			
├ serviceStatus	string	必须			枚举: ERROR,RUNNING,DEPLOYING,DEPLOY_FAILED,OVERDUE,PAUSED,ERROR
├ serviceTip	string	必须			
├ isServiceOn	boolean	非必须			
├ serviceType	string	必须		process / container	
├ createAt	string	非必须			
├ updateAt	string	非必须			
msg	string	非必须			
status	number	非必须			

## 启动服务

## 基本信息

**Path** : /iec/iapi/v1/aiservice/start

**Method** : POST

**接口描述** :

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		模型id	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

## 已下发的服务列表

## 基本信息

**Path** : /iec/iapi/v1/aiservice/list

**Method** : POST

## 接口描述 :

接口字段与“已下发的服务详情”一致

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object []	非必须			item 类型: object
├ id	number	必须			
├ name	string	必须			
├ description	string	必须			
├ uuid	string	必须			
├ platform	string	必须			
├ modelType	number	必须			
├ isEdgeKit	boolean	必须			
├ soc	string	必须			
├ modelName	string	必须			
├ nodeId	number	必须			
├ nodeName	string	必须			
├ CPUArch	string	必须			
├ host	string	必须			
├ port	number	必须			
├ serviceStatus	string	必须			
├ serviceTip	string	必须			
├ isServiceOn	boolean	必须			
├ serviceType	string	必须			
├ createAt	string	必须			
├ updateAt	string	必须			
msg	string	非必须			
status	number	非必须			

## 停止服务

🔗 基本信息

**Path** : /iec/iapi/v1/aiservice/stop

**Method** : POST

接口描述 :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须			

🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

🔗 删除服务

🔗 基本信息

**Path** : /iec/iapi/v1/aiservice/delete

**Method** : POST

接口描述 :

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须			

🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

🔗 更新服务

🔗 基本信息

**Path** : /iec/iapi/v1/aiservice/update

**Method** : POST

接口描述 :

更新服务后,会自动触发服务重启

🔗 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须			
host	string	非必须			
port	number	非必须			
env	array []	非必须			item 类型: array

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
----	----	------	-----	----	------

## 服务请求校验

## 基本信息

**Path** : /iec/iapi/v1/aiservice/demo

**Method** : POST

**接口描述** : 单次请求下发的服务。返回体 data 字段的内容即为原始服务的返回内容。可参考 : <https://ai.baidu.com/ai-doc/EASYDL/1k3qy99te#预测图像>

## 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须			
image	string	必须		图像的base64编码	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
data	object	非必须			
├ error_code	number	非必须			
├ cost_ms	number	非必须			
├ results	object []	非必须			item 类型: object
├─ confidence	number	必须			
├─ index	number	必须			
├─ label	string	必须			
├─ x1	number	必须			
├─ x2	number	必须			
├─ y1	number	必须			
├─ y2	number	必须			
msg	string	非必须			
status	number	非必须			

### 🔗 摄像头设备相关接口

#### 🔗 校验摄像头

#### 🔗 基本信息

**Path** : /iec/iapi/v1/camera/check-validity

**Method** : POST

**接口描述** :

#### 🔗 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
nodeId	number	必须		节点ID	
cameraAddr	string	必须		摄像头地址	
cameraType	string	必须		摄像头类型, IP表示网络摄像头, LOCAL表示本地摄像头	枚举: IP,LOCAL

