

# BODY 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
产品价格	5
API文档	15
简介	15
调用方式	17
人体关键点识别	19
人体检测和属性识别	28
请求说明	29
人流量统计	35
手势识别	38
人像分割	43
驾驶行为分析	48
人流量统计-动态版	53
手部关键点识别	59
指尖检测（邀测）	64
错误码	66
错误码	68
私有化部署	70
整体介绍	70
部署说明	76
接口调用	81
常见问题	118
更新日志	121
导览	121
在线API更新日志	121
历史文档	124
C++ SDK文档	124
简介	124
快速入门	125
人体关键点识别	126
人体检测与属性识别	132
人流量统计	139
手势识别	141
人像分割	144
驾驶行为分析	147
人流量统计-动态版	151
手部关键点识别	156
错误码	159
C# SDK文档	161
简介	161

快速入门	162
人体关键点识别	163
人体检测与属性识别	171
人流量统计	178
手势识别	179
人像分割	183
驾驶行为分析	186
人流量统计-动态版	190
错误码	195
PHP SDK文档	197
简介	197
快速入门	198
人体关键点识别	199
人体检测与属性识别	205
人流量统计	212
手势识别	213
人像分割	217
驾驶行为分析	220
人流量统计-动态版	224
手部关键点识别	229
错误码	232
Python SDK文档	234
简介	234
快速入门	235
人体关键点识别	236
人体检测与属性识别	242
人流量统计	249
手势识别	251
人像分割	254
驾驶行为分析	257
人流量统计-动态版	261
手部关键点识别	266
错误码	270
Node.js SDK文档	272
简介	272
快速入门	273
人体关键点识别	274
人体检测与属性识别	280
人流量统计	288
手势识别	290
人像分割	293



驾驶行为分析	296
人流量统计-动态版	300
手部关键点识别	305
错误码	309
Java SDK文档	311
简介	311
快速入门	312
人体关键点识别	315
人体检测与属性识别	321
人流量统计	328
手势识别	330
人像分割	333
驾驶行为分析	336
人流量统计-动态版	340
手部关键点识别	345
错误码	349
人像分割-证件照版 (邀测)	351

# 产品价格

## 计费简介

人体分析已支持在线付费的接口有：人体关键点识别、人体检测与属性识别、人流量统计、人流量统计（动态版）、人像分割、手势识别、驾驶行为分析、手部关键点识别。在线接口主要有3种计费方式：

- **按调用量后付费**：免费测试量用完后，可选择开通 **按调用量后付费** 的计费方式付费使用。在 **控制台** 直接开通后，默认按调用量后付费的方式进行阶梯计费，QPS升至10。当前全部8个接口均可采用按调用量后付费的方式。
- **购买次数包**：免费测试量用完后，也可选择购买**次数包** 方式付费使用，在 **控制台** 即可直接购买对应服务的次数包，购买后QPS升至10。若次数包额度消耗完毕后自动转为按调用量后付费方式。当前**驾驶行为分析、手部关键点识别**2个接口采用**次数包购买**的方式。
- **按QPS（并发量）计费**：如果您通过百度云的个人认证，免费测试QPS为2；通过企业认证，免费测试量QPS最高将扩充至5。同时您可以根据业务需求随时购买扩充QPS，QPS可包月购买，也可按天单独购买，灵活多样，适应多场景需求。当前**人体关键点识别、人体检测与属性识别、人流量统计、人像分割、手势识别**5个接口支持**在线购买QPS**，购买QPS后调用量无限制。该计费方式与按调用量后付费方式互斥，如需切换应将原计费方式停止或失效。注意：同一个账号下多个应用共享接口QPS。

说明：QPS（query per second）指每秒向服务发送的请求数量峰值，相当于每个API每秒可以允许请求的最大上限数量。

## 免费测试资源

在 [控制台-免费测试资源领取页](#) 可领取所需接口的免费测试资源.人体方向免费资源领取需完成个人或企业实名认证，企业认证可领取更多测试资源，[了解个人/企业认证](#)。

人体分析各个接口都具有免费测试资源（调用量和QPS并发），具体免费额度如下：

产品服务	个人认证免费测试量	企业认证免费测试量	有效期
人体关键点识别	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
人体检测与属性识别	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
人流量统计	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
人流量统计(动态版)	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
人像分割	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
手势识别	调用量：总共10000次， QPS：2QPS	调用量：总共50000次， QPS：5QPS	365天
驾驶行为分析	调用量：总共5000次， QPS：2QPS	调用量：总共10000次， QPS：2QPS	365天
手部关键点识别	调用量：总共5000次， QPS：2QPS	调用量：总共10000次， QPS：2QPS	365天

注：

- 成功调用与失败调用均消耗免费测试量。

- 如有作弊或违禁使用情况，百度有权停止其使用免费测试量。

## 🔗 价目表

### 🔗 1. 按调用量后付费

当前人体关键点识别、人体检测与属性识别、人流量统计、人流量统计-动态版、人像分割、手势识别、驾驶行为分析、手部关键点识别8个接口均可采用按调用量后付费的方式。开通按调用量后付费后，默认保证10QPS并发。

#### 🔗 1.1 人体关键点识别接口

人体关键点识别接口：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	人体关键点识别接口 (元/千次)
0<月调用量<=5	2.0
5<月调用量<=30	1.8
30<月调用量<=100	1.6
100<月调用量<=300	1.4
300<月调用量	1.0

说明：按调用量后付费与按QPS购买方式互斥。如需切换至按QPS购买，您可在控制台自助点击停止按量后付费。

#### 🔗 1.2 人体检测与属性识别接口

人体检测与属性识别接口：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	人体检测与属性识别接口 (元/千次)
0<月调用量<=5	2.0
5<月调用量<=30	1.8
30<月调用量<=100	1.6
100<月调用量<=300	1.4
300<月调用量	1.0

说明：按调用量后付费与按QPS购买方式互斥。如需切换至按QPS购买，您可在控制台自助点击停止按量后付费。

#### 🔗 1.3 人流量统计接口

人流量统计接口：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	人流量统计接口 (元/千次)
0<月调用量<=5	2.0
5<月调用量<=30	1.8
30<月调用量<=100	1.6
100<月调用量<=300	1.4
300<月调用量	1.0

说明：按调用量后付费与按QPS购买方式互斥。如需切换至按QPS购买，您可在控制台自助点击停止按量后付费。

#### 1.4 人流量统计接口-动态版

**人流量统计-动态版接口**：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	人流量统计接口-动态版 (元/千次)
0<月调用量<=5	2.0
5<月调用量<=30	1.8
30<月调用量<=100	1.6
100<月调用量<=300	1.4
300<月调用量	1.0

#### 1.5 人像分割接口

**人像分割接口**：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	人像分割接口 (元/千次)
0<月调用量<=5	4.0
5<月调用量<=30	3.6
30<月调用量<=100	3.2
100<月调用量<=300	2.8
300<月调用量	2.0

说明：按调用量后付费与按QPS购买方式互斥。如需切换至按QPS购买，您可在控制台自助点击停止按量后付费。

#### 1.6 手势识别接口

**手势识别接口**：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后5万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	手势识别接口 (元/千次)
0<月调用量<=5	2.0
5<月调用量<=30	1.8
30<月调用量<=100	1.6
100<月调用量<=300	1.4
300<月调用量	1.0

说明：按调用量后付费与按QPS购买方式互斥。如需切换至按QPS购买，您可在控制台自助点击停止按量后付费。

### 1.7 驾驶行为分析接口

**驾驶行为分析接口**：采用分段阶梯定价，调用单价按照自然月累积调用量所落阶梯区间而变化。超出本阶梯部分按照下一阶梯单价计算，调用量越大，单价越低。月初，上月累积的调用量清零，重新开始累积本月调用量。

企业认证后1万次免费测试量，用尽后开始计费，价格如下：

月调用量 (万次)	驾驶行为分析接口 (元/千次)
0<月调用量<=30	2.0
30<月调用量<=100	1.6
100<月调用量<=200	1.2
200<月调用量<=500	1.0
500<月调用量	0.8

说明：调用失败不计费。如有购买对应服务的次数包，则优先消耗次数包额度，抵扣完毕后自动转为按量后付费方式。

### 1.8 手部关键点识别接口

**手部关键点识别接口**：采用单一定价，企业认证后1万次免费测试量，用尽后开始计费，价格如下：

手部关键点识别接口 (元/千次)
6.0

说明：调用失败不计费。如有购买对应服务的次数包，则优先消耗次数包额度，抵扣完毕后自动转为按量后付费方式。

## 2. 购买次数包

当前**驾驶行为分析**、**手部关键点识别**免费测试调用限额用完后，可在控制台选择购买**次数包**，购买次数包费后，默认保证10QPS并发。

可根据业务量级一次性购买对应规格的**次数包**，**购买后12个月内有效**。有效期内产生计费的调用量，系统每小时将优先抵扣次数包额度，从已购买的次数包中按照购买时间顺序、以及次数包规格由小至大依次扣除相应次数；抵扣完毕后将自动转为按调用量后付费方式继续计费。

超出有效期未抵扣的**次数包额度**自动失效，无法继续使用。

### 2.1 驾驶行为分析接口

规格 (次)	驾驶行为分析接口 (单位 : 元)
30万	500
100万	1200
200万	2000
500万	4000
1000万	7000

### 2.1 手部关键点识别接口

规格 (次)	手部关键点识别接口 (单位 : 元)
5万	275
10万	500
30万	1350
100万	4000
200万	7000
500万	15000

### 3. 按QPS购买

当前人体关键点识别、人体检测与属性识别、人流量统计、人像分割、手势识别5个接口支持在线购买QPS。

购买QPS后，无论是否通过企业认证，调用量额度都会升级为无限制。购买QPS与按调用量后付费方式互斥，如需从按QPS预付费切换只按调用量后付费，您需等待所购QPS失效或完成退款后操作。

QPS可按月、按天灵活购买扩充，具体价格如下：

购买QPS数量	按月购买	按天购买
0<QPS<=10	500元/月/QPS	50元/天/QPS
10<QPS<=50	400元/月/QPS	40元/天/QPS
50<QPS<=100	300元/月/QPS	30元/天/QPS
100<QPS	200元/月/QPS	20元/天/QPS

#### 说明：

- 1、阶梯价格梯度按照单次购买的数量计算，不同订单不累计。
- 2、每个接口单独计算QPS，若您需要人体关键点识别、人像分割接口各10个QPS，则您需要分别购买10个QPS的人体关键点识别和人像分割接口（企业认证用户免费赠送5 QPS）。
- 3、驾驶行为分析、手部关键点识别、人流量统计（动态版）3个接口暂不支持QPS购买，如需更多QPS，请[提交工单](#)与我们联系。

**示例一**：在测试期内，如果您需要20个QPS，除测试赠送的5个QPS（需要企业认证），还需购买15个QPS。则计费方式为： $500 \times 10 + 400 \times 5 = 7000$ 元/月

**示例二**：如果您需要购买1个月零5天的2个QPS，可以按照包月+按天的搭配模式购买，即： $500 \times 2 + 50 \times 5 \times 2 = 1500$ 元

## 🔗 操作指南

### 🔗 1. 开通按调用量后付费

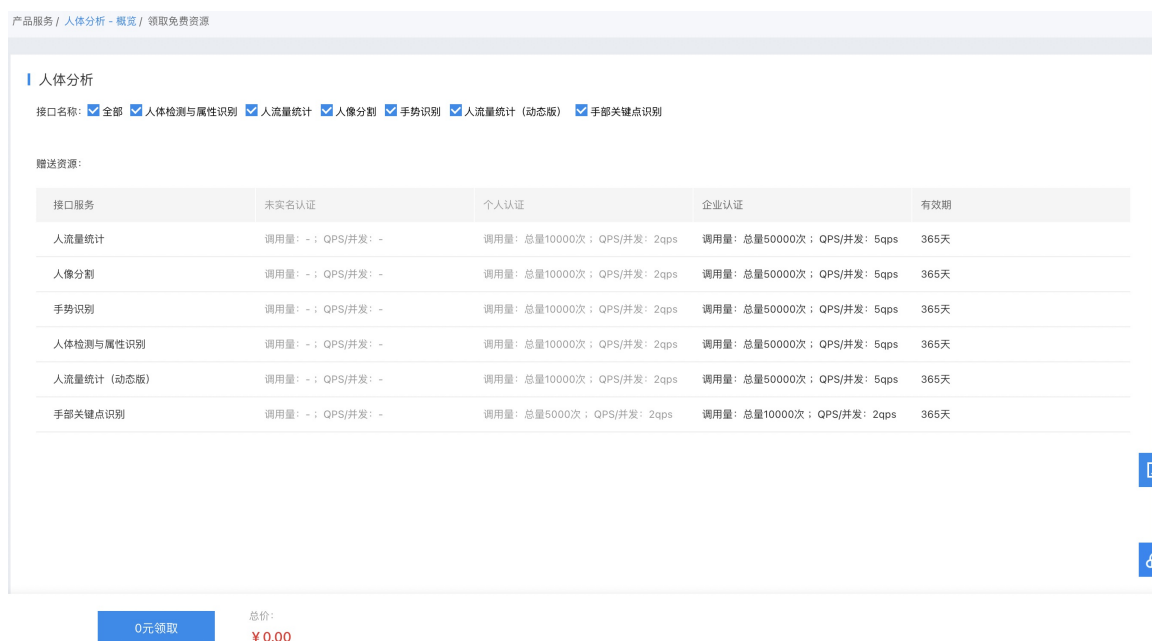
请按照如下流程，使用人体分析在线服务并开通计费。

**Step1 个人/企业认证：**您前往用户中心，对账号进行[个人认证](#)或[企业认证](#)。认证成功后，您将可以领取免费测试资源。若您已完成个人/企业认证，可忽略该步骤。

**Step2 进入人体分析控制台：**点击“控制台”，选择“人体分析”进入[人体分析概览页](#)。



**Step3 领取免费测试资源：**点击“领取免费资源”，进入[免费测试资源领取页](#)选择您需要接口服务领取，领取后即可生效，有效期365天。



**Step4 选择接口，开通计费：**选择您需要计费的接口，确认开通。

**可用服务列表**

完成企业认证，人体分析支持付费购买QPS的接口免费QPS由'2'扩充至'5'。[您已完成企业认证] [不再提醒](#)

API	状态	调用量限制	QPS限制	开通按量后付费 <a href="#">?</a>	购买次数包 <a href="#">?</a>
驾驶行为分析	● 免费使用	剩余免费10000次	不保证并发	开通 <a href="#">?</a>	购买
人体关键点识别	● 免费使用	无限制	5	购买QPS   <a href="#">配额详情</a>	--
人体检测与属性识别	● 免费使用	无限制	5	购买QPS   <a href="#">配额详情</a>	--
人流量统计	● 免费使用	无限制	5	购买QPS   <a href="#">配额详情</a>	--
人像分割	● 免费使用	无限制	5	购买QPS   <a href="#">配额详情</a>	--
手势识别	● 免费使用	无限制	5	购买QPS   <a href="#">配额详情</a>	--
人流量统计(动态版)	● 免费使用	50000次/天免费	不保证并发	免费试用	--
手部关键点识别	● 免费使用	剩余免费10000次	不保证并发	开通 <a href="#">?</a>	购买
危险行为识别	● 免费使用	50次/天免费	不保证并发	免费试用	--

**开通付费**

产品服务 / 人体分析 - 概览 / 开通付费

1 选择开通接口 > 2 确认订单 > 3 开通成功

\* 选择服务： 驾驶行为分析  手部关键点识别

价目表参考如下（开通付费后，默认使用按量后付费形式进行计费 [?](#)）

月调用量 (n)	0<n≤30w	30w<n≤100w	100w<n≤200w	200w<n≤500w	500w<n
API					
驾驶行为分析	0.002元/次	0.0016元/次	0.0012元/次	0.001元/次	0.0008元/次

注：“w”表示万次，“调用量”指使用按量后付费形式产生的计费调用量

[确认开通](#)

## ● 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，超过额度的请求系统将不再响应，且不再保证并发处理。

## 2. 购买次数包

请按照如下流程，使用人体分析在线服务并购买对应规格的次数包。

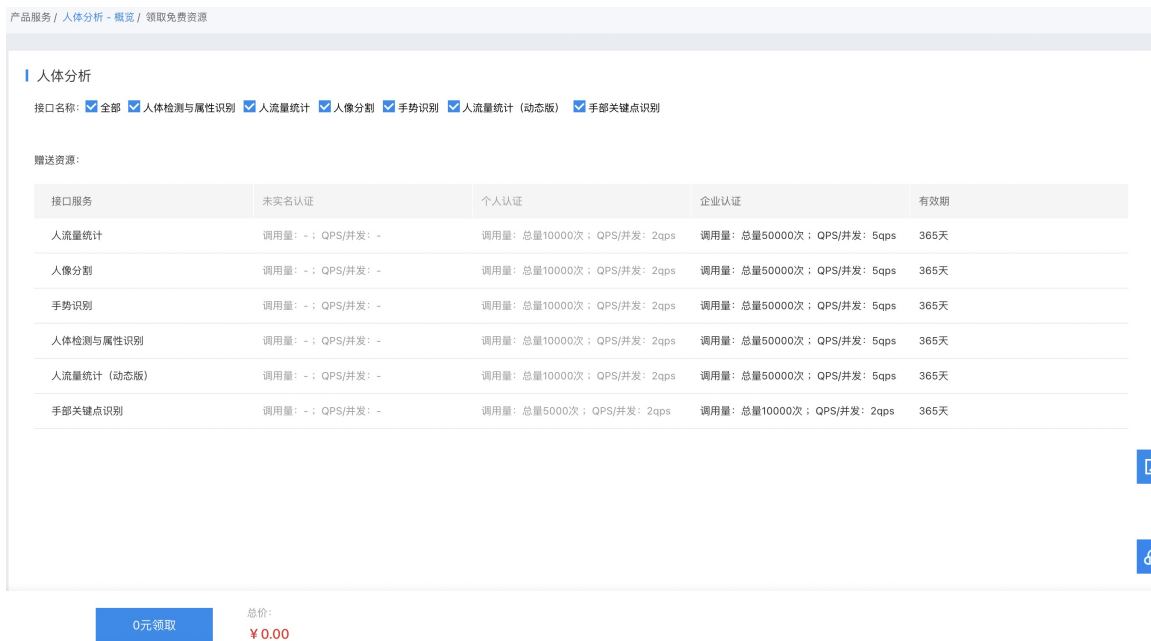
**Step1 个人/企业认证：**您前往用户中心，对账号进行[个人认证](#)或[企业认证](#)。认证成功后，您将可以领取免费测试资源。若您已完成个人/企业认证，可忽略该步骤。

**Step2 进入人体分析控制台：**点击“控制台”，选择“人体分析”进入[人体分析概览页](#)。



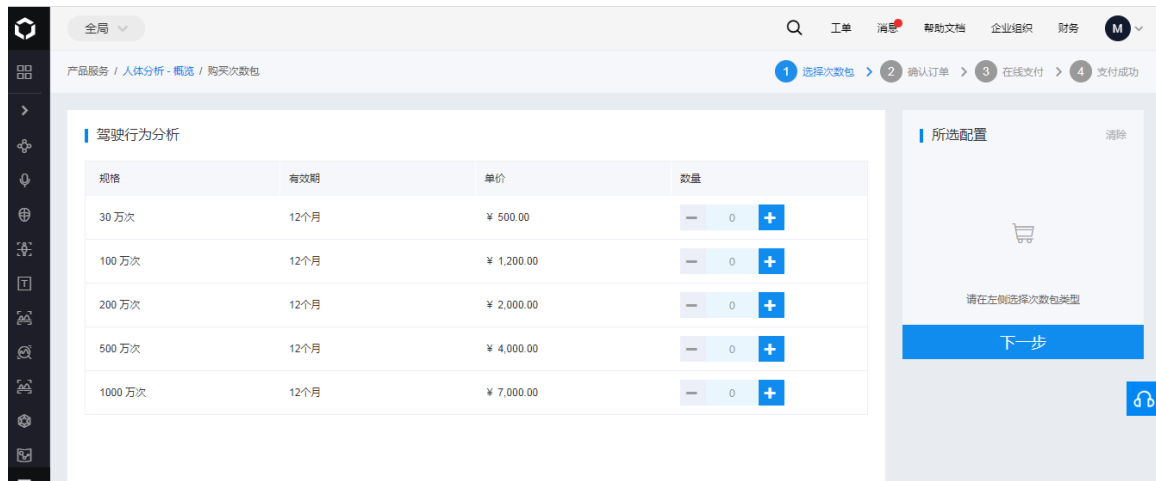


**Step3 领取免费测试资源：** 点击“领取免费资源”，进入[免费测试资源领取页](#)选择您需要接口服务领取，领取后即可生效，有效期365天。



**Step4 选择接口，购买所需规格的次数包：** 根据需要选择次数包购买。





### 3. 购买QPS

请按照如下流程，使用人体分析在线服务并购买QPS。

**Step1 个人/企业认证：**您前往用户中心，对账号进行[个人认证](#)或[企业认证](#)。认证成功后，您将可以领取免费测试资源。若您已完成个人/企业认证，可忽略该步骤。

**Step2 进入人体分析控制台：**点击“控制台”，选择“人体分析”进入[人体分析概览页](#)。



**Step3 领取免费测试资源：**点击“领取免费资源”，进入[免费测试资源领取页](#)选择您需要接口服务领取，领取后即可生效，有效期365天。

产品服务 / 人体分析 - 概览 / 领取免费资源

### 人体分析

接口名称: 全部 人体检测与属性识别 人流量统计 人像分割 手势识别 人流量统计(动态版) 手部关键点识别

赠送资源:

接口服务	实名认证	个人认证	企业认证	有效期
人流量统计	调用量: -; QPS/并发: -	调用量: 总量10000次; QPS/并发: 2qps	调用量: 总量50000次; QPS/并发: 5qps	365天
人像分割	调用量: -; QPS/并发: -	调用量: 总量10000次; QPS/并发: 2qps	调用量: 总量50000次; QPS/并发: 5qps	365天
手势识别	调用量: -; QPS/并发: -	调用量: 总量10000次; QPS/并发: 2qps	调用量: 总量50000次; QPS/并发: 5qps	365天
人体检测与属性识别	调用量: -; QPS/并发: -	调用量: 总量10000次; QPS/并发: 2qps	调用量: 总量50000次; QPS/并发: 5qps	365天
人流量统计(动态版)	调用量: -; QPS/并发: -	调用量: 总量10000次; QPS/并发: 2qps	调用量: 总量50000次; QPS/并发: 5qps	365天
手部关键点识别	调用量: -; QPS/并发: -	调用量: 总量5000次; QPS/并发: 2qps	调用量: 总量10000次; QPS/并发: 2qps	365天

0元领取 总价: ¥0.00

**Step4 购买QPS**：如果您还需更多的QPS，请选择相应的API，点击[立即购买](#)，即可灵活按天或按月购买QPS。

QPS限制	QPS购买
不保证并发	免费试用
5	<a href="#">立即购买</a> <a href="#">配额详情</a>
5	<a href="#">立即购买</a> <a href="#">配额详情</a>
5	<a href="#">立即购买</a> <a href="#">配额详情</a>
5	<a href="#">立即购买</a> <a href="#">配额详情</a>
5	<a href="#">立即购买</a> <a href="#">配额详情</a>
不保证并发	免费试用

### QPS购买

购买方式:

选择接口:

QPS 数量	0<n≤10	10<n≤50	50<n≤100	100<n
单价(元/天/QPS)	50	40	30	20

购买期限: 2019.03.25 - 2019.04.01

购买数量:  1  QPS

购买费用: ¥400

注: 单个接口一天的QPS资源购买上限为50, 若需要更多QPS, 请[提交工单](#)处理。

**Step5 查看QPS配额详情：** 购买完成后，您还可点击人体分析概览页中的“配额详情”了解QPS购买配置的情况。



## API文档

### 简介

Hi，您好，欢迎使用百度人体分析在线API服务。

本文档主要针对API开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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**AI社区交流**：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天调用量超出限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)申请提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格文档](#)