#### 🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	boolean	必须		摄像头是否合法	

### 🔗 新增摄像头

#### 🔗 基本信息

**Path** : /iec/iapi/v1/camera/new

**Method** : POST

**接口描述** :

#### 🔗 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
nodeId	number	必须		节点ID	
name	string	必须		摄像头名称	
camAddr	string	非必须		摄像头地址，IP或LOCAL时必填	
camType	string	必须		摄像头类型，IP表示网络摄像头，LOCAL表示本地摄像头，ONVIF表示ONVIF摄像头，GB28181表示国标摄像头	枚举: LOCAL,IP,ONVIF,GB28181
onvifConfig	object	非必须		ONVIF时必填	
├─ ip	string	必须		IP	
├─ port	number	必须		端口	
├─ username	string	必须		用户名	
├─ password	string	必须		密码	
gb28181Config	object	非必须		GB28181时必填	
├─ sipName	string	必须		信令服务器用户名	
├─ sipDeviceId	string	必须		信令服务器设备ID	
├─ sipPassword	string	必须		信令服务器密码	
remark	string	必须		摄像头备注	
tag	array []	必须		标签	item 类型: array
├─		非必须			
├─		非必须			
timeRange	array []	必须		摄像头运行区间	item 类型: array
├─		非必须			
├─		非必须		秒	
frameExtract	object	必须		抽帧配置	
├─ everySecond	number	必须		每多少秒	
├─ frames	number	必须		抽多少帧	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	object	必须			
├─ id	number	必须		当前节点唯一ID	
├─ uuid	string	必须		全局唯一ID	

## 摄像头列表

## 基本信息

**Path** : /iec/iapi/v1/camera/list

**Method** : POST



## 接口描述：

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
pageNo	number	非必须		页数	
pageSize	number	非必须		每页数量	
nodeId	number	非必须		节点ID	
filters	object	非必须		筛选项	
├ name	string	非必须		按摄像头名称筛选	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	object []	必须			item 类型: object
├ id	number	必须		摄像头ID	
├ uuid	string	必须		摄像头全局唯一ID	
├ nodeId	number	必须		节点ID	
├ name	string	必须		摄像头名称	
├ remark	string	必须		摄像头备注	
├ cameraType	string	必须		摄像头类型	枚举: LCOAL,IP,ONVIF,GB28181
├ cameraStatus	string	必须		摄像头状态	枚举: RUNNING,ERROR,STOPPED

## 摄像头详情

## 基本信息

Path : /iec/iapi/v1/camera/get

Method : GET

## 接口描述：

## 请求参数

## Query

参数名称	是否必须	示例	备注
id	是		摄像头ID

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object	必须			
├ id	number	必须		摄像头ID	
├ uuid	string	必须		摄像头全局唯一ID	
├ nodeId	number	必须		节点ID	
├ name	string	必须		摄像头名称	
├ remark	string	必须		摄像头备注	
├ cameraType	string	必须		摄像头类型	枚举: LCOAL,IP,ONVIF,GB28181
├ cameraAddr	string	必须		摄像头地址, IP或LOCAL时有意义	
├ onvifConfig	object	非必须			
├ ip	string	必须		IP	
├ port	number	必须		端口	
├ username	string	必须		用户名	
├ password	string	必须		密码	
├ gb28181Config	object	非必须			
├ sipName	string	必须		信令服务器用户名	
├ sipDeviceId	string	必须		信令服务器设备ID	
├ sipPassword	string	必须		信令服务器密码	
├ frameExtractInterval	number	必须		每多少秒	
├ frameExtractNum	number	必须		抽多少帧	
├ onlineTime	array []	必须		摄像头运行区间	item 类型: array
├		非必须			
├		非必须		秒	
├ cameraStatus	string	必须		摄像头状态	枚举: RUNNING,ERROR,STOPPED