🔗 接口能力

接口名称	能力简述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含四肢、脖颈、鼻子等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测和属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、我爱你、rock、掌心向上、竖中指、双手比心（3种）、数字（9种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图
人流量统计-动态版	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘等行为，可用于分析预警危险驾驶行为
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、物体交互或AR特效等场景
指尖检测（邀测）	精准定位食指指尖、及4个辅助关键点的坐标位置，模型针对儿童学习机点读场景进行专项调优，可用于点读搜题、自动跟读等应用

## 🔗 请求格式

### POST方式调用

注意：Content-Type为application/x-www-form-urlencoded，然后通过urlencode格式化请求体。

## 🔗 返回格式

### JSON格式

## 🔗 请求限制

请求图片需经过base64编码：图片的base64编码指将一副图片数据编码成一串字符串，使用该字符串代替图像地址。您可以首先得到图片的二进制，然后用Base64格式编码即可。

注意：图片的base64编码是不包含图片头的，如（data:image/jpg;base64,）

请求格式支持：PNG、JPG、BMP

接口名称	图片编码后大小限额
人体关键点识别	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50px，最长边最大4096px，建议长宽比3：1以内
人体检测和属性识别	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50px，最长边最大4096px，建议长宽比3：1以内
人流量统计	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50px，最长边最大4096px，建议长宽比3：1以内
手势识别	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50px，最长边最大4096px，建议长宽比3：1以内
人像分割	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50px，最长边最大4096px，建议长宽比3：1以内
驾驶行为分析	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50px，最长边最大4096px，建议长宽比3：1以内
人流量统计-动态版	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50px，最长边最大4096px，建议长宽比3：1以内
手部关键点识别	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50px，最长边最大4096px，建议长宽比3：1以内
指尖检测（邀约）	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50px，最长边最大4096px，建议长宽比3：1以内

## 调用方式

调用人体分析服务的API接口有两种调用方式，两种不同的调用方式采用相同的接口URL。

区别在于 **请求方式** 和 **鉴权方法** 不一样，请求参数和返回结果一致。

### 调用方式一

#### 请求URL数据格式

向API服务地址使用POST发送请求，必须在URL中带上参数：

access\_token：必须参数，参考“[Access Token获取](#)”。

注意：access\_token的有效期为30天，切记需要每30天进行定期更换，或者每次请求都拉取新token。

POST中参数按照API接口说明调用即可。

例如人体关键点识别API，使用HTTPS POST发送：

```
https://aip.baidubce.com/rest/2.0/image-classify/v1/body_analysis?
access_token=24.f9ba9c5241b67688bb4adbed8bc91dec.2592000.1485570332.282335-8574074
```

获取access\_token示例代码

bash
PHP
Java

Python

CPP

C#

Node

```
#### !/bin/bash
curl -i -k 'https://aip.baidubce.com/oauth/2.0/token?grant_type=client_credentials&client_id=【百度云应用的AK】&client_secret=【百度云应用的SK】'
```

说明：方式一鉴权使用的Access\_token必须通过API Key和Secret Key获取。

## 调用方式二

请求头域内容

在请求的HTTP头域中包含以下信息：

- host (必填)
- x-bce-date (必填)
- x-bce-request-id (选填)
- authorization (必填)
- content-type (必填)
- content-length (选填)

作为示例，以下是一个标准的人脸识别的请求头域内容：

```
POST /rest/2.0/face/v1/detect HTTP/1.1
accept-encoding: gzip, deflate
x-bce-date: 2015-03-24T13:02:00Z
connection: keep-alive
accept: */*
host: aip.baidubce.com
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
content-type: application/x-www-form-urlencoded
authorization: bce-auth-v1/46bd9968a6194b4bbdf0341f2286ccce/2015-03-24T13:02:00Z/1800/host;x-bce-date/994014d96b0eb26578e039fa053a4f9003425da4bfedf33f4790882fb4c54903
```



说明：方式二鉴权使用的API认证机制authorization必须通过百度云的AK/SK生成。

## 人体关键点识别

### 🔗 接口描述

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髋部、右髋部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：



多人场景：





### 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

### 请求说明

请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/image-classify/v1/body\\_analysis](https://aip.baidubce.com/rest/2.0/image-classify/v1/body_analysis)

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px

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。

bash
PHP
Java
Python
Cpp
C#
<p>人体关键点识别</p> <pre>curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/body_analysis?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码, 需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'</pre>

## 返回说明

接口除了返回人体框和每个关键点的坐标信息外，还会输出人体框和关键点的概率分数，实际应用中可以基于概率分数进行过滤，排除掉分数低的误识别“无效人体”，推荐的过滤方案：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.03时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.06甚至更高。

## 返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id, 用于问题定位
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数, 取值0-1, 得分越接近1表示识别准确的概率越大
+body_parts	是	object	身体部位信息, 包含21个关键点
++top_head	是	object	头顶

++top_lipau	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数

++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数

++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，**如果part被截断，则x、y坐标为part被截断的图片边界位置**，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体**。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

返回示例

```
{
  "person_num": 2,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        },
        "top_head": {
          "y": 242.53125,
          "x": 620,
          "score": 0.87757384777069
        },
        "right_mouth_corner": {
          "y": 308.625,
          "x": 606.78125,
          "score": 0.90121293067932
        },
        "neck": {
          "y": 335.0625,
          "x": 620,
          "score": 0.84662038087845
        },
        "left_shoulder": {
          "y": 361.5,
          "x": 699.3125,
          "score": 0.83550786972046
        },
        "left_knee": {
          "y": 731.625,
          "x": 699.3125,
          "score": 0.83575332164764
        },
        "left_ankle": {
          "y": 877.03125,
          "x": 725.75,
          "score": 0.85220056772232
        },
        "left_mouth_corner": {
          "y": 308.625,
          "x": 633.21875,
```

```
"score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 586.21875,
  "x": 593.5625,
  "score": 0.79803824424744
},
"left_wrist": {
  "y": 374.71875,
  "x": 884.375,
  "score": 0.89635348320007
},
"left_ear": {
  "y": 295.40625,
  "x": 659.65625,
  "score": 0.86607384681702
},
"left_elbow": {
  "y": 361.5,
  "x": 791.84375,
  "score": 0.83910942077637
},
"right_shoulder": {
  "y": 348.28125,
  "x": 553.90625,
  "score": 0.85635334253311
},
"right_ankle": {
  "y": 890.25,
  "x": 580.34375,
  "score": 0.85149073600769
},
"right_knee": {
  "y": 744.84375,
  "x": 580.34375,
  "score": 0.83749794960022
},
```

```
"right_wrist": {
  "y": 348.28125,
  "x": 368.84375,
  "score": 0.83893859386444
},
"location": {
  "height": 703.20654296875,
  "width": 652.61810302734,
  "top": 221.92272949219,
  "score": 0.99269664287567,
  "left": 294.03039550781
},
{
  "body_parts": {
    "left_hip": {
      "y": 576,
      "x": 1239.5625,
      "score": 0.84608125686646
    },
    "top_head": {
      "y": 261.15625,
      "x": 1176.59375,
      "score": 0.871442258358
    },
    "right_mouth_corner": {
      "y": 336.71875,
      "x": 1164,
      "score": 0.90951544046402
    },
    "neck": {
      "y": 361.90625,
      "x": 1176.59375,
      "score": 0.85904294252396
    },
    "left_shoulder": {
      "y": 361.90625,
      "x": 1239.5625,
      "score": 0.8512310385704
    },
    "left_knee": {
      "y": 714.53125,
      "x": 1277.34375,
      "score": 0.82312393188477
    },
    "left_ankle": {
      "y": 853.0625,
      "x": 1315.125,
      "score": 0.83786374330521
    },
    "left_mouth_corner": {
      "y": 336.71875,
      "x": 1189.1875,
      "score": 0.90610301494598
    },
    "right_elbow": {
      "y": 387.09375,
      "x": 1025.46875,
      "score": 0.88956367969513
    },
    "right_ear": {
      "y": 211.53125
```

```
    "y": 311.53125,  
    "x": 1138.8125,  
    "score": 0.86518502235413  
  },  
  "nose": {  
    "y": 324.125,  
    "x": 1176.59375,  
    "score": 0.9168484210968  
  },  
  "left_eye": {  
    "y": 311.53125,  
    "x": 1189.1875,  
    "score": 0.91715461015701  
  },  
  "right_eye": {  
    "y": 311.53125,  
    "x": 1164,  
    "score": 0.90343600511551  
  },  
  "right_hip": {  
    "y": 576,  
    "x": 1164,  
    "score": 0.81976848840714  
  },  
  "left_wrist": {  
    "y": 298.9375,  
    "x": 1378.09375,  
    "score": 0.86095398664474  
  },  
  "left_ear": {  
    "y": 311.53125,  
    "x": 1201.78125,  
    "score": 0.86899447441101  
  },  
  "left_elbow": {  
    "y": 324.125,  
    "x": 1315.125,  
    "score": 0.89198768138885  
  },  
  "right_shoulder": {  
    "y": 387.09375,  
    "x": 1101.03125,  
    "score": 0.85161662101746  
  },  
  "right_ankle": {  
    "y": 878.25,  
    "x": 1151.40625,  
    "score": 0.83667933940887  
  },  
  "right_knee": {  
    "y": 727.125,  
    "x": 1151.40625,  
    "score": 0.85485708713531  
  },  
  "right_wrist": {  
    "y": 387.09375,  
    "x": 949.90625,  
    "score": 0.83042001724243  
  }  
},  
"location": {  
  "height": 670.80139160156,  
  "width": 524.25476074219,
```



```
"top": 241.42504882812,
"score": 0.98725789785385,
"left": 902.15216064453
}
}
],
"log_id": "6362401025381690607"
}
```

## 人体检测和属性识别

### 🔗 接口描述

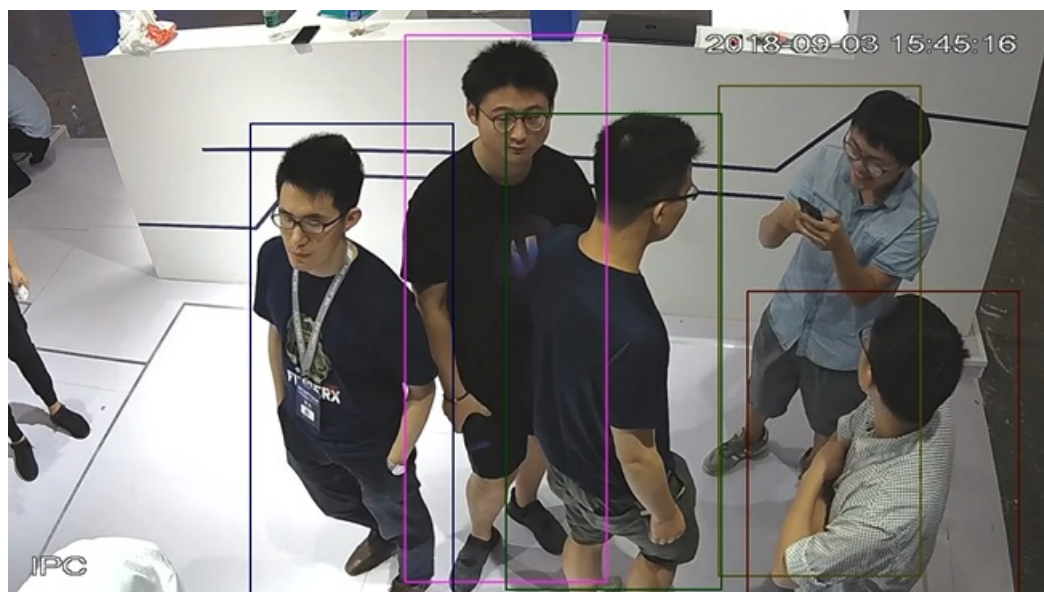
对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持17种属性，包括：性别、年龄阶段、上下身服饰（含类别/颜色）、是否戴帽子、是否戴口罩、是否背包、是否吸烟、是否使用手机、人体朝向等。

主要适用于监控场景的中低空斜拍视角，支持人体轻度截断、轻度遮挡、背面、侧面、动作变化等复杂场景。

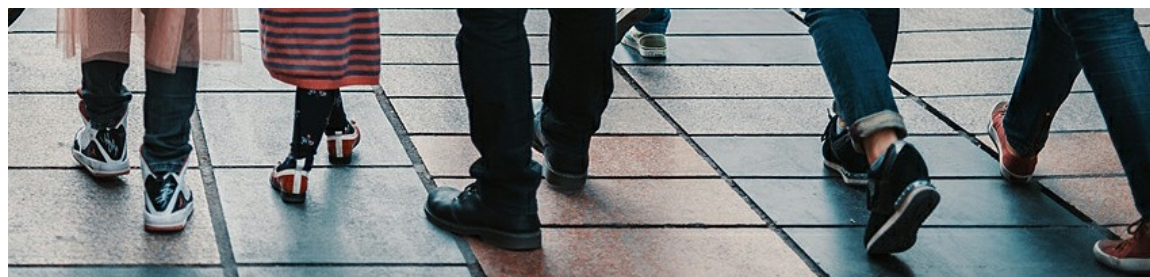
摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。夜间红外监控图片的识别效果可能欠佳。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：



非正常人体示例：严重截断



17种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	背包	bag	无背包、单肩包、双肩包
8	是否戴帽子	headwear	无帽、普通帽、安全帽
9	是否戴口罩	face_mask	无口罩、戴口罩、不确定
10	是否使用手机	cellphone	未使用手机、看手机、打电话、不确定
11	人体朝向	orientation	正面、背面、左侧面、右侧面
12	是否吸烟	smoke	吸烟、未吸烟、不确定
13	上方截断	upper_cut	无上方截断、有上方截断
14	下方截断	lower_cut	无下方截断、有下方截断
15	侧方截断	side_cut	无侧方截断、有侧方截断
16	遮挡情况	occlusion	无遮挡、轻度遮挡、重度遮挡
17	是否是正常人体	is_human	非正常人体、正常人体；用于判断说明人体的截断/遮挡情况，并非判断动物等非人类生物。 正常人体：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； 非正常人体：严重截断、或严重遮挡的人体，一般看不到肚脐眼，比如只有个脑袋、一条腿

注意：接口默认输出所有17个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

[在线调试](#)

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

## 请求说明

请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rest/2.0/image-classify/v1/body\_attr

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px
type	否	string	gender, age, lower_wear, upper_wear, headwear, face_mask, glasses, , upper_color, lower_color, cellphone, upper_wear_fg, , upper_wear_texture, lower_wear_texture, orientation, umbrella, bag, smoke, vehicle, carrying_item, upper_cut, lower_cut, occlusion, is_human	<p>1) 可选值说明： gender-性别， age-年龄阶段， lower_wear-下身服饰， upper_wear-上身服饰， headwear-是否戴帽子， face_mask-是否戴口罩， upper_color-上身服饰颜色， lower_color-下身服饰颜色， cellphone-是否使用手机， orientation-人体朝向， bag-背包， smoke-是否吸烟， upper_cut-上方截断， lower_cut-下方截断， side_cut-侧方截断， occlusion-遮挡情况， is_human-是否是正常人体</p> <p>2) type 参数值可以是可选值的组合，用逗号分隔；<b>若无此参数默认输出全部17个属性</b></p>

## 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。

bash
PHP
Java
Python
Cpp
C#
<p>人体检测和属性识别</p> <pre>curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/body_attr?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码, 需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'</pre>

## 返回说明

### 返回参数

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id, 用于问题定位
person_num	是	uint32	<b>检测到的人体数目</b>
person_info	是	object[]	每个人体框的信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的置信度分数, 取值0-1, 得分越接近1表示人体框准确的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性 女性

+++name	否	string	刀江、义江
+++score	否	float	对应概率分数
++age	否	object	年龄阶段
+++name	否	string	幼儿、青少年、青年、中年、老年
+++score	否	float	对应概率分数
++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、看手机、打电话、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定（背面或者头部被截断的人体，一般为不确定）
+++score	否	float	对应概率分数
++orientation	否	object	人体朝向
+++name	否	string	正面、背面、侧面
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包
+++score	否	float	对应概率分数
++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断、有上方截断
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断、有下方截断

+++score	否	float	对应概率分数
++side_cut	否	object	侧方截断
+++name	否	string	无侧方截断、有侧方截断
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡情况
+++name	否	string	无遮挡、轻度遮挡、重度遮挡
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体、非正常人体
+++score	否	float	对应概率分数

### 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "orientation": {
          "score": 0.99913859367371,
          "name": "正面"
        },
        "is_human": {
          "score": 0.99985098838806,
          "name": "正常人体"
        },
        "headwear": {
          "score": 0.7862206697464,
          "name": "无帽"
        },
        "face_mask": {
          "score": 0.99429154396057,
          "name": "无口罩"
        },
        "lower_wear": {
          "score": 0.96919763088226,
          "name": "长裤"
        },
        "vehicle": {
          "score": 0.99982362985611,
          "name": "无交通工具"
        },
        "upper_wear_fg": {
          "score": 0.40853327512741,
          "name": "T恤"
        },
        "lower_color": {
          "score": 0.77661311626434,
          "name": "灰"
        },
        "upper_cut": {
          "score": 0.99979013204575,
          "name": "无上方截断"
        },
        "lower_cut": {
          "score": 0.99873107671738,
```

```
{
  "name": "无下方截断"
},
"upper_wear": {
  "score": 0.98250997066498,
  "name": "短袖"
},
"cellphone": {
  "score": 0.99923753738403,
  "name": "未使用手机"
},
"gender": {
  "score": 0.91399890184402,
  "name": "女性"
},
"age": {
  "score": 0.94620317220688,
  "name": "青年"
},
"bag": {
  "score": 0.88887012004852,
  "name": "无背包"
},
"smoke": {
  "score": 0.99988353252411,
  "name": "未吸烟"
},
"upper_color": {
  "score": 0.94646281003952,
  "name": "灰"
},
"occlusion": {
  "score": 0.99341428279877,
  "name": "无遮挡"
}
},
"location": {
  "height": 826,
  "width": 219,
  "top": 176,
  "score": 0.9693506360054,
  "left": 754
}
},
"log_id": "1324248829299466129"
}
```

## 人流量统计

### 🔗 接口描述

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，不支持追踪和去重）。

适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：





注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

#### 🔗 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

#### 🔗 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/image-classify/v1/body\\_num](https://aip.baidubce.com/rest/2.0/image-classify/v1/body_num)

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px
area	否	string	-	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p><b>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</b></p> <p>area参数设置错误的示例：某个坐标大于或等于原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p><b>area参数设置格式：</b></p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域： xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;...<b>注意：area区域坐标需要从 (1, 1) 开始，(0, 0) 属于图片边界之外</b></p>
show	否	string	true,false	是否输出渲染的图片，默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false

#### 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。

bash
PHP
Java
Python
Cpp
C#

## 人流量统计

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/body_num?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码, 需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

## 返回说明

## 返回参数

字段	是否必选	类型	
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

## 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

## 手势识别

## 接口描述

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种常见手势，适用于手势特效、智能家居手势交互等场景。

支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。




图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

注：




1) 上述24类以外的其他手势会划分到other类。

2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1 (原食指)	One	
2	数字5 (原掌心向前)	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
			

11	我爱你	ILY	A hand gesture where the index, middle, and ring fingers are extended straight out, representing the letters 'I', 'L', and 'Y' respectively.
12	掌心向上	Palm_up	A hand gesture with the palm facing upwards, fingers slightly spread.
13	双手比心1	Heart_1	A hand gesture where both hands are used to form a heart shape by touching the thumbs and index fingers.
14	双手比心2	Heart_2	A hand gesture where both hands are used to form a heart shape by touching the thumbs and index fingers, similar to Heart_1 but with a different angle.
15	双手比心3	Heart_3	A hand gesture where both hands are used to form a heart shape by touching the thumbs and index fingers, similar to Heart_1 and Heart_2 but with a different angle.
16	数字2	Two	A hand gesture showing two fingers (index and middle) extended straight out.
17	数字3	Three	A hand gesture showing three fingers (index, middle, and ring) extended straight out.
18	数字4	Four	A hand gesture showing four fingers (index, middle, ring, and pinky) extended straight out.
19	数字6	Six	A hand gesture showing six fingers (index, middle, ring, pinky, and thumb) extended straight out.
20	数字7	Seven	A hand gesture showing seven fingers (index, middle, ring, pinky, thumb, and little) extended straight out.
21	数字8	Eight	A hand gesture showing eight fingers (index, middle, ring, pinky, thumb, little, and ring) extended straight out.
			A hand gesture showing nine fingers (index, middle, ring, pinky, thumb, little, ring, and middle) extended straight out.

22	数字9	Nine	
23	Rock	Rock	
24	竖中指	Insult	

### 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rest/2.0/image-classify/v1/gesture>

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px

#### 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。

bash
PHP
Java
Python

Cpp

C#

### 手势识别

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/gesture?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码, 需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

## 返回说明

### 返回参数

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标, 手势、人脸
+classname	否	string	目标所属类别, 24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id, 用于问题定位

### 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [{
    "probability": 0.9844077229499817,
    "top": 20,
    "height": 156,
    "classname": "Face",
    "width": 116,
    "left": 173
  },
  {
    "probability": 0.4679304957389832,
    "top": 157,
    "height": 106,
    "classname": "Heart_2",
    "width": 177,
    "left": 183
  }
]}
```

## 人像分割

### 🔗 接口描述

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

#### 1) 原图



#### 2) 二值图





3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果，示例代码如下；灰度图和前景人像图不用处理，直接解码保存图片即可。

Python :

```
{
  import cv2
  import numpy as np
  import base64
  labelmap = base64.b64decode(res['labelmap']) # res为通过接口获取的返回json
  nparr = np.fromstring(labelmap, np.uint8)
  labelimg = cv2.imdecode(nparr, 1)
  # width, height为图片原始宽、高
  labelimg = cv2.resize(labelimg, (width, height), interpolation=cv2.INTER_NEAREST)
  im_new = np.where(labelimg==1, 255, labelimg)
  cv2.imwrite('path/to/your/outputfile', im_new)
}
```

Java :

```
public static BufferedImage resize(BufferedImage img, int newW, int newH) {
    Image tmp = img.getScaledInstance(newW, newH, Image.SCALE_SMOOTH);
    BufferedImage dimg = new BufferedImage(newW, newH, BufferedImage.TYPE_INT_ARGB);

    Graphics2D g2d = dimg.createGraphics();
    g2d.drawImage(tmp, 0, 0, null);
    g2d.dispose();

    return dimg;
}

public static void convert(String labelmapBase64, int realWidth, int realHeight) {
    try {

        byte[] bytes = Base64.getDecoder().decode(labelmapBase64);
        InputStream is = new ByteArrayInputStream(bytes);
        BufferedImage image = ImageIO.read(is);
        BufferedImage newImage = resize(image, realWidth, realHeight);
        BufferedImage grayImage = new BufferedImage(realWidth, realHeight, BufferedImage.TYPE_BYTE_GRAY);
        for(int i= 0 ; i < realWidth ; i++){
            for(int j = 0 ; j < realHeight; j++){
                int rgb = newImage.getRGB(i, j);
                grayImage.setRGB(i, j, rgb * 255); //将像素存入缓冲区
            }
        }
        File newFile = new File("gray.jpg");
        ImageIO.write(grayImage, "jpg", newFile);
    } catch (IOException e) {
        e.printStackTrace();
    }
}
```

## 🔗 在线调试

您可以在[示例代码中心](#)中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

## 🔗 请求说明

### 请求示例

HTTP 方法 : POST

请求URL : [https://aip.baidubce.com/rest/2.0/image-classify/v1/body\\_seg](https://aip.baidubce.com/rest/2.0/image-classify/v1/body_seg)

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px
type	否	string	labelmap, scoremap, foreground	<p>可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费</p> <p>1) 可选值说明：</p> <p>labelmap - 二值图像，需二次处理方能查看分割效果</p> <p>scoremap - 人像前景灰度图</p> <p>foreground - 人像前景抠图，透明背景</p> <p>2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部3类结果图</p>

#### 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。

bash
PHP
Java
Python
Cpp
C#

## 人像分割

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/body_seg?access_token=【调用鉴权接口获取的token】'
--data 'image=【图片Base64编码，需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

## 返回说明

## 返回参数

字段	是否必选	类型	说明
labelmap	否	string	分割结果图片，base64编码之后的二值图像，需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap，归一到0-255，不用进行二次处理，直接解码保存图片即可。 Base64编码后的灰度图文件，图片中每个像素点的灰度值 = 置信度 * 255，置信度为原图对应像素点位于人体轮廓内的置信度，取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图，透明背景，Base64编码后的png格式图片，不用进行二次处理，直接解码保存图片即可。将置信度大于0.5的像素抠出来，并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度，注意当值为0时 数据类型为int
+left	否	float	人体区域离左边界的距离，注意当值为0时 数据类型为int
+top	否	float	人体区域离上边界的距离，注意当值为0时 数据类型为int
+width	否	float	人体区域的宽度，注意当值为0时 数据类型为int
+score	否	float	人体框的概率分数，取值0-1，，注意当值为0时 数据类型为int

## 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg"
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ],
}
```

## 驾驶行为分析

### 🔗 接口描述

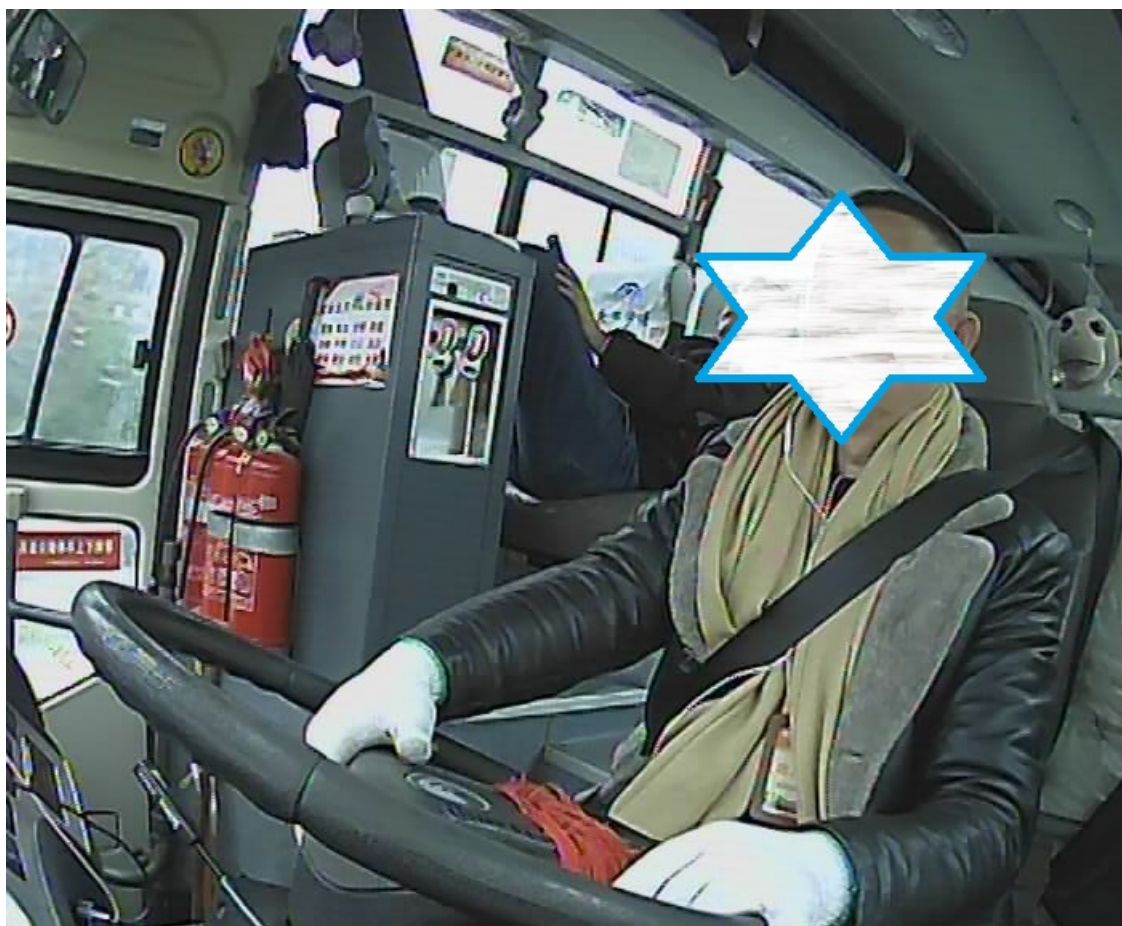
对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 1、服务只适用于车载监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别](#)服务。
- 2、车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 3、车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 4、服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 5、图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：



#### 🔗 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

#### 🔗 请求说明

##### 请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/image-classify/v1/driver\\_behavior](https://aip.baidubce.com/rest/2.0/image-classify/v1/driver_behavior)

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px
type	否	string	smoke,cellphone,not_buckling_up,both_hands_leaving_wheel,not_facing_front,no_face_mask,yawning,eyes_closed,head_lowered	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	0,1	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

#### 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。

bash
PHP
Java
Python
Cpp
C#

## 驾驶行为分析

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/driver_behavior?access_token=【调用鉴权接口获取的token】' --data 'image=【图片Base64编码，需urlencode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

## 返回说明

## 返回参数

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可



+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

#### 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      }
    },
    {
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

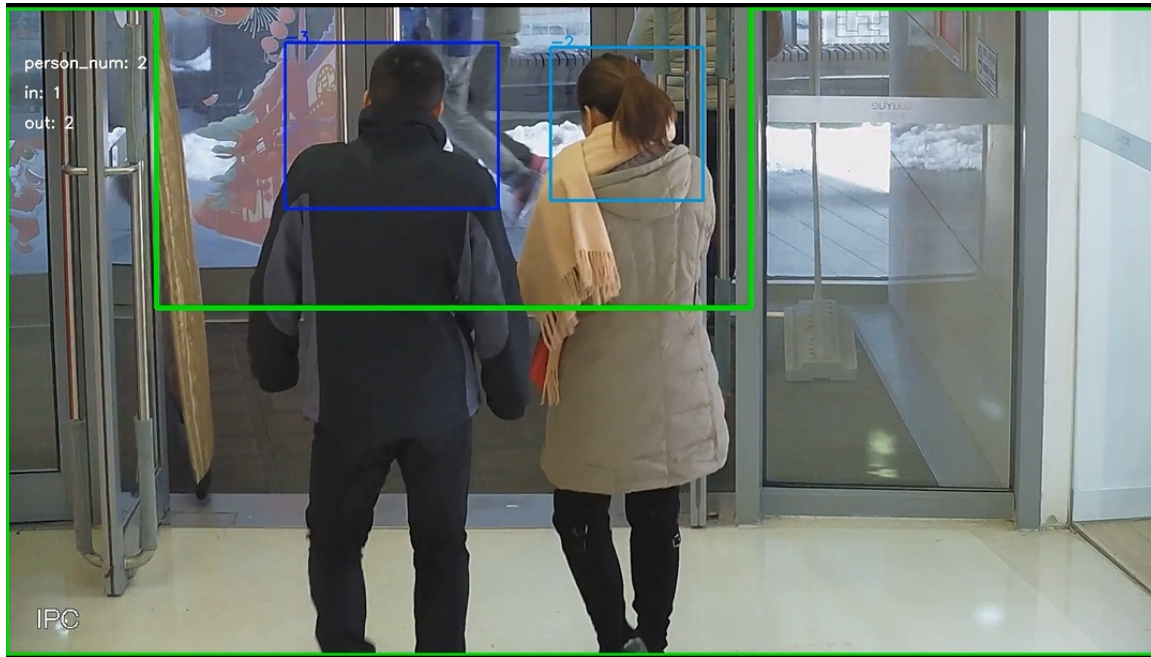
### 🔗 接口描述

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以人体头肩为主要识别目标，核心功能：

- 静态人数统计：传入单帧图像，检测图片中的人体头肩，返回图中总人数。

- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



🔗 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

🔗 请求说明

请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/image-classify/v1/body\\_tracking](https://aip.baidubce.com/rest/2.0/image-classify/v1/body_tracking)

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
dynamic	是	string	true,false	true：动态人流量统计，返回总人数、跟踪ID、区域进出人数；false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	int32	-	任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	string	true,false	每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
image	是	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。图片的base64编码是不包含图片头的，如(data:image/jpeg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px。
show	否	string	true,false	是否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	string	小于原图像素范围	静态人数统计时，只统计区域内的人，缺省时为全图统计。动态人流量统计时，进出区域的人流会被统计。逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。 <b>坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。</b>

#### 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。

bash
PHP
Java
Python
Cpp
C#

### 人流量统计 (动态版)

```
curl -i -k 'https://aip.baidubce.com/rest/2.0/image-classify/v1/body_tracking?access_token=【调用鉴权接口获取的token】' --data 'area=1,1,719,1,719,719,1,719&case_id=1&case_init=false&dynamic=true&image=【图片Base64编码,需UrlEncode】' -H 'Content-Type:application/x-www-form-urlencoded'
```

#### area参数设置说明

进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

##### 示例1：

如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。

如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。

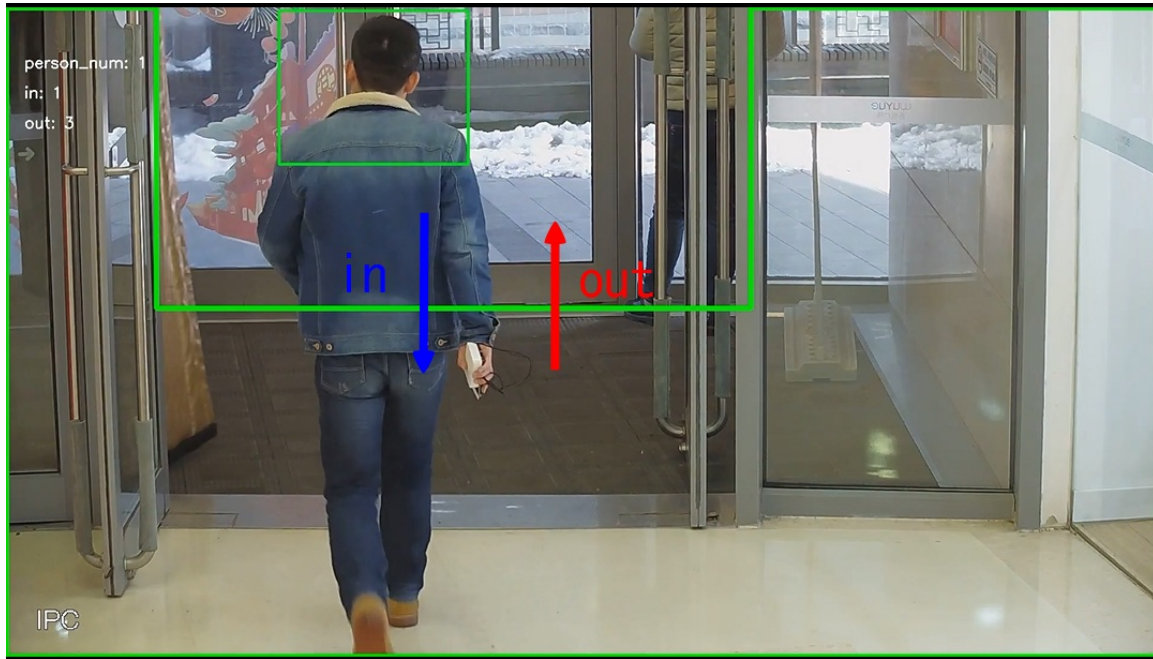


同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

##### 示例2：

如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框

里，红色箭头的方向代表out，人走出区域框，走向门外。



[返回说明](#)

返回参数

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：

```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：



```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

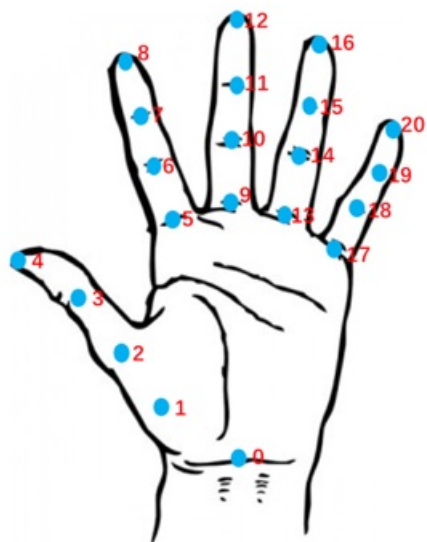
## 手部关键点识别

### 🔗 接口描述

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

21个关键点对应位置示意图：



### 🔗 在线调试



您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

## 请求说明

### 请求示例

HTTP 方法：POST

请求URL：[https://aip.baidubce.com/rest/2.0/image-classify/v1/hand\\_analysis](https://aip.baidubce.com/rest/2.0/image-classify/v1/hand_analysis)

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

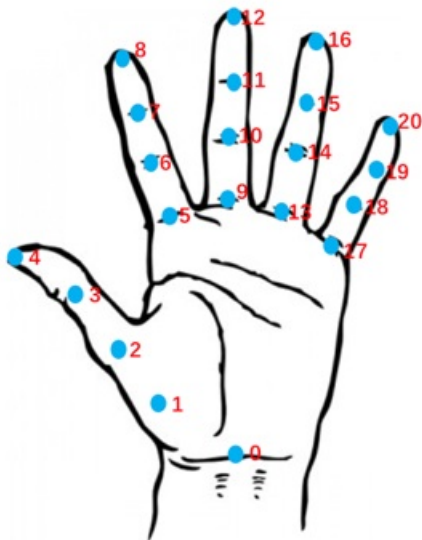
参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px。

## 返回说明

### 返回参数

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
++score	是	float	置信度分数，当前取值范围0-21，应用时可除以21进行归一化处理，下一版服务会进行更新、修正
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数，取值范围0-1。可基于置信度分数进行过滤，排除掉分数低的误识别“无效关键点”，推荐的过滤方案：置信度分数 $\geq 0.2$ 。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

返回关键点对应位置示意图：



## 返回示例

```
{
  "log_id": "4653273132348525916",
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        "2": {
          "y": 533,
          "x": 916,
          "score": 0.75532239675522
        },
        "3": {
          "y": 467,
          "x": 938,
          "score": 0.79282909631729
        },
        "4": {
          "y": 380,
          "x": 993,
          "score": 0.88114696741104
        },
        "5": {
          "y": 424,
          "x": 829,
          "score": 0.90723776817322
        },
        "6": {
          "y": 358,
          "x": 862,
          "score": 0.62833058834076
        },
        "7": {
          "y": 391,
          "x": 916,
          "score": 0.77525424957275
        },
        "8": {
          "y": 456,
          "x": 927,
          "score": 0.87964057922363
        },
        "9": {
          "y": 402,
          "x": 775,
          "score": 0.91389346122742
        },
        "10": {
          "y": 325,
          "x": 786,
          "score": 0.85947573184967
        }
      }
    }
  ]
}
```

```
},
"11": {
  "y": 249,
  "x": 818,
  "score": 0.88918441534042
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
  "y": 325,
  "x": 567,
  "score": 0.91110396385193
}
}
"location": {
  "height": 556,
  "width": 426,
  "top": 151,
  "score": 17.495880126953,
  "left": 567
}
}
]
```

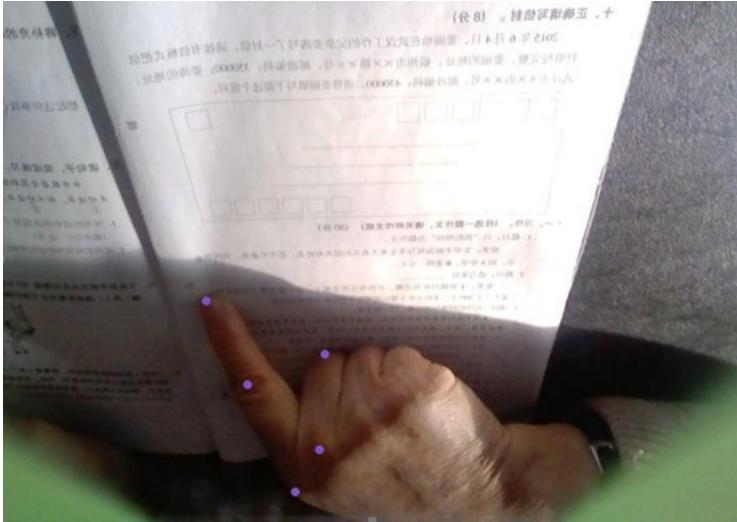
## 指尖检测（邀测）

### 接口描述

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，定位食指指尖、及4个辅助关键点的坐标位置，模型针对儿童学习机点读场景进行专项调优，可用于点读搜题、自动跟读等应用。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

示例图如下：



注：

- 1、邀测的接口，不能直接在控制台调用，可通过百度人体分析官方QQ群（860337848）联系群管、或提交工单申请开通测试权限。
- 2、邀测的接口，暂未封装服务端SDK，只能通过API调用。

### 在线调试

您可以在 [示例代码中心](#) 中调试该接口，可进行签名验证、查看在线调用的请求内容和返回结果、示例代码的自动生成。

### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：<https://aip.baidubce.com/rest/2.0/image-classify/v1/fingertip>

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	是	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。图片的base64编码是不包含图片头的，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px。

[返回说明](#)

### 返回参数

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
++score	是	float	置信度分数，取值范围0-1
+hand_parts	是	object	指尖关键点信息，包含食指指尖、及辅助的4个关键点
++0	是	object	<b>食指指尖</b>
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数，取值范围0-1。可基于置信度分数进行过滤，排除掉分数低的误识别“无效关键点”， <b>推荐的过滤方案：置信度分数<math>\geq 0.2</math></b> 。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用
++1	是	object	食指中间关节
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
++2	是	object	食指指根
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
++3	是	object	中指中间关节
+++x	是	uint32	x坐标

+++y	是	uint32	y坐标
+++score	是	float	置信度分数
++4	是	object	中指指根
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

### 返回示例