#### 更新摄像头

#### 基本信息

**Path** : /iec/iapi/v1/camera/update

**Method** : POST

**接口描述** :

#### 请求参数

##### Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

##### Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	
name	string	必须		摄像头名称	
camAddr	string	非必须		摄像头地址，IP或LOCAL时必填	
camType	string	必须		摄像头类型，IP表示网络摄像头，LOCAL表示本地摄像头，ONVIF表示ONVIF摄像头，GB28181表示国标摄像头	枚举: LOCAL,IP,ONVIF,GB28181
onvifConfig	object	非必须		ONVIF时必填	
├─ ip	string	必须		IP	
├─ port	string	必须		端口	
├─ username	string	必须		用户名	
├─ password	string	必须		密码	
gb28181Config	object	非必须		GB28181时必填	
├─ sipName	string	必须		信令服务器用户名	
├─ sipDeviceId	string	必须		信令服务器设备ID	
├─ sipPassword	string	必须		信令服务器密码	
remark	string	必须		摄像头备注	
tag	array []	必须		标签	item 类型: array
├─		非必须			
├─		非必须			
timeRange	array []	必须		摄像头运行区间	item 类型: array
├─		非必须			
├─		非必须		秒	
frameExtract	object	必须		抽帧配置	
├─ everySecond	number	必须		每多少秒	
├─ frames	number	必须		抽多少帧	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		修改是否成功	

## 启动摄像头

## 基本信息

**Path** : /iec/iapi/v1/camera/start

**Method** : POST

**接口描述** :

## 🔗 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		开启摄像头是否成功	

## 🔗 停止摄像头

## 🔗 基本信息

**Path** : /iec/iapi/v1/camera/stop

**Method** : POST

**接口描述** :

## 🔗 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		关闭摄像头是否成功	

## 🔗 删除摄像头

## 🔗 基本信息

**Path** : /iec/iapi/v1/camera/delete

**Method** : POST

**接口描述** :

## 🔗 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		摄像头ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		删除摄像头是否成功	

## 物联网设备相关接口

## 新增设备

## 基本信息

**Path** : /iec/iapi/v1/plc/new

**Method** : POST

**接口描述** :

## 请求参数

**Headers**

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

**Body**

名称	类型	是否必须	默认值	备注	其他信息
nodeId	number	必须		节点ID	
name	string	必须		设备名称	
deviceType	string	必须		设备类型	枚举: modbus,opcua
modbus	object	非必须		modbus时必须填	
├─ protocol	string	必须		协议类型	枚举: tcp,rtu
├─ rtu	object	非必须		rtu时必须填	
├─ port	string	必须		端口	
├─ baudrate	number	必须		波特率	
├─ databit	number	必须		数据位	枚举: 5,6,7,8
├─ stopbit	number	必须		停止位	枚举: 1,2
├─ parity	string	必须		校验位	枚举: N,E,O
├─ tcp	object	非必须		tcp时必须填	
├─ address	string	必须		连接地址	

port	number	必须	端口	
slaveId	number	必须	从站号	
interval	number	必须	采样间隔	
opcua	object	非必须	opcua时必须填	
endpoint	string	必须	通道地址	
securityPolicy	string	必须	安全策略	枚举: None,Basic256Sha256,Aes128Sha256RsaOaep,Aes256Sha256RsaPss
securityMode	string	必须	安全模式	枚举: None,Sign,SignAndEncrypt
certificate	string	非必须	数字证书	
privateKey	string	非必须	密钥证书	
username	string	非必须	用户名	
password	string	非必须	密码	
timeout	number	必须	连接超时时间	
interval	number	必须	采样间隔	
remark	string	必须	设备备注	
tag	array []	必须	标签	item 类型: array
		非必须		
		非必须		
attributes	object []	必须	属性	item 类型: object
id	string	必须	标识符	
name	string	必须	属性名	
type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,string
defaultValue	string	非必须	默认值	
unit	string	必须	单位	
required	boolean	必须	是否必填	
properties	object []	必须	测点	item 类型: object
id	string	必须	标识符	
name	string	必须	测点名	
type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,string
mode	string	必须	读写类型	枚举: ro,rw
unit	string	必须	单位	
modbus	object	非必须	modbus时必须填	