```
{
  "log_id": "4653273132348525916",
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        "2": {
          "y": 533,
          "x": 916,
          "score": 0.75532239675522
        },
        "3": {
          "y": 467,
          "x": 938,
          "score": 0.79282909631729
        },
        "4": {
          "y": 380,
          "x": 993,
          "score": 0.88114696741104
        }
      }
    },
    {
      "location": {
        "height": 556,
        "width": 426,
        "top": 151,
        "score": 0.795880126953,
        "left": 567
      }
    }
  ]
}
```

### 错误码

[错误返回格式](#)

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。

#### 🔗 错误码信息

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282004	invalid parameter(s)	请求参数错误，请参考API文档，修改参数后重试
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## 错误码

### 错误返回格式

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。

### 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

## 服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
6	No permission to access data	无权访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队

216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## 私有化部署

### 整体介绍

Hi，您好，欢迎使用百度人体分析私有化部署产品。

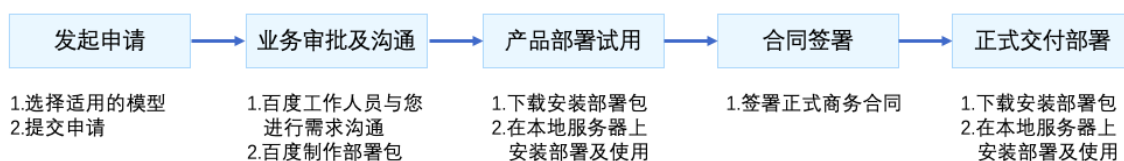
人体分析私有化部署方案是基于人体分析模型封装而成的能够本地化部署的纯软件方案，将软件包部署在本地服务器或专有云服务器（需GPU显卡）上后，启动服务，即可调用与在线API功能基本完全相同的接口（参数有少量区别，具体说明请参考[接口调用文档](#)）。

#### 方案概述

私有化部署方案适用于网络环境受限制（例如专网）或者对数据私密性有要求的各类业务场景，如公共安防、园区管理、校园监控等，提供分布式高并发部署方案，纯软件方案可以快速测试、快速交付，且不与硬件进行捆绑，更高效、灵活。

注：百度为开发者提供的是包含人体分析模型的部署包软件，硬件采购和服务器部署请提前准备，服务器配置推荐请参考下文[硬件配置](#)部分。

私有化部署的整体业务流程如下，下文将针对各个环节分别说明。



#### 产品功能

人体分析的各个模型拆分为不同的私有部署包，目前已上线人体关键点识别、人流量统计、人体检测、人体属性识别（单人版）、手势识别、人像分割、手部关键点识别7个服务的部署包，可选取所需能力灵活应用。

##### • 人体关键点识别

检测图像中的所有人体，标记出每个人体的坐标位置；定位人体的21个核心关键点，包含头顶、五官、四肢等主要关节部位，输出每个关键点的坐标信息。

适应轻度遮挡、轻度截断、背面、侧面、中低空斜拍、大动作等复杂场景，实际应用中可根据接口输出的人体框概率分数、关键点概率分数进行过滤，排除掉分数低的误识别“无效人体”。

### • 人体检测

检测图像中的所有人体，返回每个人体的矩形框位置；支持人体重叠、遮挡、截断、背面、侧面、动作变化等复杂场景。

适应轻度遮挡、轻度截断、背面、侧面、中低空斜拍等复杂场景，**实际应用中可根据接口输出的人体框概率分数进行过滤，排除掉分数低的误识别“无效人体”。**

### • 人体属性识别（单人版）

输入单个人体的图片，识别人体的静态属性和行为，共支持28种属性，包括：性别年龄、服饰类别、服饰颜色、戴帽子（可区分安全帽/普通帽）、戴口罩、背包、抽烟、使用手机等。

**主要适用于中低空斜拍视角**，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等场景。

### • 人流量统计

识别和统计图像当中的人体个数（静态统计，不支持追踪和去重）。支持框选多个不规则区域统计局部人数，方便统计观察图片中多个重点区域的人群热度，同时可输出渲染图片。

**适用于3米以上的中远距离俯拍**，以头部为主要识别目标统计人数，无需正脸、全身照；无人数上限，适用于机场、车站、商场、展会、景区等各类人群密集场所。**支持轻度畸变的鱼眼摄像头。**

### • 手势识别

识别图片中的手部位置和手势类型，可识别24种常见手势，包括拳头、OK、比心、我爱你、点赞、Diss、Rock、竖中指、数字等。

支持自拍、他人拍摄、各种拍摄角度，**拍摄距离3米以内，1米内为佳。**

### • 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）。

### • 手部关键点识别

精准定位手部的21个主要骨节点，包括指尖、各节指骨连接处等，返回每个骨节点的坐标信息；可用于自定义手势检测、AR特效、人机交互等场景。支持多个手部、他人拍摄视角。

### • 驾驶行为分析

针对车载场景，识别驾驶员使用手机、抽烟、不系安全带、未佩戴口罩、闭眼、打哈欠、双手离开方向盘等动作姿态，分析预警危险驾驶行为，提升行车安全性。

## 🔗 购买指南

人体分析私有化部署方案为付费产品，可自用或集成为客户使用，需具备基本的IT能力进行部署使用。

- **免费测试**：人体分析的各个部署包，均提供免费测试，**最高可获得3个月免费测试期。**
- **正式购买**：按照授权数量计费（具体授权方式见下文[授权说明](#)），不同功能的部署包分别计费，支持按年授权、永久授权两种购买方式，具体价格请联系客户经理/工作人员沟通。

## 🔗 硬件配置

人体分析的模型需在GPU服务器上运行，请准备好GPU再提交部署包申请。

### 1、GPU显卡选型建议

序号	显卡系列	推荐型号	说明
1	NVIDIA Tesla	P4、P40	P40显存更大，但运行模型的计算单元并不多，因此整体性能与P4差别不大
2	NVIDIA GTX系列	1060、1070、1070Ti、1080、1080Ti	显存需要≥6G
3	NVIDIA RTX系列	2070、2070Ti、2080、2080Ti	
4	NVIDIA T系列	T4	

注：

- ① GTX 1060以下版本的显卡，由于性能较低、版本较老，不建议使用。
- ② 由于GTX的计算单元比P4更多，因此单卡支持的QPS数量也会更多，但因为GTX并非服务器专用显卡，请大家基于自身业务需求进行选择。

## 2、基础硬件要求

环境类别	环境需求	备注
硬件要求	CPU：≥1核 内存：>=16 G，推荐32 G 硬盘：>100 GB，推荐500 GB以上	INTEL i7/i9等市场主流CPU均可，推荐使用志强系列 如：INTEL Xeon E5-2650 V4 12C 2.2GHZ*2
操作系统（64位）	CentOS 7及以上、Ubuntu 14/16/18、RedHat 7.2	
网络环境	有线局域网，不支持无线网络	机器需在局域网内，且能获取到ip地址
服务端口	8443 端口不能被占用（鉴权服务默认端口）	

## 性能指标

人体分析各个部署包的性能说明如下，硬件配置选型、服务调用时可参考。

序号	部署包模型	显存占用	单显卡QPS承载量（以P4卡为例）	说明
1	人体关键点识别	5.2G	25~35 QPS	与图片中的人数有关，图片中人越多，单卡支持的QPS数量越少
2	人体检测	2G	45~50 QPS	与图片中的人数有关，图片中人越多，单卡支持的QPS数量越少
3	人体属性识别（单人版）	2.3G	80 QPS	仅针对单人图片进行识别，需搭配人体检测模型使用
4	人流量统计	5G	10~15 QPS	与区域个数、是否输出渲染图有关
5	手势识别	< 1G	70 QPS	支持多个手部，但对单个手部的识别效果更好
6	人像分割	4G	~20 QPS	与图片中的人体数量有关
7	手部关键点识别	3G	45 QPS	支持多个手部，但对单个手部的识别效果更好
8	驾驶行为分析	1.6G	40 QPS	支持夜间红外场景

注：以上性能指标仅供参考，具体应用中以项目实测为准

**抽帧策略建议：**人体分析的应用场景与人脸有一些差异，因不需要对人员进行身份判断，一般情况下无需频繁抽帧识别，实际

应用中，可根据场景特性、业务要求，灵活设置，权衡应用效果和资源利用率。

- 人体关键点识别、人体检测、人体属性识别：需要实时监控的核心区域，帧率可设置高一些，如1fps~2fps，避免漏检；夜晚或很少有人出现的非核心区域，帧率可相应低一些，如2秒一帧、甚至5秒一帧，减少传输、并发等资源消耗。
- 人流量统计：主要用于人群密度分析监控，理论上监控画面中人数变化不会太快，实际应用中可根据业务需求调整帧率，如5秒一帧。
- 手势识别、手部关键点识别：用户的手势变化可能较快，帧率可设置稍高一些，如2fps或者更高。

## 如何接入

### 1、申请私有化部署包

登录**百度云控制台**，选择**人体分析**技术类别，进入私有服务部署管理页面，发起申请。



产品服务 / 人体分析 - 私有部署包管理

私有部署包管理

可以部署到本地GPU服务器的「纯软件」人体识别模型包，支持主流Linux系统，适配最新NVIDIA GPU显卡；提供分布式高并发部署方案，适用于安防监控、园区管理等场景。 [查看详情介绍](#)

[申请流程](#) [展开教程](#)

人体私有化部署包授权列表

测试版 正式版

[发起申请](#)

序号	申请编号	申请模型	显卡类型	申请日期	到期日期	状态	操作
1	1590	人像分割(1)	P4/GTX系列	2020-03-13	2020-05-12	● 申请成功	<a href="#">查看详情</a> <a href="#">下载部署包</a>
2	1576	人流量统计(1)	P4/GTX系列	2020-03-11	2020-05-10	● 审核中	--
3	1460	人体关键点识别(1)	T4/RTX系列	2020-01-20	2020-05-10	● 审核失败	<a href="#">查看详情</a>



产品服务 / 人体分析 - 私有部署包管理 / 发起申请

申请测试部署包

选择模型配置

选择模型

\* 模型

人体检测 [?]

人体关键点识别 [?]

人体属性识别 [?]

人流量统计 [?]

手势识别 [?]

手部关键点识别 [?]

人像分割 [?]

\* 显卡类型

P4/GTX系列  T4/RTX系列

注：如果使用其他型号显卡或使用CPU，请[提交工单](#)咨询

申请信息

申请类型 测试部署包







证书有效期 2020.04.01 - 2020.06.01

[保存并继续](#)

提交部署包申请后，有如下关键动作需要您的配合：





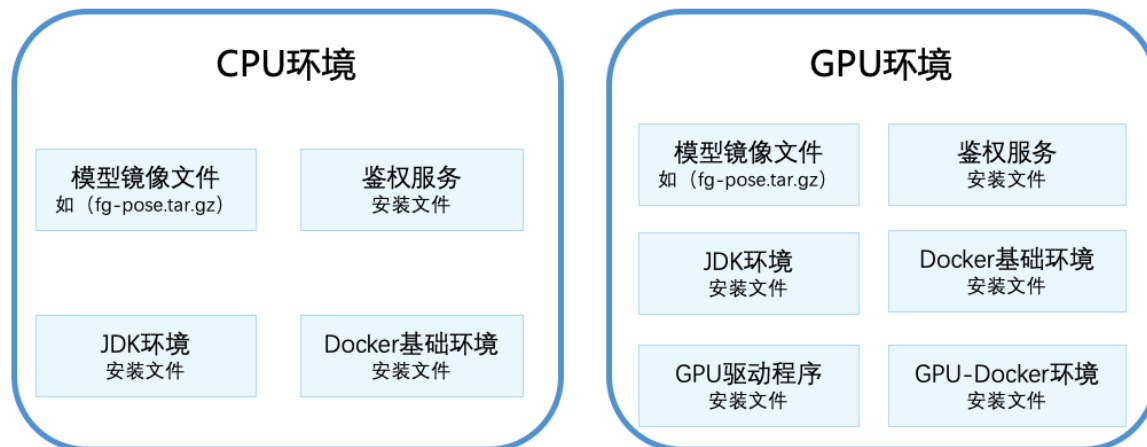
名称	修改日期	大小	种
 download.bat	2018年12月7日 下午2:34	67 字节	Ba
 download.sh	2018年12月7日 下午2:34	4 KB	Te
 face-1c	今天 下午2:36	--	文
 package	今天 下午2:36	--	文
 refs.txt	2018年12月7日 下午2:34	5 KB	纯
 shell.w32-ix86	今天 下午2:37	--	文

最后点击download.bat执行下载脚本，下载安装包。

最终得到的部署包，包含以下内容：鉴权服务安装包、应用镜像文件安装包以及docker安装包等基础依赖环境。

备注：进入original目录后可打开refs.txt文件查看详细的下载内容。

CPU环境部署和GPU环境部署下载后包含的的安装文件有所区别。具体示例如下：



## 授权说明

私有化方案需要对服务器硬件进行授权，您可以在GPU服务器或者普通PC物理机上运行百度提供的硬件[指纹提取工具](#)，提取硬件信息(提取成功后会生成一个指纹文件)，然后在控制台人体分析私有化部署包申请界面上上传指纹文件，审批通过后即可获得授权。

将运行人体分析模型的GPU与运行授权服务的硬件间建立通信，就能运行起人体分析模型了，可运行模型的显卡数量与授权数量相同。

注：授权服务可以运行在CPU设备上（比如一台固定的PC机），针对这台设备生成license，license文件中包含几个授权，就可以人体分析模型在多少个显卡上运行，只要保证被采集了指纹的硬件不被替换，授权也不受影响（即使替换GPU服务器也不会受影响）。

## 授权有效期

人体分析的私有部署包提供2种授权方式：按年授权、永久授权，不同的授权方式价格不同，具体费用信息请联系客户经理/工作人员沟通。

- 按年授权：自购买日期生效，到约定的日期后license失效，部署包不可继续使用，如要继续使用，需重新购买新的部署包。
- 永久授权：自购买日期生效，license永久有效，授权的硬件不替换、损坏，即可一直使用。

注：百度一直保持授权服务的兼容，只要您之前已经购买过某个模型的授权，那么百度在这个模型上的性能提升后的新版部署包，都可以自助升级。但如果新增功能，形成了一个新的接口，则需要重新购买。



## 🔗 服务部署

部署环境准备、部署方式、部署流程（含鉴权服务和应用服务）详见[部署说明文档](#)。

## 🔗 接口调用

应用服务部署成功后，即可获得与在线API基本完全相同的接口，可参考[接口调用](#)文档开始调用测试。

# 部署说明

Hi，您好，欢迎使用百度人体分析私有化部署产品。

人体分析私有化部署方案是基于人体分析模型封装而成的能够本地化部署的纯软件方案，将软件包成功部署在本地服务器或专有云服务器（需GPU显卡）上后，启动服务，即可调用与在线API功能基本完全相同的接口（参数有少量区别，具体说明请参考[接口调用](#)文档）。

本文档主要介绍人体分析私有化软件包的部署流程。

## 🔗 前提条件

- GPU服务器：人体分析模型需在GPU服务器上运行，请准备好GPU资源再提交部署包申请。
- 获取部署包：部署包的申请、下载，请参考[如何接入](#)。下载好部署包之后，请参考本文档进行安装、部署。

## 🔗 准备服务器

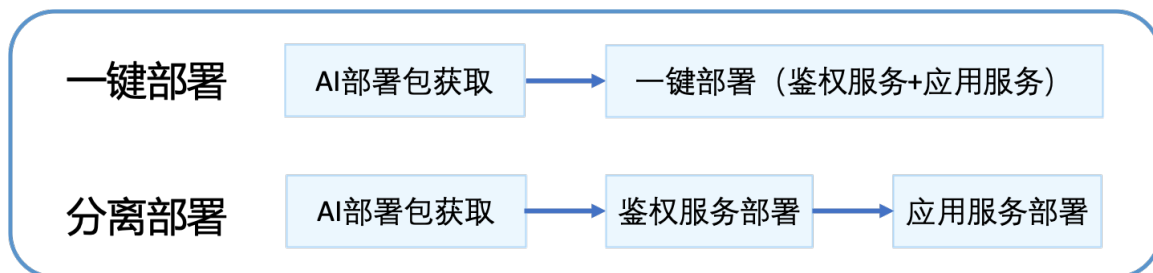
GPU显卡选型建议、基础硬件要求，请参考[硬件配置](#)。

安装系统：目前只支持CentOS 7、Ubuntu 14/16/18、RedHat 7.2，系统的安装流程请按照官方介绍进行安装。

## 🔗 部署指南

部署方案支持以下两种模式，可根据不同业务场景选择适合自身环境的部署方式。

- 单机一键部署：将鉴权服务、应用服务部署在一台物理机上的场景。即执行一条命令将鉴权服务、应用服务安装完成。
- 多机分离部署：将鉴权服务、应用服务分别部署的场景。首先部署鉴权服务、然后部署应用服务。



**鉴权服务部署**：鉴权服务包含百度发布的服务授权证书，如果不安装鉴权服务，后续的应用服务也将无法启动

**应用服务部署**：包含Docker等基础环境以及相关技术方向的算法模型，是私有化产品的核心。部署应用服务的前提是部署鉴权服务，应用服务在运行时会实时请求鉴权服务，需要保障两个服务之间能够顺利通信。

名词解析：

- 基线目录：为了方便介绍目录结果，约定私有化部署解压目录为基线目录，路径为/home/private，读者可以自行替换。
- License：由百度发布的服务授权证书，只有经过授权的服务器才能成功部署服务，待部署服务器的信息请使用百度提供的机器指纹采集工具采集指纹后提交到申请后台。
- 工作目录：/home/baidu为百度服务使用的工作目录。

## 🔗 单机一键部署

1、获取部署包安装文件下载链接，下载部署包；

下载完成的文件示例如下：609859F08F4B4FB782948D669EE3CFE3.tar.gz；

2、执行以下命令解压部署包；

```
tar zxvf 609859F08F4B4FB782948D669EE3CFE3.tar.gz
```

3、解压后进入original目录执行**bash download.sh**命令获取全部安装文件，执行脚本后会自动下载以下安装文件：鉴权服务安装包、应用服务安装包以及docker安装包等基础依赖环境；

```
bash download.sh
```

4、将带有全部安装文件的original文件夹上传到待部署的服务器中；

5、进入以下文件路径；

```
cd original/package/Install
```

6、执行一键部署命令；

```
python install.py inall
```

7、安装过程中会自动进行环境检查，如果安装进程停住并提示"Environment checking failed! Please fix them before installation."表明环境检查失败,请先排查失败的环境检查项，再重新执行安装，或者输入continue强制继续安装；

8、安装过程中会提示输入**auth server cluster's ip**、**local server's ip**，两个ip地址均输入本机实际网络地址即可；

9、提示安装成功后，可检查服务是否启动；

```
docker ps ;
```

10、如果服务没有启动，可以使用**docker ps -a**，查看容器id；

11、鉴权服务默认的http端口为8443，通过**netstat -apn | grep 8443**可以看端口是否已经被监听；

应用服务默认的http端口请参考[接口调用文档](#)。

## 🔗 多机分离部署

### 🔗 1. 鉴权服务部署

#### 硬件要求（物理机）

- CPU架构：AMD 64/x86\_64
- 内存：>=32G（推荐，不强制）
- 硬盘：>=512G（推荐）
- 网络环境：机器需要在局域网内，且能获取到ip地址

## 操作系统要求

- 基于安全的角度，暂时不支持虚拟机部署，包括但不限于Virtual Box、VMware等
- 支持的Linux发行版列表
  - Ubuntu: 14、16、18
  - CentOS : 7.0及以上
  - RedHat : 7.2
  - SUSE 12
- Linux内核要求  $\geq 3.10$
- GLIBC  $\geq 2.17$
- GLIBCXX  $\geq 3.4.19$
- Python 2.7

## 鉴权服务部署流程

1、获取部署包安装文件下载链接，下载部署包；

下载完成的文件示例如下：609859F08F4B4FB782948D669EE3CFE3.tar.gz；

2、执行以下命令解压部署包；

```
tar xzvf 609859F08F4B4FB782948D669EE3CFE3.tar.gz
```

3、解压后进入original目录执行**bash download.sh**命令获取全部安装文件，执行脚本后会自动下载以下安装文件：鉴权服务安装包、人体服务安装包以及docker安装包等基础依赖环境；

```
bash download.sh
```

4、将带有全部安装文件的original文件夹上传到待部署的服务器中；

5、进入以下文件路径；

```
cd original/package/Install/
```

6、执行以下命令完成鉴权服务的安装；

```
python install.py in c-offline-security-server
```

7、安装过程中会自动进行环境检查，如果安装进程停住并提示"Environment checking failed! Please fix them before installation."表明环境检查失败,请先排查失败的环境检查项，再重新执行安装，或者输入continue强制继续安装；

8、安装过程中会提示输入**auth server cluster's ip**、**local server's ip**，两个ip地址均输入本机实际网络地址即可；

9、提示安装成功后，可检查服务是否启动；

```
docker ps ;
```

10、如果服务没有启动，可以使用**docker ps -a**，查看容器id；

11、鉴权服务默认的http端口为8443，通过**netstat -apn | grep 8443**可以看端口是否已经被监听。

## 2. 应用服务部署

### 部署环境需求

环境类别	环境需求
硬件要求	支持物理机、虚拟机部署方式；CPU架构：AMD 64/x86_64；内存：>=32G；主板主频>1833mHZ；硬盘:>=512G（应用服务安装目录磁盘空间>100G）；机器需要在局域网内，且能获取到ip地址
操作系统 (64位)	CentOS 7、Ubuntu 14/16/18、RedHat7.2
网络环境	有线局域网，不支持无线网络
服务端口	请参考接口调用文档

### 应用服务部署流程

#### 单机部署

1、获取部署包安装文件下载链接，下载部署包。

下载完成的文件示例如下：609859F08F4B4FB782948D669EE3CFE3.tar.gz；

2、执行以下命令解压部署包

```
tar zxvf 609859F08F4B4FB782948D669EE3CFE3.tar.gz
```

3、解压后进入original目录执行**bash download.sh**命令获取全部安装文件，执行脚本后会自动下载以下安装文件：鉴权服务安装包、人体服务安装包以及docker安装包等基础依赖环境。

```
bash download.sh
```

4、将带有全部安装文件的original文件夹上传到待部署的服务器中。

5、进入以下文件路径；

```
cd original/package/Install
```

6、查看可用模块，

```
python install.py se
```

7、查看已安装模块；

```
python install.py li
```

8、安装应用服务（xxx为服务名）；

```
python install.py in xxx
```

9、提示安装成功后，可检查服务是否启动

```
docker ps ;
```

10、如果服务没有启动，可以使用docker ps -a, 查看容器id；

单实例启动后，默认的http端口请参考[接口调用文档](#)

11、参考[接口调用](#)文档，开始调用接口服务。

## 多机部署

多机部署方式即多台机器上重复单机部署的操作流程。

部署方法参考单机部署方法。

## License更新说明

适用于已经完成私有化部署后，进行License延期、License扩容、License新增产品授权的场景。

- License延期：延长私有化部署包使用时间。
- License扩容：增加私有化部署的应用实例数，使私有化部署包可在更多的设备上部署。
- License新增产品授权：如部署完人体关键点识别服务后还需要在当前环境（同一台机器或同样的局域网环境）进行人流量统计服务的部署，需进行License变更。

## 如何获取更新后的License

- 1、License延期：在控制台[人体分析-私有部署服务管理/查看详情](#)页面点击延期申请，发起License延期，审核通过后即可获得更新的License证书文件。或与工作人员线下沟通后获取新的License证书文件。
- 2、License扩容：请线下沟通后获取新的License证书文件。
- 3、License新增产品授权：请线下沟通后获取新的License证书文件。

## 更新流程

一键部署工具支持对私有化部署包的License证书进行一键更新，一键更新时不需要下载任何包及安装任何模块，请参考以下流程进行License证书更新。

- 1、获取到包含License的部署包后，解压缩，进入以下文件路径；

```
cd original/package/Install
```

- 2、执行以下命令完成License更新；

```
python install.py lu
```

- 3、操作成功后如下图所示，提示License update successfully即更新完成。

```
[root@localhost Install]# python install.py lu
```



```
2019-06-05 16:55:49,031 - 29135 - install - INFO - start to prepare work directory...
2019-06-05 16:55:49,033 - 29135 - install - INFO - install option: lu, user name: baidu
2019-06-05 16:55:49,039 - 29135 - install - INFO - subprocess start,cmd : cp -rf /home/baolong/original/lc//license/license /home/baidu/work/c-offline-security-server/license
2019-06-05 16:55:49,044 - 29135 - install - INFO - subprocess finished,cmd : cp -rf /home/baolong/original/lc//license/license /home/baidu/work/c-offline-security-server/license
Please wait a moment to complete license update...
2019-06-05 16:56:19,073 - 29135 - install - INFO - License update successfully.
```

## 常见问题

### 1、鉴权服务是做什么用的？

鉴权服务是人体分析私有化应用服务运行的基础，只有在鉴权服务运行成功后，才能启动应用服务。

### 2、鉴权服务启动异常，一直处于"wait to be ready"状态

服务日志在解压后目录的/home/baidu/work/c-offline-security-server/log下，提供该目录下的所有日志文件，可以通过日志定位问题。

跟license文件的签发人确认license的正确性，以及授权文件是否已过期。

### 3、怎样判断鉴权服务是否运行正常

确认鉴权服务进程是否运行：ps aux | grep c-offline-security-server

检查鉴权服务是否可访问，如执行下述命令后，有:baidu-rpc 字样的logo返回，则授权服务启动正常。

```
curl (鉴权服务机器ip):8443
```

### 4、如何重启鉴权服务？

停止服务：cd /home/baidu/work/c-offline-security-server/ && bash start/c-offline-security-server-stop.sh

启动服务：cd /home/baidu/work/c-offline-security-server/ && bash start/c-offline-security-server-start.sh

5、出现'GLIBC\_2.14' not found错误，原因是glibc-2.14没有安装，导致服务启动失败，检查命令：strings /lib64/libc.so.6 |grep GLIBC

6、服务没有连有线局域网，导致鉴权服务失败，日志关键词"machine auth check failure"，建议使用ifconfig命令请检查网络。

7、部署的机器错误，如果提取指纹和安装服务的机器不一致，会导致服务部署失败，参考关键日志"machine auth check failure"

8、机器指纹采集后，服务器ip发生变更，导致授权失败，参考关键日志"machine auth check failure"

9、执行命令 netstat -apn | grep {port}，检测不到8443端口时，提供鉴权服务日志，服务日志在解压后目录的/home/baidu/work/c-offline-security-server/log下，提供该目录下的所有日志文件，可以通过日志定位问题。压缩后发送给百度的技术支持，待技术支持定位问题后，反馈解决方案；

10、安装docker失败时，通过docker info命令查看是否已安装docker，如果未安装成功，检查apt/yum的源配置最近是否操作过变更，且没有进行环境更新或重启，如果是，请重启服务器再重新进行安装；

11、日志窗口出现nginx install fail关键字或者通过docker ps命令无法看到nginx容器时，打开/home/baidu/work/logs/openresty/error.log日志，查看具体的异常信息，如果是端口占用，请关闭服务器上已使用该端口的进程，然后重试；其他无法解决的问题，请联系技术支持。

## 接口调用

Hi, 您好, 欢迎使用百度人体分析私有化部署产品。

人体分析私有化部署包部署成功后, 即可获得与在线API基本完全相同的接口, 相关接口将会启动, 即可参考本文档开始调用测试。

人体分析的各个接口拆分为不同的私有部署包, 人体关键点识别、人流量统计、人体检测、人体属性识别(单人版)、手势识别、人像分割、手部关键点识别、驾驶行为分析对应8个不同的部署包, 方便选取所需能力灵活应用。

## 🔗 接口能力介绍

### 1、人体关键点识别

检测图片中的所有人体, 识别每个人体的21个主要关键点, 包含四肢、脖颈、五官等部位, 同时可输出人体的坐标信息。支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

- 21个关键点的位置: 头顶、左眼、右眼、左耳、右耳、左嘴角、右嘴角、鼻子、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髋部、右髋部、左膝、右膝、左脚踝、右脚踝。

注: 接口会返回人体坐标框和每个关键点的置信度分数, 在应用时可综合置信度score分数, 过滤掉置信度低的“无效人体”, 推荐的过滤阈值在下文“[接口调用说明](#)”部分展开。

### 2、人体检测

检测图像中的所有人体, 返回每个人体的矩形框位置; 支持人体重叠、遮挡、截断、背面、侧面、动作变化等复杂场景。人体像素需大于60px \* 60px。

### 3、人体属性识别(单人版)

输入单个人体的图片, 识别人体的静态属性和行为, 共支持28种属性。主要适用于中低空大角度斜拍视角, 支持人体轻度重叠、轻度遮挡、背面、侧面等不同拍摄角度。

- 可识别28种属性: 性别、年龄阶段、下身服饰类别、上身服饰类别、戴帽子(可区分普通帽/安全帽)、戴口罩、上身服饰颜色、下身服饰颜色、使用手机、吸烟、身体朝向、撑伞、背包、是否有交通工具……

注: 接口返回的属性信息包括人体的遮挡、截断情况, 在应用时可基于此过滤掉“无效人体”, 比如严重遮挡、严重截断的人体。

### 4、人流量统计

识别和统计图像当中的人体个数(静态统计, 不支持追踪和去重); 支持框定多个不规则区域统计局部人数, 同时可输出渲染图片。

适用于3米以上的中远距离俯拍, 5米以上为佳, 以头部为主要识别目标统计人数, 无需正脸、全身照, 适应各类人流密集场景。支持轻度畸变的鱼眼摄像头。

### 5、手势识别

识别图片中的手势类型, 返回手势名称、手势矩形框、概率分数, 可识别24种常见手势, 适用于手势特效、智能家居手势交互等场景。

- 支持的24类手势列表: 拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心(3种)、数字(9种)、Rock、竖中指。

适用于3米以内的拍摄距离, 1米内为佳, 自拍和他人拍摄均支持, 拍摄距离尽量近一些, 否则手势目标太小, 容易漏识别。

### 6、人像分割

识别人体的轮廓范围, 与背景进行分离, 适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片, 返回分割后

的二值结果图、灰度图、透明背景的人像图 (png格式)。美颜、P图等图片美化手段会影响分割效果，请使用原图进行分割。

## 7、手部关键点识别

检测图片中的手部，输出手部坐标框、21个骨节点的坐标信息。当前主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

## 8、驾驶行为分析

针对车载场景，识别驾驶员使用手机、抽烟、不系安全带、未佩戴口罩、闭眼、打哈欠、双手离开方向盘等动作姿态，分析预警危险驾驶行为，提升行车安全性。

### 🔗 接口格式说明

### 🔗 变量类型定义

类型	定义
string	普通的字符串，可能会有长度要求，具体参见接口说明中的备注
uint32	整形数字，最大取值为4字节int。自然数
int64	整形数字，最大取值为8字节int。允许负数
json	无论是request还是response中某个字段定义为json，那么它其实是一个json格式的字符串，需要二次解析
array	request的query中表示array请使用key[]。response的json中的array即为jsonArray
double	双精度，小数点后最大8位四舍五入

### 🔗 返回格式

- **error\_code**、**error\_msg**即错误码和错误描述，详细含义请参考错误码表，**error\_code**为0代表请求成功
- **result**是接口返回的详细信息，格式为数组。
- **log\_id**是请求的日志id，13位长(**bigint**)，用于定位请求。