└─ func	number	必须		寄存器类型：1.线圈寄存器 2.离散输入寄存器 3.保持寄存器 4.输入寄存器	枚举: 1,2,3,4
└─ address	string	必须		寄存器地址	
└─ quantity	number	必须		寄存器数量	
└─ opcua	object	非必须		opcua时必须填	
└─ nodeid	string	必须		节点ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	object	必须			
└─ id	number	必须		当前节点唯一ID	
└─ uuid	string	必须		全局唯一ID	

## 设备列表

## 基本信息

**Path** : /iec/iapi/v1/plc/list

**Method** : POST

**接口描述** :

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
pageNo	number	非必须		页数	
pageSize	number	非必须		每页数量	
nodeId	number	非必须		节点ID	
filters	object	非必须		筛选项	
└─ name	string	非必须		按名称筛选	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object []	必须			item 类型: object
├ id	number	必须		设备ID	
├ uuid	string	必须		设备全局唯一ID	
├ nodeId	number	必须		节点ID	
├ name	string	必须		设备名称	
├ remark	string	必须		设备备注	
├ deviceType	string	必须		设备类型	
├ status	string	必须		设备状态	枚举: RUNNING,ERROR,STOPPED

### 🔗 设备详情

### 🔗 基本信息

**Path :** /iec/iapi/v1/plc/get

**Method :** GET

**接口描述 :**

### 🔗 请求参数

**Query**

参数名称	是否必须	示例	备注
id	是		设备ID

### 🔗 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	object	必须			
├ id	number	必须		设备ID	
├ uuid	string	必须		设备全局唯一ID	
├ nodeId	number	必须		节点ID	
├ name	string	必须		设备名称	
├ deviceType	string	必须		设备类型	枚举: modbus,opcua
├ modbus	object	非必须		modbus时必须填	
├ protocol	string	必须		协议类型	枚举: tcp,rtu
├ rtu	object	非必须		rtu时必须填	
├ port	string	必须		端口	
├ baudrate	number	必须		波特率	
├ databit	number	必须		数据位	枚举: 5,6,7,8
├ stopbit	number	必须		停止位	枚举: 1,2
├ parity	string	必须		校验位	枚举: N,E,O
├ tcp	object	非必须		tcp时必须填	
├					



address	string	必须	连接地址	
└─ port	number	必须	端口	
└─ slaveId	number	必须	从站号	
interval	number	必须	采样间隔	
└─ opcua	object	非必须	opcua时必须填	
└─ endpoint	string	必须	通道地址	
└─ securityPolicy	string	必须	安全策略	
└─ securityMode	string	必须	安全模式	
└─ certificate	string	非必须	数字证书	
└─ privateKey	string	非必须	密钥证书	
└─ username	string	非必须	用户名	
└─ password	string	非必须	密码	
└─ timeout	number	必须	连接超时时间	
└─ interval	number	必须	采样间隔	
└─ remark	string	必须	设备备注	
└─ tag	array []	必须	标签	item 类型: array
└─		非必须		
└─		非必须		
└─ attributes	object []	必须	属性	item 类型: object
└─ id	string	必须	标识符	
└─ name	string	必须	属性名	
└─ type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,s tring
└─ defaultValue	string	非必须	默认值	
└─ unit	string	必须	单位	
└─ required	boolean	必须	是否必填	
└─ properties	object []	必须	测点	item 类型: object
└─ id	string	必须	标识符	
└─ name	string	必须	测点名	
└─ type	string	必须	类型	枚举: bool,int16,int32,int64,float32,float64,s tring
└─ mode	string	必须	读写类型	枚举: ro,rw
└─ unit	string	必须	单位	
└─ modbus	object	非必须	modbus时必须填	

安方器米刑 · 1 线圈安方器 · 离散输入安方器 · 保持安方

func	number	必须	寄存器地址	枚举: 1,2,3,4
address	string	必须	寄存器地址	
quantity	number	必须	寄存器数量	
opcua	object	非必须	opcua时必填	
nodeid	string	必须	节点ID	
status	string	必须	设备状态	枚举: RUNNING,ERROR,STOPPED
url	string	非必须	测点数据获取websocket连接地址	
realData	object	非必须	当前测点数据	