```
{
  "error_code": 0, //错误码 0代表成功
  "error_msg": "SUCCESS", //错误信息
  "result": {...} //返回结果 具体内容详见相关接口
  "log_id": 3535325235 //请求的日志id
  "timestamp": 1512391548 //请求到达的时间戳 精确到秒级
  "cached": 0 //未启用 无需处理
}
```

### 🔗 接口调用说明

### 🔗 人体关键点识别

检测图片中的所有人体，识别每个人体的21个主要关键点，包含四肢、脖颈、五官等部位，同时可输出人体的坐标信息。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手



腕、右手腕、左腕部、右腕部、左膝、右膝、左脚踝、右脚踝。示意图如下：



调用接口的地址示例：[192.168.0.1]:8124/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8124

#### 路径

/GeneralClassifyService/classify

#### 请求参数

参数	必选	类型	可选值范围	说明
image	true	string	0-255彩色图像， size >50	图像数据，base64编码，图片长宽比需介于0.1-10之间，图片尺寸长宽小于50pixel时，会提示尺寸过小

#### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data)
    }
  )
}
```

python代码示例如下：

```

import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data)
}

request_body = {
    # 将data转为json, 并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)

```

## 返回参数

接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法**：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.2时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.3甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object数组	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	置信度分数，取值0-1，越接近1代表识别准确的概率越大
+body_parts	是	object	身体部位信息，包含21个关键点
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数

+++score	是	float	置信度分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++neck	是	object	颈
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标

+++score	是	float	置信度分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++right_ear	是	object	右耳朵
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数
++left_ear	是	object	左耳朵
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	置信度分数

说明：body\_parts，一共21个part，每个part包含x，y两个坐标，如果part被截断，则x、y坐标为part被截断的图片边界位

置，part顺序以实际返回顺序为准。

### 返回示例

```
{
  "err_no": 0,
  "err_msg": "fg_human_service[status:succeed]",
  "result": {
    "person_num": 1,
    "person_info": [
      {
        "body_parts": {
          "left_ankle": {
            "score": 0.868368387222290,
            "x": 643.8750,
            "y": 649.6250
          },
          "left_ear": {
            "score": 0.8685630559921265,
            "x": 643.8750,
            "y": 303.68750
          },
          "left_elbow": {
            "score": 0.8781360983848572,
            "x": 677.6250,
            "y": 413.3750
          },
          "left_eye": {
            "score": 0.9187903404235840,
            "x": 635.43750,
            "y": 303.68750
          },
          "left_hip": {
            "score": 0.8074261546134949,
            "x": 652.31250,
            "y": 480.8750
          },
          "left_knee": {
            "score": 0.7978388071060181,
            "x": 652.31250,
            "y": 565.250
          },
          "left_mouth_corner": {
            "score": 0.8915061950683594,
            "x": 635.43750,
            "y": 320.56250
          },
          "left_shoulder": {
            "score": 0.8677763342857361,
            "x": 660.750,
            "y": 354.31250
          },
          "left_wrist": {
            "score": 0.8828375339508057,
            "x": 702.93750,
            "y": 464.0
          },
          "neck": {
            "score": 0.8687257170677185,
            "x": 627.0,
            "y": 337.43750
          },
          "nose": {
```

```
"score":0.9034447669982910,
"x":627.0,
"y":312.1250
},
"right_ankle":{
"score":0.8528900742530823,
"x":610.1250,
"y":649.6250
},
"right_ear":{
"score":0.8603078126907349,
"x":610.1250,
"y":303.68750
},
"right_elbow":{
"score":0.8440912961959839,
"x":559.50,
"y":413.3750
},
"right_eye":{
"score":0.8926745653152466,
"x":618.56250,
"y":303.68750
},
"right_hip":{
"score":0.7821152210235596,
"x":601.68750,
"y":480.8750
},
"right_knee":{
"score":0.7996079921722412,
"x":610.1250,
"y":573.68750
},
"right_mouth_corner":{
"score":0.8937355875968933,
"x":618.56250,
"y":320.56250
},
"right_shoulder":{
"score":0.8571984171867371,
"x":584.81250,
"y":354.31250
},
"right_wrist":{
"score":0.8279093503952026,
"x":542.6250,
"y":455.56250
},
"top_head":{
"score":0.8723988533020020,
"x":627.0,
"y":278.3750
}
},
"location": {
"height":449.7995910644531,
"left":514.0347900390625,
"score":0.9874296784400940,
"top":256.1484069824219,
"width":226.6530609130859
}
```

```

    }
  ]
},
"format": "json",
"img": "./data/img/normal_109.jpg" //这个字段为测试端新加，为方便核对图片结果，与服务端无关。
}

```

### 错误返回值

参数	类型	必选	说明
err_msg	string	是	错误信息，只在异常中出现（参考错误码）
err_code	unit32	是	错误码，只在异常中出现（参考错误码）

### 错误码表

err_code	err_msg	解释
0	Succeed!	请求成功
216100	Failed to parse input json	json读取错误
216201	Failed to load the image	图片加载出错
216200	Image empty	图像数据为空
216202	Image size error	图片尺寸错误
216202	Image size is too small	图片太小，比如小于10像素
216202	Invalid image ratio	图片长宽比超出范围，比如超过10
216203	Image detection error	图片检测错误

## 🔗 人体检测

检测图像中的所有人体，返回每个人体的矩形框位置和置信度分数。人体目标需大于30px \* 30px，否则模型会直接丢弃，无法检测到有效人体。

调用接口的地址示例：[192.168.0.1]:8125/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8125

### 路径

/GeneralClassifyService/classify

### 请求参数

参数	必选	类型	说明
image	true	string	图像数据，base64编码。若图片尺寸长或宽小于50pixel，会提示尺寸过小。
return_crop	false	int	0或1，0表示false，1表示true，表示是否将检测到的人体框裁剪下来。 异常：若该参数存在无法识别的单词，则忽略不进行处理，默认为false，不裁剪人体框

### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data),
      "return_crop": "0,1",
    }
  )
}
```

python代码示例如下：

```
import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data),
    "return_crop": 0,
}

request_body = {
    # 将data转为json，并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)
```

## 返回参数

字段	必选	类型	说明
err_msg	true	string	错误信息，只在异常中出现（参考错误码表）
err_no	true	uint32	错误码，只在异常中出现（参考错误码表）
format	true	string	返回格式说明，默认添加，值为"json"
result	true	string	人体检测结果，服务对返回结果做了base64编码

正确返回值说明（返回参数为base64编码格式，将result字段base64解码后可得到以下内容）：



字段	类型	说明
person_num	int	检测到的人体框数目
det_res	object数组	每个人体框的具体信息
+classname	string	类型名称，默认为person
+left	int	检测框左坐标
+top	int	检测框顶坐标
+width	int	检测框宽度
+height	int	检测框高度
+probability	float	对应人体框的概率分数，取值0-1，越接近1代表识别准确的概率越大
+label	int	下标，默认为1，无实际含义
+crop	string	如果打开return_crop的请求参数为true，这里会返回该人体的patch图像，base64编码

### 返回示例

```
{
  "person_num": 3,
  "det_res":
  [
    {
      "classname": "person",
      "crop": "",
      "left": 1092,
      "top": 594,
      "width": 112,
      "height": 200,
      "probability": 0.6849,
      "label": 1
    },
    {
      "classname": "person",
      "crop": "",
      "left": 1263,
      "top": 281,
      "width": 63,
      "height": 130,
      "probability": 0.5300,
      "label": 1
    },
    {
      "classname": "person",
      "crop": "",
      "left": 969,
      "top": 192,
      "width": 71,
      "height": 75,
      "probability": 0.4311,
      "label": 1
    }
  ]
}
```

### 错误码表

err_no	value	err_msg	解释
GENERAL_CLASSIFY_SUCCEED	0	GeneralClassify [status: human detect succeed]	人体检测成功（即整体流程成功）
GENERAL_CLASSIFY_CONF_FILE_ERR	216401	GeneralClassifyProcessorFactory [status:reading conf file error]	读取conf文件出错
GENERAL_CLASSIFY_BBOX_PREDICT_ERR	216401	GeneralClassify [status: bbox predict error!]	boudingbox检测过程出错
GENERAL_CLASSIFY_INPUT_FORMAT_ERR	216101	GeneralClassify [status:parse input format error]	输入数据中不存在“image”字段
GENERAL_CLASSIFY_INPUT_PARSING_ERR	216401	pedestrian-detection-service [status:input parsing failed]	读取base64输入图片出错
GENERAL_CLASSIFY_IMAGE_EMPTY_ERR	216200	pedestrian-detection-service [status:image empty]	输入数据图片读取结果为空
GENERAL_CLASSIFY_IMAGE_SIZE_ERR	216202	pedestrian-detection-service [status:image size not between 50 and 4096]	输入图片尺寸不在允许范围之内
GENERAL_CLASSIFY_AUTH_ERR	216401	pedestrian-detection-service [auth verify error]	权限问题
GENERAL_CLASSIFY_ENCODE_ERR	216401	pedestrian-detection-service [status: encode error]	人像patch编码成base64字符串

## 🔗 人体属性识别（单人版）

输入单个人体的图片，识别人体的静态属性和行为，共支持28种属性。

**注：模型默认将输入的整张图片当作“1个人体”直接识别属性，需提前使用人体检测模型，将原图中的每个人体检测、裁剪出来，再调用本服务识别属性信息。**

接口返回每个属性的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的属性。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

调用接口的地址示例：[192.168.0.1]:8126/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，**端口默认为：8126**

### 路径

/GeneralClassifyService/classify

### 请求参数

参数	必选	类型	说明
image	true	string	图像数据，base64编码。若图片尺寸长或宽小于50pixel，会提示尺寸过小。
type	false	string	如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔，无空格。（如"gender,headwear,carrying_item,cellphone"）。 <b>不提供该字段则输出默认的17个属性，若需获取其余属性，请将type参数值设定为需要返回的全部属性。</b> 异常情况： ①若该参数存在无法识别的单词，则忽略不进行处理，返回17个默认输出的属性； ②若提供了该参数，但不含任何可识别的合法值，则按err_no=8返回

## 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data" : base64encode(
    {
      "image" : base64encode(binary image data),
      "type" : "gender,headwear,carrying_item,cellphone"
    }
  )
}
```

python代码示例如下：

```
import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data),
    "type": "gender,headwear,carrying_item,cellphone"
}

request_body = {
    # 将data转为json，并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)
```

## type字段说明

ID	type	说明	是否默认输出	类别数	类别取值
0	gender	性别	是	2	男性、女性
1	age	年龄阶段	是	5	幼儿、青少年、青年、中年、老年
2	action	动作姿态	否	4	站立、蹲或坐、走、跑
3	hair_length	发长	否	4	长发、中长发、短发、秃顶
4	bag	背包	是	3	无背包、单肩包、双肩包
5	upper_wear	上身服饰	是	2	长袖、短袖
6	lower_wear	下身服饰	是	5	长裤、短裤、长裙、短裙、不确定
7	upper_color	上身颜色	是	11	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
8	lower_color	下身颜色	是	12	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
9	upper_wear_fing	上身服饰细分	否	9	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套
10	upper_wear_texture	上身服饰纹理	否	4	纯色、图案、碎花、条纹或格子
11	headwear	是否戴帽子	是	3	无帽、普通帽、安全帽
12	glasses	是否戴眼镜	否	4	无眼镜、戴眼镜、戴墨镜、不确定
13	smoke	是否吸烟	是	3	未吸烟、吸烟、不确定
14	cellphone	是否使用手机	是	4	未使用手机、打电话、看手机、不确定
15	orientation	人体朝向	是	4	正面、背面、左侧面、右侧面
16	umbrella	是否打伞	否	2	未打伞、打伞
17	carrying_baby	是否抱小孩	否	2	未抱小孩、抱小孩
18	face_mask	是否戴口罩	是	3	无口罩、戴口罩、不确定
19	glove	是否戴手套	否	2	无手套、戴手套
20	carrying_item	是否有手提物	否	3	无手提物、有手提物、不确定
21	vehicle	是否有交通工具	否	4	无交通工具、骑摩托车、骑自行车、骑三轮车
22	luggage	是否有拉杆箱	否	2	无拉杆箱、有拉杆箱
23	upper_cut	上方截断	是	2	无上方截断、有上方截断
24	lower_cut	下方截断	是	2	无下方截断、有下方截断
25	side_cut	侧方截断	是	2	无侧方截断、有侧方截断
26	occlusion	遮挡情况	是	3	无遮挡、轻度遮挡、重度遮挡
27	is_human	是否是正常人体	是	2	非正常人体、正常人体。用于判断说明人体的截断/遮挡情况，并非判断动物等非人类生物。 正常人体：身体露出大于二分之一的人体，一般以能看到腰部肚挤眼为标准； 非正常人体：严重截断、严重遮挡的人体，一般是看不到肚挤眼的，比如只有个脑袋、一条腿

## 返回参数

字段	必选	类型	说明
err_msg	true	string	错误信息，只在异常中出现（参考错误码表）
err_no	true	uint32	错误码，只在异常中出现（参考错误码表）
format	true	string	返回格式说明，默认添加，值为"json"
result	true	string	人体属性识别结果，服务对返回结果做了base64编码

正确返回值说明（返回参数为base64编码格式，将result字段base64解码后可得到以下内容）：

字段	类型	说明
person_num	int	人体框数目，固定为1
person_info	object数组	每个人体框的具体信息
+location	object	人体框位置，固定为整图范围（即：原图大小）
++left	int	检测框左坐标
++top	int	检测框顶坐标
++width	int	检测框宽度
++height	int	检测框高度
++score	float	人体框的置信度分数，固定为1
+attributes	object	人体属性内容
++gender	object	性别
+++name	string	如"男性"
+++score	float	对应概率分数
++age	object	年龄阶段
+++name	string	如"青年"
+++score	float	对应概率分数
++action	object	动作姿态
+++name	string	如"站立"
+++score	float	对应概率分数
++hair_length	object	发长
+++name	string	如"短发"
+++score	float	对应概率分数
++bag	object	背包
+++name	string	如"双肩包"
+++score	float	对应概率分数
++upper_wear	object	上身服饰
+++name	string	如"短袖"
+++score	float	对应概率分数
++lower_wear	object	下身服饰
+++name	string	如"长裤"
+++score	float	对应概率分数

++upper_color	object	上身颜色
+++name	string	如"白"
+++score	float	对应概率分数
++lower_color	object	下身颜色
+++name	string	如"蓝"
+++score	float	对应概率分数
++upper_wear_fg	object	上身服饰细分
+++name	string	如"衬衫"
+++score	float	对应概率分数
++upper_wear_texture	object	上身服饰纹理
+++name	string	如"纯色"
+++score	float	对应概率分数
++headwear	object	是否戴帽子
+++name	string	如"无帽"
+++score	float	对应概率分数
++glasses	object	是否戴眼镜
+++name	string	如"戴眼镜"
+++score	float	对应概率分数
++smoke	object	是否吸烟
+++name	string	如"未吸烟"
+++score	float	对应概率分数
++cellphone	object	是否使用手机
+++name	string	如"未使用手机"
+++score	float	对应概率分数
++orientation	object	人体朝向
+++name	string	如"右侧面"
+++score	float	对应概率分数
++umbrella	object	是否打伞
+++name	string	如"未打伞"
+++score	float	对应概率分数
++carrying_baby	object	是否抱小孩
+++name	string	如"未抱小孩"
+++score	float	对应概率分数
++face_mask	object	是否戴口罩
+++name	string	如"无口罩"
+++score	float	对应概率分数
++glove	object	是否戴手套
+++name	string	如"无手套"
+++score	float	对应概率分数
++carrying_item	object	是否有手提物

属性名	数据类型	说明
+++name	string	如“有手提物”
+++score	float	对应概率分数
++vehicle	object	是否有交通工具
+++name	string	如“无交通工具”
+++score	float	对应概率分数
++luggage	object	是否有拉杆箱
+++name	string	如“有拉杆箱”
+++score	float	对应概率分数
++upper_cut	object	上方截断
+++name	string	如“无上方截断”
+++score	float	对应概率分数
++lower_cut	object	下方截断
+++name	string	如“无下方截断”
+++score	float	对应概率分数
++side_cut	object	侧方截断
+++name	string	如“无侧方截断”
+++score	float	对应概率分数
++occlusion	object	遮挡情况
+++name	string	如“轻度遮挡”
+++score	float	对应概率分数
++is_human	object	是否是正常人体
+++name	string	对应概率分数
+++score	float	如“正常人体”

说明：接口返回每个属性的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的属性。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

#### 返回示例

```
{
  "person_num": 1,
  "person_info":
  [
    {
      "location":
      {
        "left": 0,
        "top": 0,
        "width": 200,
        "height": 400
        "score": 1.0
      }
      "attributes":
      {
        "gender":
        {
          "name": "男性",
          "score": 0.937
        }
        "hair_length":
        {
          "name": "短发",
          "score": 0.889
        }
        "lower_wear":
        {
          "name": "长裤",
          "score": 0.925
        }
        "upper_wear":
        {
          "name": "短袖",
          "score": 0.774
        }
      }
    }
  ]
}
```

## 错误码表



err_no	value	err_msg	解释
GENERAL_CLASSIFY_SUCCEED	0	GeneralClassify [status: human attrib succeed]	人体属性检测成功（即整体流程成功）
GENERAL_CLASSIFY_CONF_FILE_ERR	1	GeneralClassifyProcessorFactory [status:reading conf file error]	读取conf文件出错
GENERAL_CLASSIFY_BBOX_PREDICT_ERR	2	GeneralClassify [status: bbox predict error!]	boudingbox检测过程出错
GENERAL_CLASSIFY_ATTRIB_PREDICT_ERR	3	GeneralClassify [status: attrib predict error!]	人体属性检测过程出错
GENERAL_CLASSIFY_INPUT_FORMAT_ERR	4	GeneralClassify [status:parse input format error]	输入数据中不存在“image”字段
GENERAL_CLASSIFY_INPUT_PARSING_ERR	5	fg_human_attribute [status:input parsing failed]	读取base64输入图片出错
GENERAL_CLASSIFY_IMAGE_EMPTY_ERR	6	fg_human_attribute [status:image empty]	输入数据图片读取结果为空
GENERAL_CLASSIFY_IMAGE_SIZE_ERR	7	fg_human_attribute [status:image size not between 50 and 4096]	输入图片尺寸不在允许范围之内
GENERAL_CLASSIFY_ATTRIB_TYPE_PARSING_ERR	8	fg_human_attribute [status:attrib type parsing failed]	读取提供的type参数出错

## 🔗 人流量统计

识别和统计图像当中的人体个数（静态统计，不支持追踪和去重）。默认识别整图中的人数，支持指定多个不规则区域统计局部人数，同时可输出渲染图片（会增加接口延时）。

适用于3米以上的中远距离俯拍，5米以上为佳，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景。支持轻度畸变的鱼眼摄像头。

渲染图示例如下：



调用接口的地址示例：[192.168.0.1]:8122/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8122

#### 路径

/GeneralClassifyService/classify

#### 请求参数

参数	必选	类型	说明
image	true	string	图像数据，base64编码，图片尺寸长宽小于50pixel，会提示尺寸过小
area	false	array < array < int > >	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</p> <p>area参数设置错误的示例：某个坐标超过原图大小（area坐标取值需&lt;原图，如原图为1080 * 720，area取值范围为1~1079、1~719），x、y坐标未成对出现等；</p> <p>注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p>area参数设置格式：</p> <ol style="list-style-type: none"> <li>1) 多个区域用英文逗号“,”分隔；</li> <li>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</li> </ol> <p>示例：[[xa1,ya1,xa2,ya2,...,xan,yan], [xb1,yb1,xb2,yb2,...,xbn,ybn], ...]</p>
show	false	int	<p>取值范围：0，1，2，3</p> <p>是否返回热力图渲染结果，含义如下：</p> <p>0：不返回渲染图（性能最高）</p> <p>1：只返回渲染图，如果设定了area区域参数，同时返回区域框</p> <p>2：返回渲染图+左上角的总人数，如果设置了area参数，也会返回区域框</p> <p>3：返回渲染图+左上角的总人数，如果设置了area参数，不会返回区域框</p> <p>小于0按0计，大于3按3计。上述结果中，渲染图为红云效果，区域框为蓝线绘制（如果设置了area参数）</p>

#### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data)
    }
  )
}
```

python代码示例如下：

```
import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data)
}

request_body = {
    # 将data转为json, 并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)
```

## 返回参数

字段	必选	类型	说明
err_msg	true	string	错误信息，只在异常中出现（参考错误码表）
err_no	true	uint32	错误码，只在异常中出现（参考错误码表）
format	true	string	返回格式说明，默认添加，值为“json”
result	true	string	人流密度估计结果，服务对返回结果做了base64编码

正确返回值说明：（返回参数为base64编码格式，将result字段base64解码后可得到以下内容）

字段名称	类型	说明
person_num	int32	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	string	渲染图片文件byte内容的base64编码，请求端得到后先做解码再以字节流形式直接写入文件
area_counts	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

特别说明：

- 1) person\_num固定返回，image只有当请求字段中"show"为true时候返回；
- 2) image 的内容是图片字节内容做byte64编码的string字符，拿到后反编码再按照字节写成文件即可；

返回示例


```

{"person_num": 86} // 请求时，show为0或者不传
{"person_num": 86, "image": "/9j/4AAoFS2P/9k="} // 请求时，show为1

```

错误码表

err_code	err_msg	名称	说明
0	fg-crowd-counting [status:succeed]	GENERAL_CLASSIFY _SUCCEED	整体流程成功
1	fg-crowd-counting [status:input format illegal]	GENERAL_CLASSIFY _INPUT_FORMAT_ERR	请求未包含image
2	fg-crowd-counting [status:input parsing failed]	GENERAL_CLASSIFY _INPUT_PARSING_ERR	输入格式有误
3	fg-crowd-counting [status:image empty]	GENERAL_CLASSIFY _IMAGE_EMPTY	图片为空
4	fg-crowd-counting [status:image size < 50]	GENERAL_CLASSIFY _IMAGE_SIZE_ERR	图片尺寸过小
5	fg-crowd-counting [status:heatmap prediction fail]	GENERAL_CLASSIFY _HM_PREDICT_ERR	热力图forward出错
6	fg-crowd-counting [auth verify error]	GENERAL_CLASSIFY _AUTH_ERR	鉴权错误
7	fg-crowd-counting [status:detection prediction fail]	GENERAL_CLASSIFY _BBOX_PREDICT_ERR	检测器forward出错

 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种常见手势：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。每种手势的示例图参考：<https://ai.baidu.com/ai-doc/BODY/4k3cpywrv>

适用于3米以内的拍摄距离，1米内为佳，自拍和他人拍摄均支持，拍摄距离尽量近一些，否则手势目标太小，容易漏识别。

调用接口的地址示例：[192.168.0.1]:8120/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，**端口默认为：8120**

#### 路径

/GeneralClassifyService/classify

#### 请求参数

参数	是否必选	类型	说明
appid	false	string	固定值，示例：123456
logid	false	int	随机数
format	false	string	固定值，示例：json
from	false	string	固定值，示例：test-python
cmdid	false	string	固定值，示例：123
clientip	false	string	固定值，示例：0.0.0.0
data	true	string	图片的base64编码字符串

#### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data),
    }
  )
}
```

python代码示例如下：

```

import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data)
}

request_body = {
    # 将data转为json, 并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)

```

## 返回参数

字段	是否必选	类型	说明
err_msg	true	string	返回错误信息
err_no	true	int	返回错误代码
format	true	string	固定值, json
result	true	json	包含所有检测到的目标
+object	true	json	检测到的一个实例
++classname	false	string	目标所属的类别, 即手势名称
++label	false	int	目标的标签
++left	false	int	目标框最左坐标
++top	false	int	目标框最上坐标
++width	false	int	目标框的宽
++height	false	int	目标框的高
++probability	false	float	目标属于该类别的概率

其中classname和label的对应关系如下表所示：

label	classname	手势中文名
1	One	数字1 (也可以叫做食指)
2	Five	数字5 (也可以叫做掌心向前)
3	Fist	拳头
4	Ok	OK
5	Prayer	祈祷
6	Congratulation	作揖 (也可以叫恭喜)
7	Honour	作别
8	Heart_single	单手比心
9	Thumb_up	点赞
10	Thumb_down	diss
11	ILY	我爱你
12	Palm_up	掌心向上
13	Face	人脸
14	Heart_1	双手比心1
15	Heart_2	双手比心2
16	Heart_3	双手比心3
17	Two	数字2 (也可以叫做比V)
18	Three	数字3
19	Four	数字4
20	Six	数字6
21	Seven	数字7
22	Eight	数字8
23	Nine	数字9
24	Rock	摇滚
25	Insult	竖中指

返回示例：

```

{
  u'err_no': 0,
  u'err_msg': u'DetectionSsd[status:succeed]',
  u'result':
    '{
      "object":
        [
          {
            "classname":"Thumb_up",
            "height":145,
            "label":9,
            "left":453,
            "probability":0.5996769666671753,
            "top":563,
            "width":106
          }
        ]
    }
  }\n',
  u'format': u'json'
}

```

### 错误码表

错误码	错误信息	说明
0	DetectionSsd [status:succeed]	成功检测并返回结果
216100	gestureservice [status:Failed to parse input json]	输入参数解析失败
216101	gestureservice [status:Invalid image string in input json]	输入参数不足
216200	gestureservice [status:image empty]	输入图片为空
216201	gestureservice [status:Failed to load the image]	输入图片解析失败
216202	gestureservice [status:Image size is too small]	输入图片尺寸过小或者裁切后的图片尺寸过小
216203	gestureservice [status:Image detection error]	图片可以正常解析，但图片无法通过网络处理
216204	gestureservice [status:Not authenticated]	请求没有被授权

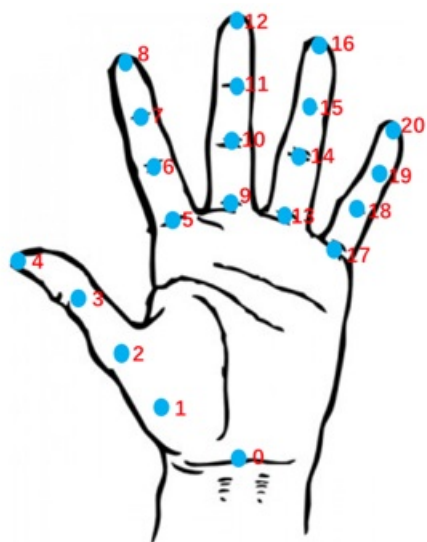
### 🔗 手部关键点识别

检测图片中的所有手部，返回每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

21个关键点对应位置示意图：





调用接口的地址示例：[192.168.0.1]:8128/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8128

### 路径

/GeneralClassifyService/classify

### 请求参数

参数	是否必选	类型	说明
appid	false	string	固定值，示例：123456
logid	false	int	随机数
format	false	string	固定值，示例：json
from	false	string	固定值，示例：test-python
cmdid	false	string	固定值，示例：123
clientip	false	string	固定值，示例：0.0.0.0
data	true	string	图片的base64编码字符串

### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data),
    }
  )
}
```

python代码示例如下：

```

import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data)
}

request_body = {
    # 将data转为json, 并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)

```

## 返回参数

字段	是否必选	类型	说明
err_no	true	string	错误码, 只在异常中出现 (参考错误码表)
err_msg	true	uint32	错误信息, 只在异常中出现 (参考错误码表)
result	true	string	base64编码的手部关键点识别结果

正确返回值说明：(返回参数为base64编码格式, 将result字段base64解码后可得到以下内容)

字段名称	类型	说明
hand_info	object数组	检测到的所有手部信息
+hand_num	uint32	检测到的手部数量
+location	object	手部所在的位置信息
++left	uint32	手部区域离左边界的距离
++top	uint32	手部区域离上边界的距离
++height	uint32	手部区域的高度
++width	uint32	手部区域的宽度
++score	float	手部的置信度分数
+hand_parts	object	单个手部的关键点信息, 包含21个关键点
++[0-20]	object	检测到的关键点
+++x	uint32	关键点的x坐标
+++y	uint32	关键点中的y坐标
+++score	float	关键点的置信度分数

## 返回示例

```

{
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        .....
        "20": {
          "y": 325,
          "x": 567,
          "score": 0.91110396385193
        }
      },
      "location": {
        "height": 556,
        "width": 426,
        "top": 151,
        "score": 17.495880126953,
        "left": 567
      }
    }
  ]
}

```

## 错误码表

err_code	err_msg	说明
0	Succeed! Congrats, May the force be with you ...	成功检测并返回结果
216100	Failed to parse input json	输入参数解析失败
216101	Invalid image string in input json	输入参数不足
216200	Image empty	输入图片为空
216201	Failed to load the image	输入图片解析失败
216202	Image size is too small (less than 10 pixels)	输入图片尺寸过小或者裁切后的图片尺寸过小
216203	Image detection error	图片可以正常解析, 但图片无法通过网络处理
216401	Not authenticated	请求没有被授权

## 人像分割

识别人体的轮廓范围, 与背景进行分离, 返回分割后的二值结果图、灰度图、透明背景的人像图 (png格式)。分割效果示例图请参考: <https://ai.baidu.com/ai-doc/BODY/Fk3cpyxua>

美颜、P图等图片美化手段会影响分割效果, 请使用原图进行分割。

调用接口的地址示例：[192.168.0.1]:8127/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8127

### 路径

/GeneralClassifyService/classify

### 请求参数

参数	必选	类型	说明
image	true	bytes	图像数据，base64编码。必须字段，待分割的图片

### 请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64.b64encode(data)
}
```

python 2.x 代码示例如下：

```
import base64
import json

def encode_img_to_file(image_data):
    img_encode = cv2.imencode('.png', image_data)[1]
    data_encode = np.array(img_encode)
    str_encode = data_encode.tostring()
    return str_encode

file = "test.jpg"
image = cv2.imread(file, -1)
img_data = encode_img_to_file(image)
proto_data.image = img_data
data = proto_data.SerializeToString()

data = {
    # 将图片进行json编码
    "data": base64.b64encode(image_string)
}

#### 将数据转为json字符串
request_body = json.dump(data)

#### 最终应该传入http body的内容
print json.dumps(request_body)
```

python 3.x 代码示例下载地址：<https://ai.baidu.com/file/8044251026DB40619E8C8E77AB4152A5>

### 返回参数

字段	必选	类型	说明
err_no	true	int	返回错误代码
err_msg	true	string	返回错误信息
result	true	string	分割结果信息
+type	true	string	分割类型，目前仅支持person
+labelmap	true	string	分割后的二值图结果，经过base64之后的字符串，单通道图片；需二次处理方能查看分割效果，Python、Java的处理示例代码见： <a href="https://ai.baidu.com/docs#/Body-API/a28d94ec">https://ai.baidu.com/docs#/Body-API/a28d94ec</a>
+scoremap	true	string	分割后人像前景的scoremap，归一到0-255，单通道图片；不用进行二次处理，直接解码保存图片即可
+foreground	true	string	分割后的人像前景抠图结果，png图片，透明背景，四通道；不用进行二次处理，直接解码保存图片即可
format	true	string	固定值，json

### 返回示例

```
{
  "err_no": 0,
  "err_msg": "ParseService[status:succeed]",
  "result": {
    "labelmap": "",
    "scoremap": "",
    "type": "person",
    "foreground": ""
  },
  "format": "json"
}
```

### 错误码表

err_no	解释
216100	参数无效或者错误
216200	无效图片
216201	图片格式错误
216401	预测过程出错

### 🔗 驾驶行为分析

针对车内驾驶室监控画面，识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

#### 图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识](#)

别服务。

- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

调用接口的地址示例：[192.168.0.1]:8132/GeneralClassifyService/classify，其中ip需要替换为用户自己服务器的ip，端口默认为：8132

路径

/GeneralClassifyService/classify

请求参数

参数	必选	类型	说明
image	true	string	图像数据，base64编码。支持图片格式：jpg、bmp、png，若图片尺寸长或宽小于50pixel，会提示尺寸过小。
type	否	string	如只需识别某几个属性，可用type参数控制接口返回的属性list，英文逗号分隔，如： smoke,cellphone,not_buckling_up；默认所有属性都识别，目前支持的属性列表如下： smoke // 吸烟， cellphone // 使用手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	<b>有效取值范围：0,1</b> 默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

请求参数构造及python代码示例

请求参数为json格式，请求时请将Content-Type设置为application/json格式。

请求参数格式如下：

```
{
  "data": base64encode(
    {
      "image": base64encode(binary image data),
      "type": "smoke,cellphone"
    }
  )
}
```

python代码示例如下：

```
import base64
import json

#### 输入图片为/home/work/01.jpg
image_file = "/home/work/01.jpg"

#### 将图片内容读取至image_data
with open(image_file, 'rb') as f:
    image_data = f.read()

data = {
    # 将image_data进行base64编码
    "image": base64.b64encode(image_data),
    "type": "smoke,cellphone"
}

request_body = {
    # 将data转为json, 并进行base64编码
    "data": base64.b64encode(json.dumps(data))
}

#### 最终应该传入http body的内容
print json.dumps(request_body)
```

## 返回参数

字段	必选	类型	说明
err_msg	true	string	错误信息，只在异常中出现（参考错误码表）
err_no	true	uint32	错误码，只在异常中出现（参考错误码表）
format	true	string	返回格式说明，默认添加，值为"json"
result	true	string	属性识别结果，服务对返回结果做了base64编码

正确返回值说明（返回参数为base64编码格式，将result字段base64解码后可得到以下内容）：

字段	类型	说明
person_num	int	人体框数目，固定为1
person_info	object数组	每个人体框的具体信息
+location	object	人体框位置，固定为整图范围（即：原图大小）
++left	int	检测框左坐标
++top	int	检测框顶坐标
++width	int	检测框宽度
++height	int	检测框高度
++score	float	人体框的置信度分数，固定为1
+attributes	object	人体属性内容
++gender	object	性别
+++name	string	如"男性"
+++score	float	对应概率分数

++age	object	年龄阶段
+++name	string	如"青年"
+++score	float	对应概率分数
++action	object	动作姿态
+++name	string	如"站立"
+++score	float	对应概率分数
++hair_length	object	发长
+++name	string	如"短发"
+++score	float	对应概率分数
++bag	object	背包
+++name	string	如"双肩包"
+++score	float	对应概率分数
++upper_wear	object	上身服饰
+++name	string	如"短袖"
+++score	float	对应概率分数
++lower_wear	object	下身服饰
+++name	string	如"长裤"
+++score	float	对应概率分数
++upper_color	object	上身颜色
+++name	string	如"白"
+++score	float	对应概率分数
++lower_color	object	下身颜色
+++name	string	如"蓝"
+++score	float	对应概率分数
++upper_wear_fg	object	上身服饰细分
+++name	string	如"衬衫"
+++score	float	对应概率分数
++upper_wear_texture	object	上身服饰纹理
+++name	string	如"纯色"
+++score	float	对应概率分数
++headwear	object	是否戴帽子
+++name	string	如"无帽"
+++score	float	对应概率分数
++glasses	object	是否戴眼镜
+++name	string	如"戴眼镜"
+++score	float	对应概率分数
++smoke	object	是否吸烟
+++name	string	如"未吸烟"
+++score	float	对应概率分数
++cellphone	object	是否使用手机



++cellphone	object	是否使用手机
+++name	string	如"未使用手机"
+++score	float	对应概率分数
++orientation	object	人体朝向
+++name	string	如"右侧面"
+++score	float	对应概率分数
++umbrella	object	是否打伞
+++name	string	如"未打伞"
+++score	float	对应概率分数
++carrying_baby	object	是否抱小孩
+++name	string	如"未抱小孩"
+++score	float	对应概率分数
++face_mask	object	是否戴口罩
+++name	string	如"无口罩"
+++score	float	对应概率分数
++glove	object	是否戴手套
+++name	string	如"无手套"
+++score	float	对应概率分数
++carrying_item	object	是否有手提物
+++name	string	如"有手提物"
+++score	float	对应概率分数
++vehicle	object	是否有交通工具
+++name	string	如"无交通工具"
+++score	float	对应概率分数
++luggage	object	是否有拉杆箱
+++name	string	如"有拉杆箱"
+++score	float	对应概率分数
++upper_cut	object	上方截断
+++name	string	如"无上方截断"
+++score	float	对应概率分数
++lower_cut	object	下方截断
+++name	string	如"无下方截断"
+++score	float	对应概率分数
++side_cut	object	侧方截断
+++name	string	如"无侧方截断"
+++score	float	对应概率分数
++occlusion	object	遮挡情况
+++name	string	如"轻度遮挡"
+++score	float	对应概率分数
++is_human	object	是否是正常人体

+++name	string	对应概率分数
+++score	float	如"正常人体"

说明：接口返回每个属性的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的属性。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

### 返回示例

```
{
  "person_num": 1,
  "person_info":
  [
    {
      "location":
      {
        "left": 0,
        "top": 0,
        "width": 200,
        "height": 400
        "score": 1.0
      }
      "attributes":
      {
        "gender":
        {
          "name": "男性",
          "score": 0.937
        }
        "hair_length":
        {
          "name": "短发",
          "score": 0.889
        }
        "lower_wear":
        {
          "name": "长裤",
          "score": 0.925
        }
        "upper_wear":
        {
          "name": "短袖",
          "score": 0.774
        }
      }
    }
  ]
}
```

### 错误码表

err_no	value	err_msg	解释
GENERAL_CLASSIFY_SUCCEED	0	GeneralClassify [status: human attrib succeed]	人体属性检测成功（即整体流程成功）
GENERAL_CLASSIFY_CONF_FILE_ERR	1	GeneralClassifyProcessorFactory [status:reading conf file error]	读取conf文件出错
GENERAL_CLASSIFY_BBOX_PREDICT_ERR	2	GeneralClassify [status: bbox predict error!]	boudingbox检测过程出错
GENERAL_CLASSIFY_ATTRIB_PREDICT_ERR	3	GeneralClassify [status: attrib predict error!]	人体属性检测过程出错
GENERAL_CLASSIFY_INPUT_FORMAT_ERR	4	GeneralClassify [status:parse input format error]	输入数据中不存在“image”字段
GENERAL_CLASSIFY_INPUT_PARSING_ERR	5	fg_human_attribute [status:input parsing failed]	读取base64输入图片出错
GENERAL_CLASSIFY_IMAGE_EMPTY_ERR	6	fg_human_attribute [status:image empty]	输入数据图片读取结果为空
GENERAL_CLASSIFY_IMAGE_SIZE_ERR	7	fg_human_attribute [status:image size not between 50 and 4096]	输入图片尺寸不在允许范围之内
GENERAL_CLASSIFY_ATTRIB_TYPE_PARSING_ERR	8	fg_human_attribute [status:attrib type parsing failed]	读取提供的type参数出错

## 常见问题

### 🔗 基础问题

#### Q：人体分析可以直接分析视频流吗？

A：目前除危险行为识别接口外，其余在线接口仅支持对静态图片进行识别分析，即将陆续开放离线SDK、软硬一体方案，以满足视频流的处理分析需求。

#### Q：人体分析对摄像头有要求么？图片质量会影响识别效果吗？

A：人体分析服务对摄像头选型无特殊要求，市面上主流枪机、球机、IPC、USB摄像头拍摄的图像均支持。如果图片质量较差，会一定程度上影响模型效果。对于超高清图片建议压缩后识别，对于图片质量差的图片，建议提升图片清晰度，推荐分辨率720 P以上的图片，更低分辨率的图片也能识别，只是效果可能有差异，模糊、遮挡严重、光线暗等情况下，识别效果肯定不理想。

#### Q：人体分析可以识别夜间红外监控图片么？

A：红外图片缺少颜色特征，识别效果与可见光图片相比，有一定差异；当前[人体检测与属性识别](#)、[驾驶行为分析](#)这2个服务已扩展支持红外图片，但识别效果需实测评估，其余服务暂不支持红外场景。

### 🔗 价格问题

#### Q：人体分析在线接口如何收费？未上线计费的接口能否申请提额？

A：目前人体分析在线服务已上线标准计费功能，各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格](#)文档。

未上线计费功能的接口，若免费限额不够，可以[提交工单](#)、或加入官方QQ群（860337848）联系群管申请提额。

**Q：人体私有化部署方案如何收费？**

A：人体分析私有化部署包为付费产品，可自用或集成为客户使用，需具备基本的IT能力进行部署使用。

- 免费测试：人体分析的各个部署包，均提供免费测试，最高可获得3个月免费测试权限。
- 正式购买：按照授权数量计费（具体授权方式见[授权说明](#)），不同功能的部署包分别计费，支持按年授权、永久授权两种购买方式，具体价格请联系客户经理沟通。

**🔗 场景可行性****Q：人体分析可以识别吸烟、打电话行为吗？**

A：人体检测与属性识别、驾驶行为分析2个服务均可识别吸烟、打电话行为。

- 如果是普通室内外监控场景，请使用人体检测与属性识别服务，可直接在线测试识别效果：<https://ai.baidu.com/tech/body/attr>，支持服务器端私有化部署（本地服务器和专有云服务器均支持），详细说明请参考：<https://ai.baidu.com/tech/body/private>
- 如果是车载场景，请使用驾驶行为分析服务，可直接在线测试识别效果：<http://ai.baidu.com/tech/body/driver>，该服务预计2020年2月推出离线SDK，可集成到车载硬件设备中，在设备端离线使用，如有需求，可[提交工单](#)或者[合作咨询](#)联系我们，以便后续及时对接测试。

**Q：人体分析可以识别人员在岗/离岗/睡觉行为吗？**

A：可使用人体检测与属性识别服务、或人体关键点识别服务，上述2个服务均可判断画面中是否有人、并返回每个人体的坐标位置，基于接口返回的上述信息，在摄像头画面中设定识别区域，即可实现在岗/离岗的业务判断逻辑。

同时，百度近期推出的一站式[视频监控系统配置平台EasyMonitor](#)，已经预置了“离岗检测”、“睡岗检测”等业务技能，可直接接入视频流，进行技能配置，查看监控分析结果，零开发门槛，欢迎使用。

**Q：人体分析可以识别人员是否穿戴工作服/佩戴安全帽吗？**

A：可使用人体检测与属性识别服务，通过返回字段“是否戴帽子”及“服饰颜色”、“服饰纹理特征”结果，辅助判断。

同时，百度近期推出的一站式[视频监控系统配置平台EasyMonitor](#)，已经预置了“安全帽佩戴合规检测”等业务技能，可直接接入视频流，进行技能配置，查看监控分析结果，零开发门槛，欢迎使用。

**Q：人体分析可以识别摔倒或打架行为吗？**

A：可以，百度近期推出的[危险行为识别](#)服务支持识别摔倒、砸东西、打架等行为，可以[提交工单](#)、或加入官方QQ群（860337848）联系群管申请开通权限，进行试用。

**Q：人体分析可以通过图片得到身高、三围、脚长等具体信息吗？**

A：通过人体关键点信息推算肩宽、身高等信息不可行，图片中人体远近、大小有区别，不能进行比例估算。

[人体关键点识别](#)目前可识别的21个关键点位置如下：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髌部、右髌部、左膝、右膝、左脚踝、右脚踝。

**Q：手势识别可以识别手势的方向么？我需要自定义手势动作**

A：目前没有动态的手势识别，可以用静态的手势做个替代方案，比如点赞手势（拇指向上）代表向上，diss手势（拇指向下）代表向下，具体支持的手势类别可以参考文档说明：<https://ai.baidu.com/ai-doc/BODY/4k3cpywrv>

新推出的[手部关键点识别](#)服务，可精准定位手部21个主要骨节点坐标，可用于自定义手势识别，可在线测试体验：<https://ai.baidu.com/tech/body/hand>

**🔗 私有化部署****Q：人体分析支持内网环境使用、服务器私有化部署么？**

A：支持，本地服务器或者专有云服务器（需GPU）均可一键部署，支持人体检测、人体属性识别、人体关键点定位、人流量统计、手势识别、人像分割等功能；完成企业认证，即可在控制台提交测试申请，最高可获得3个月免费测试权

限：<https://console.bce.baidu.com/ai/?fromai=1#/ai/body/package/index>，工作人员会在3个工作日内处理对接。私有化部署方案详细介绍请参考：<https://ai.baidu.com/tech/body/private>

Q：基于企业本地服务器私有化部署的人体识别对服务器GPU有什么要求？

A：人体分析私有化部署的硬件配置选型推荐，请参考：<https://ai.baidu.com/ai-doc/BODY/Ak3cpyw50#硬件配置>

Q：人体分析的本地私有部署是否还需要访问公网？

A：人体分析本地私有部署不需要访问公网，无外部网络依赖，在局域网内使用，数据无需上传至第三方服务器或云端，可有效保障企业核心生产数据的私密性和安全性。

## 🔗 离线 SDK

Q：人体分析支持在设备端离线使用吗？

A：人体关键点识别、手势识别、动态人流量统计Android端的离线SDK已在邀测中，其他开发环境暂未支持，如有需求，请[提交工单](#)或者[合作咨询](#)联系我们申请试用。

同时，近期推出[3D肢体关键点 SDK](#)，如有需求，可在页面上点击[申请试用](#)。

Q：人体分析服务的离线SDK对硬件及摄像头的要求如何？

A：离线SDK硬件推荐rk3399或者同等算力以上的芯片；对于摄像头无特殊要求，720P以上即可。其中：

- 手势识别：主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。
- 动态人流量统计：适用于各种出入口，摄像头高度3m左右为佳，垂直俯拍或大角度斜拍，需要拍到人体头部和肩部，人群过于密集会影响识别效果。抽帧频率需 > 2fps，否则无法有效跟踪，建议5~10fps。

## 🔗 人流量统计

Q：人流统计接口可以实现对单位时间内去重的进出人数统计吗？

A：请使用人流统计（动态版）接口，技术文档：<https://ai.baidu.com/ai-doc/BODY/wk3cpyyog>。

Q：人流统计对摄像头的高度、角度有什么要求吗？

A：静态人流量统计的摄像头架设的建议：高度3m以上，5m以上为佳，垂直俯拍或者大角度斜拍，需要能拍全区域内的人头，适应人群密集场所。

动态人流量统计的摄像头架设的建议：适用于各种出入口，摄像头高度3m左右为佳，垂直俯拍或斜拍，需要拍到人体头部和肩部，人群过于密集会影响识别效果。

Q：人流统计可以实现指定区域的识别与统计吗？

A：目前接口已支持图片特定框选区域的人数统计，最多可同时支持10个区域，调用接口时设置area参数即可，技术文档：<https://ai.baidu.com/ai-doc/BODY/7k3cpyy1t>。

对于环境中存在干扰识别结果的物体，也可以通过设定区域来排除干扰，提升识别效果。

Q：人流统计接口，目前只返回数量和渲染图片，能返回具体人头的坐标集合吗？

A：人流统计模型暂时不支持返回坐标。

如需要人体坐标信息，可使用[人体检测与属性识别](#)、或[人体关键点识别](#)，上述2个服务均可检测识别画面中的所有人，并返回每个人体的坐标位置。但人群密集、或拍摄距离过远时不适用，因为遮挡严重、目标太小，无法有效检测到人体目标。

Q：人流统计目前在什么场景下效果比较好？我测试效果感觉不够理想

A：目前在室内场景，例如机场、商场、教室、展馆等效果都比较好，室外场景，如交通场景、工地场景等，由于场景差异较大，建议先实测评估识别效果。若效果不理想可以[提交工单](#)、或加入官方QQ群（860337848）联系群管进行反馈。