更新设备

基本信息

Path : /iec/iapi/v1/plc/update

Method : POST

接口描述 :

请求参数

Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		设备ID	
name	string	必须		设备名称	
deviceType	string	必须		设备类型	枚举: modbus,opcua
modbus	object	非必须		modbus时必填	
protocol	string	必须		协议类型	枚举: tcp,rtu
rtu	object	非必须		rtu时必填	
port	string	必须		端口	
baudrate	number	必须		波特率	
databit	number	必须		数据位	枚举: 5,6,7,8
stopbit	number	必须		停止位	枚举: 1,2
parity	string	必须		校验位	枚举: N,E,O
tcp	object	非必须		tcp时必填	
address	string	必须		连接地址	
port	number	必须		端口	
slaveld	number	必须		从站号	

└─ interval	number	必须		采样间隔	
opcua	object	非必须		opcua时必须填	
└─ endpoint	string	必须		通道地址	
└─ securityPolicy	string	必须		安全策略	枚举: None, Basic256Sha256, Aes128Sha256RsaOaep, Aes256Sha256RsaPss
└─ securityMode	string	非必须		安全模式	枚举: None, Sign, SignAndEncrypt
└─ certificate	string	非必须		数字证书	
└─ privateKey	string	非必须		密钥证书	
└─ username	string	非必须		用户名	
└─ password	string	必须		密码	
└─ timeout	number	必须		连接超时时间	
└─ interval	number	必须		采样间隔	
remark	string	必须		设备备注	
tag	array []	必须		标签	item 类型: array
└─		非必须			
└─		非必须			
attributes	object []	必须		属性	item 类型: object
└─ id	string	必须		标识符	
└─ name	string	必须		属性名	
└─ type	string	必须		类型	枚举: bool, int16, int32, int64, float32, float64, string
└─ defaultValue	string	非必须		默认值	
└─ unit	string	必须		单位	
└─ required	boolean	必须		是否必填	
properties	object []	必须		测点	item 类型: object
└─ id	string	必须		标识符	
└─ name	string	必须		测点名	
└─ type	string	必须		类型	枚举: bool, int16, int32, int64, float32, float64, string
└─ mode	string	必须		读写类型	枚举: ro, rw
└─ unit	string	必须		单位	
└─ modbus	object	非必须		modbus时必须填	
└─ func	number	必须		寄存器类型: 1.线圈寄存器 2.离散输入寄存器 3.保持寄存器 4.输入寄存器	枚举: 1,2,3,4
└─	string	必须		寄存器地址	

address	string	必须		寄存器地址	
quantity	number	必须		寄存器数量	
opcua	object	非必须		opcua时必填	
nodeid	string	必须		节点ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	boolean	必须		修改是否成功	

## 删除设备

## 基本信息

**Path :** /iec/iapi/v1/plc/delete

**Method :** POST

**接口描述 :**

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		设备ID	

## 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码, 0表示成功	
msg	string	必须		错误信息, status不为0时有意义	
data	boolean	必须		删除设备是否成功	

## 写测点

## 基本信息

**Path :** /iec/iapi/v1/plc/property/write

**Method :** POST

**接口描述 :**

## 请求参数

## Headers

参数名称	参数值	是否必须	示例	备注
Content-Type	application/json	是		

## Body

名称	类型	是否必须	默认值	备注	其他信息
id	number	必须		设备ID	
propertyId	string	必须		测点标识符	
propertyValue	any	必须		测点值	

#### 返回数据

名称	类型	是否必须	默认值	备注	其他信息
status	number	必须		状态码，0表示成功	
msg	string	必须		错误信息，status不为0时有意义	
data	boolean	必须		修改测点值是否成功	

#### 功能更新记录

时间	版本	说明
2023-05-25	1.0.0	第一版