Q：动态人流量分析时，图片与图片之间间隔的最小时间差是多少？是否时间相隔越近，识别正确率越高？

A：不限制最小时间差，最大时差不超过500ms，即：抽帧频率需要大于2fps，建议帧率5~10fps，也就是每100ms~200ms抽一帧。

## 🔗 驾驶行为分析

Q：人体分析在车载场景识别效果好吗？

A：百度推出了[驾驶行为分析](https://ai.baidu.com/tech/body/driver)服务，可以识别驾驶员打手机、抽烟、不系安全带等违规行为，详情参见：<https://ai.baidu.com/tech/body/driver>

该服务预计2020年2月推出离线SDK，可集成到车载硬件设备中，在设备端离线使用，如有需求，可[提交工单](#)或者[合作咨询](#)联系我们，以便后续及时对接测试。

Q：驾驶行为分析的score和threshold代表什么含义？如何使用？

A：score表示属性的置信度，比如抽烟属性的score如果是0.9，说明很大程度上在抽烟；threshold是基于内部测试结果，给出的score阈值建议，假设抽烟属性的threshold是0.5，则表示“建议score≥0.5时，代表抽烟，score < 0.5时，代表没抽烟”。

threshold只是内部给出的建议阈值，仅作为参考，不同业务场景的测试结果可能有差异，需要根据实际测试结果选取合适的阈值，比如您测试了200张抽烟的图片，发现score > 0.6时，真实准确率为80%，score > 0.8时，真实准确率为90%，那么看看业务上需要多高的准确率，再选择一个合理的score阈值。

# 更新日志

## 导览

感谢您关注百度人体分析产品，此文主要用于展示人体各产品线条的更新日志，以便您第一时间了解我们的进步。

如果您在使用我们产品时，遇到任何问题，或对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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[AI社区交流](http://ai.baidu.com/forum/topic/list/197)：<http://ai.baidu.com/forum/topic/list/197>

## 🔗 在线API更新日志

- 在线API接口的更新记录请查阅：[这里](#)

在线API的接口能力列表，可前往[此处](#)查看

## 🔗 私有化部署更新日志

- 私有化部署包的更新记录请查阅：[\[这里\]](#)

私有化部署的方案介绍，可前往[此处](#)查看

## 在线API更新日志

您好，欢迎使用百度人体分析在线API服务。

本文主要展示人体在线API接口的更新日志，以便您第一时间了解我们的进步。

如果您在使用我们产品时，遇到任何问题，或对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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**AI社区交流**：<http://ai.baidu.com/forum/topic/list/197>

🔗 20200617

🔗 人像分割接口升级

- 升级内容：优化算法，改善分割效果，**减少背景误分割、边缘锯齿感等badcase**
- 在线体验：<https://ai.baidu.com/tech/body/seg>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Fk3cpyxua>

🔗 20200604

🔗 人体检测与属性识别接口升级

- 升级内容：优化人体检测的算法，**提升小目标人体、遮挡/截断等高难度场景下的人体识别率**，让模型“认识”更多的人
- 在线体验：<https://ai.baidu.com/tech/body/attr>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Ak3cpyx6v>

🔗 20200521

🔗 驾驶行为分析接口升级

- 升级内容：**专项优化驾驶员“戴口罩”状态下的识别效果**，吸烟、使用手机、未系安全带等重点属性的准确率均有显著提升，同时减少正常图片的误报
- 在线体验：<https://ai.baidu.com/tech/body/driver>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Nk3cpywct>

🔗 20200514

🔗 人像分割（证件照版）接口上线邀测

- 接口能力：针对自拍类单人图片，先基于人脸检测、人体关键点定位，裁剪出符合证件照场景的人像图片（头肩上半身），对裁剪后的图片进行发丝级精细化分割
- 结果输出：接口返回裁剪后的原图、分割结果灰度图、合成的黑底效果图，并输出人脸框坐标、关键点坐标、人像姿态（正脸/侧脸/正肩/侧肩等）等辅助信息，便于筛选结果、二次开发
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Qka6rjc93>

注：邀测的接口，不能直接在控制台调用，可通过百度人体分析官方QQ群（860337848）联系群管、或[提交工单](#)申请开通接口测试权限

🔗 20200416

### 🔗 人流量统计（动态版）接口升级

- 升级内容：**优化出入口、卡口等典型场景下的识别效果**；静态检测，平均错误率下降2个百分点；动态进/出的平均错误率下降4个百分点
- 在线体验：<https://ai.baidu.com/tech/body/num>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/wk3cpyyog>

### 🔗 20200416

### 🔗 人体关键点识别接口升级

- 升级内容：**优化关键点定位的效果**，准确率提升2个百分点
- 在线体验：<https://ai.baidu.com/tech/body/pose>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/0k3cpyxme>

### 🔗 20200305

### 🔗 人像分割接口升级

- 升级内容：**增加人体检测功能**、以便过滤“无人”时仍强行分割的badcase，同时优化主观效果（**减少白边、误分割等**），IOU指标提升1个百分点
- 在线体验：<https://ai.baidu.com/tech/body/seg>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Fk3cpyxua>

### 🔗 人体检测与属性识别接口升级

- 升级内容：**增加“戴口罩”属性**，准确率可达90%以上，人体侧身、低头等场景亦适用
- 在线体验：<https://ai.baidu.com/tech/body/attr>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Ak3cpyx6v>

### 🔗 指尖检测接口上线邀测

- 接口能力：检测图片中的手部，定位食指指尖、及4个辅助关键点的坐标位置，模型针对**儿童学习机点读场景**进行专项调优，可用于点读搜题、自动跟读等应用
- 在线体验：<https://ai.baidu.com/tech/body/fingertip>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Jk7ir38ut>

注：邀测的接口，不能直接在控制台调用，可通过百度人体分析官方QQ群（860337848）联系群管、或[提交工单](#)申请开通接口测试权限

### 🔗 20200227

### 🔗 驾驶行为分析接口升级

- 升级内容：**增加“未正确佩戴口罩、闭眼、打哈欠、低头”4个属性**，并优化“抽烟、使用手机”属性的效果
- 在线体验：<https://ai.baidu.com/tech/body/driver>



- 接口文档：<https://ai.baidu.com/ai-doc/BODY/Nk3cpywct>

🔗 20200220

🔗 人流量统计（动态版）接口升级

- 升级内容：**优化算法效果**，中低空场景下，静态人体头肩检测的平均准确率提升至85%，动态人流统计平均进出准确率提升至95%
- 在线体验：<https://ai.baidu.com/tech/body/num>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/wk3cpyyog>

🔗 20200213

🔗 人体关键点识别接口升级

- 升级内容：**优化人体检测模块的算法效果**，准确率提升2个百分点
- 在线体验：<https://ai.baidu.com/tech/body/pose>
- 接口文档：<https://ai.baidu.com/ai-doc/BODY/0k3cpyxme>

🔗 20200102

🔗 手部关键点识别接口上线计费

- 计费方式：按调用量后付费+次数包2种方式
- 在线体验：<https://ai.baidu.com/tech/body/hand>
- 价格文档：<https://ai.baidu.com/ai-doc/BODY/mk3cpyhfe>

## 历史文档

### C++ SDK文档

#### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对C++开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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**AI社区交流**：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格文档](#)

## 🔗 接口功能说明

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘等行为，可用于分析预警危险驾驶行为
人流量统计（动态版）	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、人机交互、AR特效等场景

## 🔗 版本更新记录

上线日期	版本号	更新内容
2019.12.19	0.8.5	新增手部关键点识别接口
2019.2.22	0.7.11	新增驾驶行为分析接口
2019.1.25	0.7.10	新增动态人流量统计接口
2018.9.29	0.7.6	新增手势识别、人像分割接口
2018.6.8	0.7.2	新增人体分析服务

## 快速入门

### 🔗 安装人体分析 C++ SDK

#### 人体分析 C++ SDK目录结构

```

├── base
│   ├── base.h           // 请求客户端基类
│   ├── base64.h        // base64加密相关类
│   ├── http.h          // http请求封装类
│   └── utils.h         // 工具类
└── bodyanalysis.h      // 人体分析 交互类

```

### 最低支持 C++ 11+

直接使用开发包步骤如下：

1. 在[官方网站](#)下载C++ SDK压缩包。
2. 将下载的aip-cpp-sdk-version.zip解压，其中文件为包含实现代码的头文件。
3. 安装依赖库libcurl（需要支持https） openssl jsoncpp(>1.6.2版本，0.x版本将不被支持)。
4. 编译工程时添加 C++11 支持 (gcc/clang 添加编译参数 -std=c++11)，添加第三方库链接参数 lcurl, lcrypto, ljsoncpp。
5. 在源码中include bodyanalysis.h，引入压缩包中的头文件以使用aip命名空间下的类和方法。

### 新建client

client是人体分析的C++客户端，为使用人体分析的开发人员提供了一系列的交互方法。当您引入了相应头文件后就可以新建一个client对象

用户可以参考如下代码新建一个client：

```

#include "bodyanalysis.h"

// 设置APPID/AK/SK
std::string app_id = "你的 App ID";
std::string api_key = "你的 Api key";
std::string secret_key = "你的 Secret Key";

aip::Bodyanalysis client(app_id, api_key, secret_key);

```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

**注意：**如您以前是百度云的老用户，其中API\_KEY对应百度的“Access Key ID”，SECRET\_KEY对应百度的“Access Key Secret”。

## 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髌部、右髌部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：



多人场景：



```
Json::Value result;
```

```
std::string image;
```

```
aip::get_file_content("/assets/sample.jpg", &image);
```

```
// 调用人体关键点识别
```

```
result = client.body_analysis(image, aip::null);
```

#### 人体关键点识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

#### 人体关键点识别 返回数据参数详情



接口除了返回人体框和每个关键点的坐标信息外，还会输出人体框和关键点的概率分数，实际应用中可以基于概率分数进行过滤，排除掉分数低的误识别“无效人体”，推荐的过滤方案：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.03时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息，包含21个关键点
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标

+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标

+++y	是	float	y坐标
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id，用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，**如果part被截断，则x、y坐标为part被截断的图片边界位置**，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体**。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

#### 人体关键点识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        },
        "top_head": {
          "y": 242.53125,
          "x": 620,
          "score": 0.87757384777069
        },
        "right_mouth_corner": {
          "y": 308.625,
          "x": 666.76125
        }
      }
    }
  ]
}
```

```
"x": 606.78125,
"score": 0.90121293067932
},
"neck": {
  "y": 335.0625,
  "x": 620,
  "score": 0.84662038087845
},
"left_shoulder": {
  "y": 361.5,
  "x": 699.3125,
  "score": 0.83550786972046
},
"left_knee": {
  "y": 731.625,
  "x": 699.3125,
  "score": 0.83575332164764
},
"left_ankle": {
  "y": 877.03125,
  "x": 725.75,
  "score": 0.85220056772232
},
"left_mouth_corner": {
  "y": 308.625,
  "x": 633.21875,
  "score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 586.21875,
  "x": 593.5625,
  "score": 0.79803824424744
},
"left_wrist": {
  "y": 374.71875,
  "x": 884.375,
  "score": 0.89635348320007
},
}
```



```
    },
    "left_ear": {
      "y": 295.40625,
      "x": 659.65625,
      "score": 0.86607384681702
    },
    },
    "left_elbow": {
      "y": 361.5,
      "x": 791.84375,
      "score": 0.83910942077637
    },
    },
    "right_shoulder": {
      "y": 348.28125,
      "x": 553.90625,
      "score": 0.85635334253311
    },
    },
    "right_ankle": {
      "y": 890.25,
      "x": 580.34375,
      "score": 0.85149073600769
    },
    },
    "right_knee": {
      "y": 744.84375,
      "x": 580.34375,
      "score": 0.83749794960022
    },
    },
    "right_wrist": {
      "y": 348.28125,
      "x": 368.84375,
      "score": 0.83893859386444
    }
  },
  },
  "location": {
    "height": 703.20654296875,
    "width": 652.61810302734,
    "top": 221.92272949219,
    "score": 0.99269664287567,
    "left": 294.03039550781
  }
},
],
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，**识别人体的静态属性和行为**，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、抽烟、使用手机、身体朝向等。

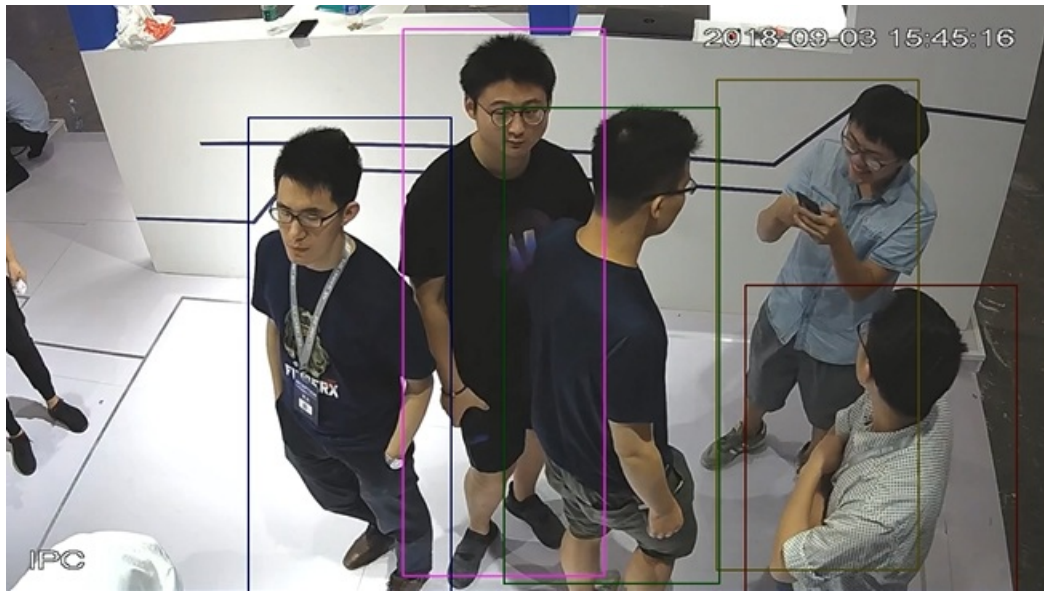
主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。

暂不适用夜间红外监控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：



非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰细分类	upper_wear_fg	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定

	属性	SK	
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚脐眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```

Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用人体检测与属性识别
result = client.body_attr(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["type"] = "gender";

// 带参数调用人体检测与属性识别
result = client.body_attr(image, options);

```

人体检测与属性识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
type	否	std::string	<p>1) 可选值说明：</p> <ul style="list-style-type: none"> <li>gender-性别，</li> <li>age-年龄阶段，</li> <li>lower_wear-下身服饰，</li> <li>upper_wear-上身服饰，</li> <li>headwear-是否戴帽子，</li> <li>face_mask-是否戴口罩，</li> <li>glasses-是否戴眼镜，</li> <li>upper_color-上身服饰颜色，</li> <li>lower_color-下身服饰颜色，</li> <li>cellphone-是否使用手机，</li> <li>upper_wear_fg-上身服饰细分类，</li> <li>upper_wear_texture-上身服饰纹理，</li> <li>orientation-身体朝向，</li> <li>umbrella-是否撑伞；</li> <li>bag-背包，</li> <li>smoke-是否吸烟，</li> <li>vehicle-交通工具，</li> <li>carrying_item-是否有手提物，</li> <li>upper_cut-上方截断，</li> <li>lower_cut-下方截断，</li> <li>occlusion-遮挡，</li> <li>is_human-是否正常人体</li> </ul> <p>2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部22个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数，取值0-1，越接近1，代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段
+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数

++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定（背面或者头部被截断的人体，一般为不确定）
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定
+++score	否	float	对应概率分数
++smoke	不	object	是否吸烟

+++score	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
+++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断，有上方截断，不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断，有下方截断，不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡，轻度遮挡，重度遮挡，不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体，非正常人体，不确定
+++score	否	float	对应概率分数

### 人体检测与属性识别 返回示例

```

{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        },
        "face_mask": {
          "score": 0.973924720287323,
          "name": "无口罩"
        }
      }
    }
  ]
}

```

```
name": "无口罩",
},
"gender": {
  "score": 0.8240132331848145,
  "name": "女性"
},
"age": {
  "score": 0.9495382308959961,
  "name": "青年"
},
"upper_cut": {
  "score": 0.9999241828918457,
  "name": "无上方截断"
},
"glasses": {
  "score": 0.9292197823524475,
  "name": "无眼镜"
},
"lower_color": {
  "score": 0.3373721539974213,
  "name": "黄"
},
"bag": {
  "score": 0.9770514369010925,
  "name": "无背包"
},
"upper_wear_texture": {
  "score": 0.7113875150680542,
  "name": "纯色"
},
"smoke": {
  "score": 0.9982085227966309,
  "name": "未吸烟"
},
"vehicle": {
  "score": 0.9907247424125671,
  "name": "无交通工具"
},
"lower_wear": {
  "score": 0.9739166498184204,
  "name": "短裤"
},
"carrying_item": {
  "score": 0.9735746383666992,
  "name": "无手提物"
},
"upper_wear": {
  "score": 0.7128473520278931,
  "name": "长袖"
},
"occlusion": {
  "score": 0.999269425868988,
  "name": "无遮挡"
},
"upper_color": {
  "score": 0.4870010614395142,
  "name": "白"
},
"is_human": {
  "score": 0.4870010614395142,
  "name": "正常人体"
}
},
```

```
"location": {  
  "height": 369,  
  "width": 316,  
  "top": 1,  
  "score": 0.9756121635437012,  
  "left": 58  
}  
}},  
"log_id": 6482262517986539034  
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。

适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。



```

Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用人流量统计
result = client.body_num(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["area"] = "x1,y1,x2,y2,x3,y3...xn,yn";
options["show"] = "false";

// 带参数调用人流量统计
result = client.body_num(image, options);

```

### 人流量统计 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
area	否	std::string	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p><b>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</b></p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p><b>area参数设置格式：</b></p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;..</p>
show	否	std::string	是否输出渲染的图片，默认不返回， <b>选true时返回渲染后的图片(base64)</b> ，其它无效值或为空则默认false

### 人流量统计 返回数据参数详情

字段	是否必选	类型	
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

### 人流量统计 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

## 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；

支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。

图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

注：

- 1) 上述24类以外的其他手势会划分到other类。
- 2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1（原食指）	One	
2	数字5（原掌心向前）	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	

7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
16	数字2	two	
17	数字3	three	

18	数字4	four	
19	数字6	six	
20	数字7	seven	
21	数字8	eight	
22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

```
Json::Value result;
```

```
std::string image;
```

```
aip::get_file_content("/assets/sample.jpg", &image);
```

```
// 调用手势识别
```

```
result = client.gesture(image, aip::null);
```

#### 手势识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

#### 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标，手势、人脸
+classname	否	string	目标所属类别，24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id，用于问题定位

### 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [
    {
      "probability": 0.9844077229499817,
      "top": 20,
      "height": 156,
      "classname": "Face",
      "width": 116,
      "left": 173
    },
    {
      "probability": 0.4679304957389832,
      "top": 157,
      "height": 106,
      "classname": "Heart_2",
      "width": 177,
      "left": 183
    }
  ]
}
```

### 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```

Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用人像分割
result = client.body_seg(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["type"] = "labelmap";

// 带参数调用人像分割
result = client.body_seg(image, options);

```

#### 人像分割 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
type	否	std::string	可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费 1) 可选值说明： labelmap - 二值图像，需二次处理方能查看分割效果 scoremap - 人像前景灰度图 foreground - 人像前景抠图，透明背景 2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部3类结果图

#### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id, 用于问题定位
labelmap	否	string	分割结果图片, base64编码之后的二值图像, 需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap, 归一到0-255, 不用进行二次处理, 直接解码保存图片即可。 Base64编码后的灰度图文件, 图片中每个像素点的灰度值 = 置信度 * 255, 置信度为原图对应像素点位于人体轮廓内的置信度, 取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图, 透明背景, Base64编码后的png格式图片, 不用进行二次处理, 直接解码保存图片即可。将置信度大于0.5的像素抠出来, 并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数, 取值0-1

### 人像分割 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg",
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ]
}
```

### 驾驶行为分析

对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。



注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：



```
Json::Value result;
```

```
std::string image;
```

```
aip::get_file_content("/assets/sample.jpg", &image);
```

```
// 调用驾驶行为分析
```

```
result = client.driver_behavior(image, aip::null);
```

```
// 如果有可选参数
```

```
std::map<std::string, std::string> options;
```

```
options["type"] = "smoke";
```

```
// 带参数调用驾驶行为分析
```

```
result = client.driver_behavior(image, options);
```

## 驾驶行为分析 请求参数详情

参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取
type	否	std::string	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

## 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
Json::Value result;
```

```
std::string image;
```

```
aip::get_file_content("/assets/sample.jpg", &image);
```

```
std::string dynamic = "true";
```

```
// 调用客流量统计-动态版
```

```
result = client.body_tracking(image, dynamic, aip::null);
```

```
// 如果有可选参数
```

```
std::map<std::string, std::string> options;
```

```
options["case_id"] = "123";
```

```
options["case_init"] = "123";
```

```
options["show"] = "true";
```

```
options["area"] = "100";
```

```
// 带参数调用客流量统计-动态版
```

```
result = client.body_tracking(image, dynamic, options);
```

客流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	std::string			图片数据的二进制字符串，可以使用aip::get_file_content函数获取
dynamic	是	std::string	true false		true：动态人流量统计，返回总人数、跟踪ID、区域进出人数； false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	std::string			任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	std::string	true false		每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
show	否	std::string	true false	false	否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	std::string			静态人数统计时，只统计区域内的人，缺省时为全图统计。 动态人流量统计时，进出区域的人流会被统计。 逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。 服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。

### area参数设置说明

进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

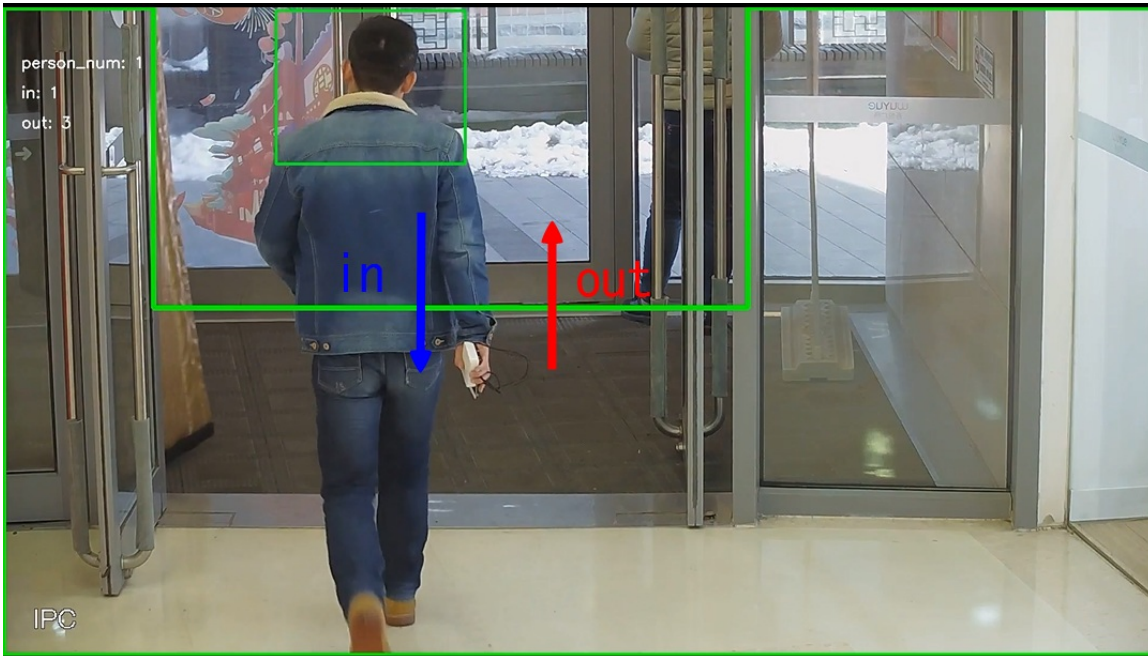
**示例1：**如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。





同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面帧的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面帧的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：



```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 手部关键点识别

### 🔗 手部关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

```
Json::Value result;

std::string image;
aip::get_file_content("/assets/sample.jpg", &image);

// 调用手部关键点识别
result = client.hand_analysis(image, aip::null);
```

## 手部关键点识别 请求参数详情

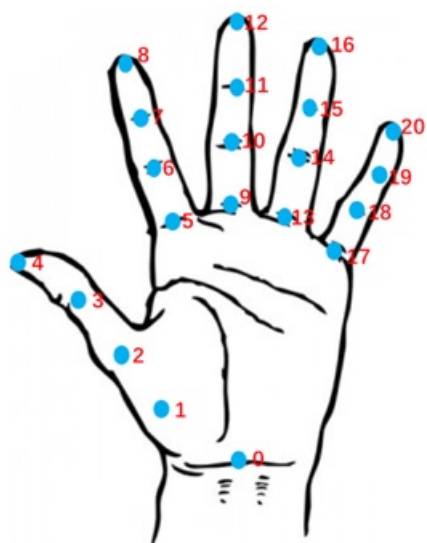
参数名称	是否必选	类型	说明
image	是	std::string	图片数据的二进制字符串，可以使用aip::get_file_content函数获取

## 手部关键点识别 返回数据参数详情

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
+++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

## 手部关键点识别 返回示例

返回关键点对应位置示意图：



```
{
  "log_id": "46532731~~~~32348525916",
  "hand_num": 1
```

```
hand_num": 1,
"hand_info": [
  {
    "hand_parts": {
      "0": {
        "y": 707,
        "x": 829,
        "score": 0.81601244211197
      },
      "1": {
        "y": 620,
        "x": 873,
        "score": 0.6850221157074
      },
      "2": {
        "y": 533,
        "x": 916,
        "score": 0.75532239675522
      },
      "3": {
        "y": 467,
        "x": 938,
        "score": 0.79282909631729
      },
      "4": {
        "y": 380,
        "x": 993,
        "score": 0.88114696741104
      },
      "5": {
        "y": 424,
        "x": 829,
        "score": 0.90723776817322
      },
      "6": {
        "y": 358,
        "x": 862,
        "score": 0.62833058834076
      },
      "7": {
        "y": 391,
        "x": 916,
        "score": 0.77525424957275
      },
      "8": {
        "y": 456,
        "x": 927,
        "score": 0.87964057922363
      },
      "9": {
        "y": 402,
        "x": 775,
        "score": 0.91389346122742
      },
      "10": {
        "y": 325,
        "x": 786,
        "score": 0.85947573184967
      },
      "11": {
        "y": 249,
        "x": 818,
        "score": 0.88918441534042
      }
    }
  }
]
```

```
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
  "y": 325,
  "x": 567,
  "score": 0.91110396385193
}
},~~~~
"location": {
  "height": 556,
  "width": 426,
  "top": 151,
  "score": 17.495880126953,
  "left": 567
}
]
}
```

错误码

[错误返回格式](#)

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。

#### 🔗 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token获取失败，可以参考 <a href="#">Access Token获取</a> 重新获取

100	invalid parameter	无效的access_token参数，token已经失效，可以参考 <a href="#">ACCESS_TOKEN失效 系列文档</a>
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## C# SDK文档

### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对C#开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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AI社区交流：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每

天流量超限额)，未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额

- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格](#)文档

## 🔗 接口功能说明

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄段、衣着（含类别/颜色）、是否戴帽子、是否戴眼镜、是否背包、是否使用手机、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘等行为，可用于分析预警危险驾驶行为
人流量统计（动态版）	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。

## 🔗 版本更新记录

上线日期	版本号	更新内容
2019.2.22	3.6.9	新增驾驶行为分析接口
2019.2.1	3.6.8	新增动态人流量统计接口
2018.9.29	3.6.4	新增手势识别、人像分割接口
2018.6.1	1.0.0	新增人体分析服务

## 快速入门

### 🔗 安装人体分析 C# SDK

C# SDK 现已开源! <https://github.com/Baidu-AIP/dotnet-sdk>

支持平台：.Net Framework 3.5 4.0 4.5，.Net Core 2.0

方法一：使用Nuget管理依赖（推荐）

在NuGet中搜索 Baidu.AI，安装最新版即可。

packet地址 <https://www.nuget.org/packages/Baidu.AI/>

方法二：下载安装

## 人体分析 C# SDK目录结构

```
Baidu.Aip
├── net35
│   ├── AipSdk.dll      // 百度AI服务 windows 动态库
│   ├── AipSdk.xml     // 注释文件
│   └── Newtonsoft.Json.dll // 第三方依赖
├── net40
├── net45
└── netstandard2.0
    ├── AipSdk.deps.json
    └── AipSdk.dll
```

如果需要在 Unity 平台使用，可引用工程源码自行编译。

### 安装

- 1.在[官方网站](#)下载C# SDK压缩工具包。
- 2.解压后，将 `AipSdk.dll` 和 `Newtonsoft.Json.dll` 中添加为引用。

### 新建交互类

`Baidu.Aip.BodyAnalysis.Body`是人体分析的交互类，为使用人体分析的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个交互类：

```
// 设置APPID/AK/SK
var APP_ID = "你的 App ID";
var API_KEY = "你的 Api Key";
var SECRET_KEY = "你的 Secret Key";

var client = new Baidu.Aip.BodyAnalysis.Body(API_KEY, SECRET_KEY);
client.Timeout = 60000; // 修改超时时间
```

在上面代码中，常量`APP_ID`在百度云控制台中创建，常量`API_KEY`与`SECRET_KEY`是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的[应用列表](#)中查看。

**注意：**如您以前是百度云的老用户，其中`API_KEY`对应百度的“Access Key ID”，`SECRET_KEY`对应百度的“Access Key Secret”。

### 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髋部、右髋部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：





多人场景：



```
public void BodyAnalysisDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    // 调用人体关键点识别，可能会抛出网络等异常，请使用try/catch捕获
    var result = client.BodyAnalysis(image);
    Console.WriteLine(result);
}
```

#### 人体关键点识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

#### 人体关键点识别 返回数据参数详情

接口除了返回人体框和每个关键点的坐标信息外，还会输出人体框和关键点的概率分数，实际应用中可以基于概率分数进行过滤，排除掉分数低的误识别“无效人体”，推荐的过滤方案：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.03时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息，包含21个关键点
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标

+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标

name	是否	数据类型	说明
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id, 用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，**如果part被截断，则x、y坐标为part被截断的图片边界位置**，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体**。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

#### 人体关键点识别 返回示例

```
{
  "person_num": 2,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        },
        "top_head": {
          "y": 242.53125,
          "x": 620,
          "score": 0.87757384777069
        },
        "right_mouth_corner": {
          "y": 308.625,
          "x": 666.76125
        }
      }
    }
  ]
}
```

```
"x": 606.78125,
"score": 0.90121293067932
},
"neck": {
  "y": 335.0625,
  "x": 620,
  "score": 0.84662038087845
},
"left_shoulder": {
  "y": 361.5,
  "x": 699.3125,
  "score": 0.83550786972046
},
"left_knee": {
  "y": 731.625,
  "x": 699.3125,
  "score": 0.83575332164764
},
"left_ankle": {
  "y": 877.03125,
  "x": 725.75,
  "score": 0.85220056772232
},
"left_mouth_corner": {
  "y": 308.625,
  "x": 633.21875,
  "score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 586.21875,
  "x": 593.5625,
  "score": 0.79803824424744
},
"left_wrist": {
  "y": 374.71875,
  "x": 884.375,
  "score": 0.89635348320007
},
}
```

```
    },
    "left_ear": {
      "y": 295.40625,
      "x": 659.65625,
      "score": 0.86607384681702
    },
    "left_elbow": {
      "y": 361.5,
      "x": 791.84375,
      "score": 0.83910942077637
    },
    "right_shoulder": {
      "y": 348.28125,
      "x": 553.90625,
      "score": 0.85635334253311
    },
    "right_ankle": {
      "y": 890.25,
      "x": 580.34375,
      "score": 0.85149073600769
    },
    "right_knee": {
      "y": 744.84375,
      "x": 580.34375,
      "score": 0.83749794960022
    },
    "right_wrist": {
      "y": 348.28125,
      "x": 368.84375,
      "score": 0.83893859386444
    }
  },
  "location": {
    "height": 703.20654296875,
    "width": 652.61810302734,
    "top": 221.92272949219,
    "score": 0.99269664287567,
    "left": 294.03039550781
  }
},
{
  "body_parts": {
    "left_hip": {
      "y": 576,
      "x": 1239.5625,
      "score": 0.84608125686646
    },
    "top_head": {
      "y": 261.15625,
      "x": 1176.59375,
      "score": 0.871442258358
    },
    "right_mouth_corner": {
      "y": 336.71875,
      "x": 1164,
      "score": 0.90951544046402
    },
    "neck": {
      "y": 361.90625,
      "x": 1176.59375,
      "score": 0.85904294252396
    },
    "left_shoulder": {
```



```
"y": 361.90625,
"x": 1239.5625,
"score": 0.8512310385704
},
"left_knee": {
  "y": 714.53125,
  "x": 1277.34375,
  "score": 0.82312393188477
},
"left_ankle": {
  "y": 853.0625,
  "x": 1315.125,
  "score": 0.83786374330521
},
"left_mouth_corner": {
  "y": 336.71875,
  "x": 1189.1875,
  "score": 0.90610301494598
},
"right_elbow": {
  "y": 387.09375,
  "x": 1025.46875,
  "score": 0.88956367969513
},
"right_ear": {
  "y": 311.53125,
  "x": 1138.8125,
  "score": 0.86518502235413
},
"nose": {
  "y": 324.125,
  "x": 1176.59375,
  "score": 0.9168484210968
},
"left_eye": {
  "y": 311.53125,
  "x": 1189.1875,
  "score": 0.91715461015701
},
"right_eye": {
  "y": 311.53125,
  "x": 1164,
  "score": 0.90343600511551
},
"right_hip": {
  "y": 576,
  "x": 1164,
  "score": 0.81976848840714
},
"left_wrist": {
  "y": 298.9375,
  "x": 1378.09375,
  "score": 0.86095398664474
},
"left_ear": {
  "y": 311.53125,
  "x": 1201.78125,
  "score": 0.86899447441101
},
"left_elbow": {
  "y": 324.125,
  "x": 1315.125,
```

```
"score": 0.89198768138885
},
"right_shoulder": {
  "y": 387.09375,
  "x": 1101.03125,
  "score": 0.85161662101746
},
"right_ankle": {
  "y": 878.25,
  "x": 1151.40625,
  "score": 0.83667933940887
},
"right_knee": {
  "y": 727.125,
  "x": 1151.40625,
  "score": 0.85485708713531
},
"right_wrist": {
  "y": 387.09375,
  "x": 949.90625,
  "score": 0.83042001724243
}
},
"location": {
  "height": 670.80139160156,
  "width": 524.25476074219,
  "top": 241.42504882812,
  "score": 0.98725789785385,
  "left": 902.15216064453
}
}
},
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、抽烟、使用手机、身体朝向等。

主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

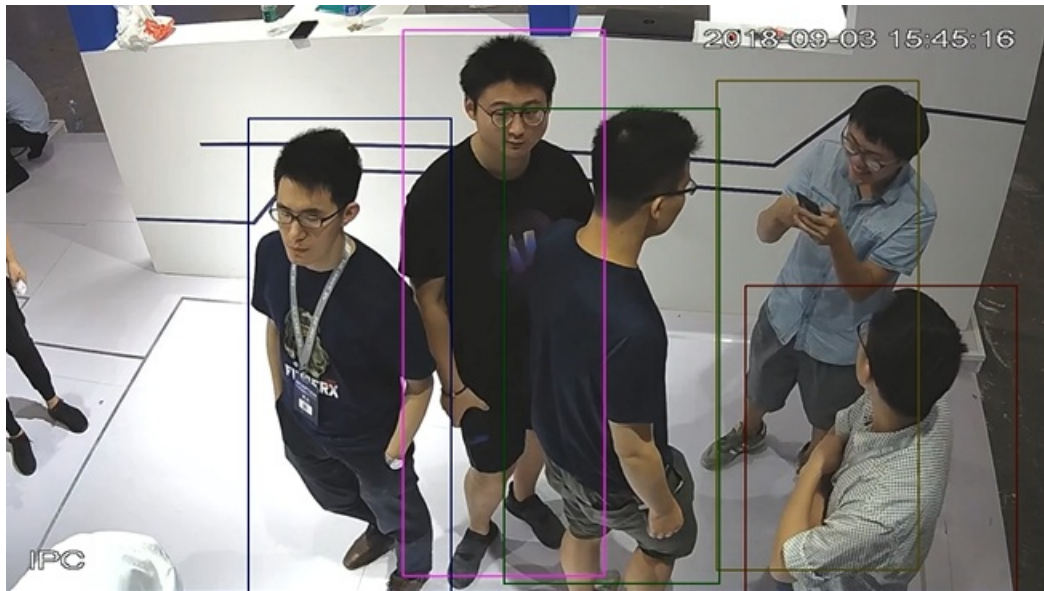
摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。

暂不适用夜间红外监控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：





非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰细分类	upper_wear_fg	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定

	汉字	SK	
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚脐眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```
public void BodyAttrDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    // 调用人体检测与属性识别，可能会抛出网络等异常，请使用try/catch捕获
    var result = client.BodyAttr(image);
    Console.WriteLine(result);
    // 如果有可选参数
    var options = new Dictionary<string, object>{
        {"type", "gender"}
    };
    // 带参数调用人体检测与属性识别
    result = client.BodyAttr(image, options);
    Console.WriteLine(result);
}
```

人体检测与属性识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据
type	否	string	<p>1) 可选值说明：</p> <p>gender-性别， age-年龄阶段， lower_wear-下身服饰， upper_wear-上身服饰， headwear-是否戴帽子， face_mask-是否戴口罩， glasses-是否戴眼镜， upper_color-上身服饰颜色， lower_color-下身服饰颜色， cellphone-是否使用手机， upper_wear_fg-上身服饰细分类， upper_wear_texture-上身服饰纹理， orientation-身体朝向， umbrella-是否撑伞； bag-背包， smoke-是否吸烟， vehicle-交通工具， carrying_item-是否有手提物， upper_cut-上方截断， lower_cut-下方截断， occlusion-遮挡， is_human-是否正常人体</p> <p>2) type 参数值可以是可选值的组合，用逗号分隔；如果无此参数默认输出全部21个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数，取值0-1，越接近1，代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段
+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数

++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定（背面或者头部被截断的人体，一般为不确定）
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定
+++score	否	float	对应概率分数
++smoke	不	object	是否吸烟

++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断，有上方截断，不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断，有下方截断，不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡，轻度遮挡，重度遮挡，不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体，非正常人体，不确定
+++score	否	float	对应概率分数

#### 人体检测与属性识别 返回示例

```

{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        },
        "face_mask": {
          "score": 0.973924720287323,
          "name": "无口罩"
        }
      }
    }
  ]
}

```

```
name": "无口罩",
},
"gender": {
  "score": 0.8240132331848145,
  "name": "女性"
},
"age": {
  "score": 0.9495382308959961,
  "name": "青年"
},
"upper_cut": {
  "score": 0.9999241828918457,
  "name": "无上方截断"
},
"glasses": {
  "score": 0.9292197823524475,
  "name": "无眼镜"
},
"lower_color": {
  "score": 0.3373721539974213,
  "name": "黄"
},
"bag": {
  "score": 0.9770514369010925,
  "name": "无背包"
},
"upper_wear_texture": {
  "score": 0.7113875150680542,
  "name": "纯色"
},
"smoke": {
  "score": 0.9982085227966309,
  "name": "未吸烟"
},
"vehicle": {
  "score": 0.9907247424125671,
  "name": "无交通工具"
},
"lower_wear": {
  "score": 0.9739166498184204,
  "name": "短裤"
},
"carrying_item": {
  "score": 0.9735746383666992,
  "name": "无手提物"
},
"upper_wear": {
  "score": 0.7128473520278931,
  "name": "长袖"
},
"occlusion": {
  "score": 0.999269425868988,
  "name": "无遮挡"
},
"upper_color": {
  "score": 0.4870010614395142,
  "name": "白"
},
"is_human": {
  "score": 0.4870010614395142,
  "name": "正常人体"
}
},
```

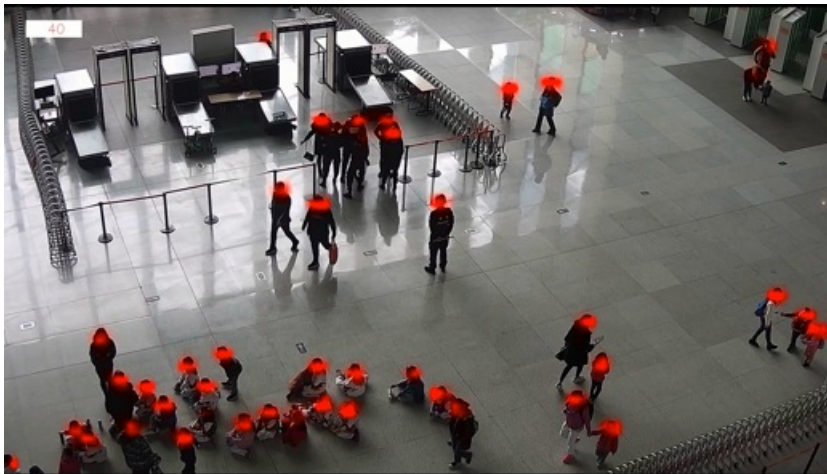
```
"location": {  
  "height": 369,  
  "width": 316,  
  "top": 1,  
  "score": 0.9756121635437012,  
  "left": 58  
}  
},  
"log_id": 6482262517986539034  
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

```
public void BodyNumDemo() {  
  var image = File.ReadAllBytes("图片文件路径");  
  // 调用人流量统计，可能会抛出网络等异常，请使用try/catch捕获  
  var result = client.BodyNum(image);  
  Console.WriteLine(result);  
  // 如果有可选参数  
  var options = new Dictionary<string, object>{  
    {"area", "0,0,100,100,200,200"},  
    {"show", "false"}  
  };  
  // 带参数调用人流量统计  
  result = client.BodyNum(image, options);  
  Console.WriteLine(result);  
}
```

人流量统计 请求参数详情

参数	是否必选	类型	可选值范围	说明
image	是	string	-	图像数据，base64编码后进行urlencode，要求base64编码和urlencode后大小不超过4M。图片的base64编码是不包含图片头的，如（data:image/jpg;base64,），支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px
area	否	string	-	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p>area参数设置格式：</p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：</p> <p>xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;...</p>
show	否	string	true,false	是否输出渲染的图片，默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false

#### 人流量统计 返回数据参数详情

字段	是否必选	类型	
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

#### 人流量统计 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

#### 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。

图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

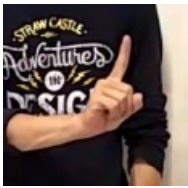


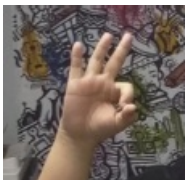
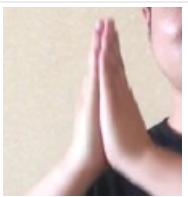

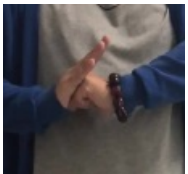
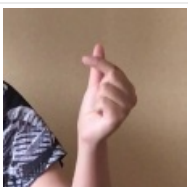



注：

1) 上述24类以外的其他手势会划分到other类。






2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下，自拍场景、他人拍摄均适用。

序号	手势名称	classname	示例图
1	数字1 (原食指)	One	
2	数字5 (原掌心向前)	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
...	...	...	

11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
16	数字2	two	
17	数字3	three	
18	数字4	four	
19	数字6	six	
20	数字7	seven	
21	数字8	eight	
			

22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

```
public void GestureDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    // 调用手势识别,可能会抛出网络等异常,请使用try/catch捕获
    var result = client.Gesture(image);
    Console.WriteLine(result);
}
```

### 手势识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据

### 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标,手势、人脸
+classname	否	string	目标所属类别,24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id,用于问题定位

### 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [{
    "probability": 0.9844077229499817,
    "top": 20,
    "height": 156,
    "classname": "Face",
    "width": 116,
    "left": 173
  },
  {
    "probability": 0.4679304957389832,
    "top": 157,
    "height": 106,
    "classname": "Heart_2",
    "width": 177,
    "left": 183
  }
  ]
}
```

## 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```

public void BodySegDemo() {
var image = File.ReadAllBytes("图片文件路径");
// 调用人像分割, 可能会抛出网络等异常, 请使用try/catch捕获
var result = client.BodySeg(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"type", "labelmap"}
};
// 带参数调用人像分割
result = client.BodySeg(image, options);
Console.WriteLine(result);
}

```

### 人像分割 请求参数详情

参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据
type	否	string	<p>可以通过设置type参数, 自主设置返回哪些结果图, 避免造成带宽的浪费</p> <p>1) 可选值说明:</p> <p>labelmap - 二值图像, 需二次处理方能查看分割效果</p> <p>scoremap - 人像前景灰度图</p> <p>foreground - 人像前景抠图, 透明背景</p> <p>2) type 参数值可以是可选值的组合, 用逗号分隔; 若无此参数默认输出全部3类结果图</p>

### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id, 用于问题定位
labelmap	否	string	分割结果图片, base64编码之后的二值图像, 需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap, 归一到0-255, 不用进行二次处理, 直接解码保存图片即可。Base64编码后的灰度图文件, 图片中每个像素点的灰度值 = 置信度 * 255, 置信度为原图对应像素点位于人体轮廓内的置信度, 取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图, 透明背景, Base64编码后的png格式图片, 不用进行二次处理, 直接解码保存图片即可。将置信度大于0.5的像素抠出来, 并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数, 取值0-1

### 人像分割 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg"
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ]
}
```

## 驾驶行为分析

对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：





```
public void DriverBehaviorDemo() {  
    var image = File.ReadAllBytes("图片文件路径");  
    // 调用驾驶行为分析, 可能会抛出网络等异常, 请使用try/catch捕获  
    var result = client.DriverBehavior(image);  
    Console.WriteLine(result);  
    // 如果有可选参数  
    var options = new Dictionary<string, object>{  
        {"type", "smoke"}  
    };  
    // 带参数调用驾驶行为分析  
    result = client.DriverBehavior(image, options);  
    Console.WriteLine(result);  
}
```

驾驶行为分析 请求参数详情



参数名称	是否必选	类型	说明
image	是	byte[]	二进制图像数据
type	否	string	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke // 吸烟， cellphone // 打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

#### 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘

+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

#### 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
public void BodyTrackingDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    var dynamic = "true";

    // 调用人流量统计-动态版，可能会抛出网络等异常，请使用try/catch捕获
    var result = client.BodyTracking(image, dynamic);
    Console.WriteLine(result);
    // 如果有可选参数
    var options = new Dictionary<string, object>{
        {"case_id", 123},
        {"case_init", "123"},
        {"show", "true"},
        {"area", "100"}
    };
    // 带参数调用人流量统计-动态版
    result = client.BodyTracking(image, dynamic, options);
    Console.WriteLine(result);
}
```

人流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	byte[]			二进制图像数据
dynamic	是	string	true false		true : 动态人流量统计, 返回总人数、跟踪ID、区域进出人数; false : 静态人数统计, 返回总人数
case_id	dynamic为true时必填	string			任务ID (通过case_id区分不同视频流, 自拟, 不同序列间不可重复即可)
case_init	dynamic为true时必填	string	true false		每个case的初始化信号, 为true时对该case下的跟踪算法进行初始化, 为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时, 直接重新初始化
show	否	string	true false	false	否返回结果图(含统计值和跟踪框渲染), 默认不返回, 选true时返回渲染后的图片(base64), 其它无效值或为空则默认false
area	否	string			静态人数统计时, 只统计区域内的人, 缺省时为全图统计。 动态人流量统计时, 进出区域的人流会被统计。 逗号分隔, 如'x1,y1,x2,y2,x3,y3...xn,yn', 按顺序依次给出每个顶点的xy坐标(默认尾点和首点相连), 形成闭合多边形区域。 服务会做范围(顶点左边需在图像范围内)及个数校验(数组长度必须为偶数, 且大于3个顶点)。只支持单个多边形区域, 建议设置矩形框, 即4个顶点。 <b>坐标取值不能超过图像宽度和高度, 比如1280的宽度, 坐标值最小建议从1开始, 最大到1279。</b>

#### area参数设置说明

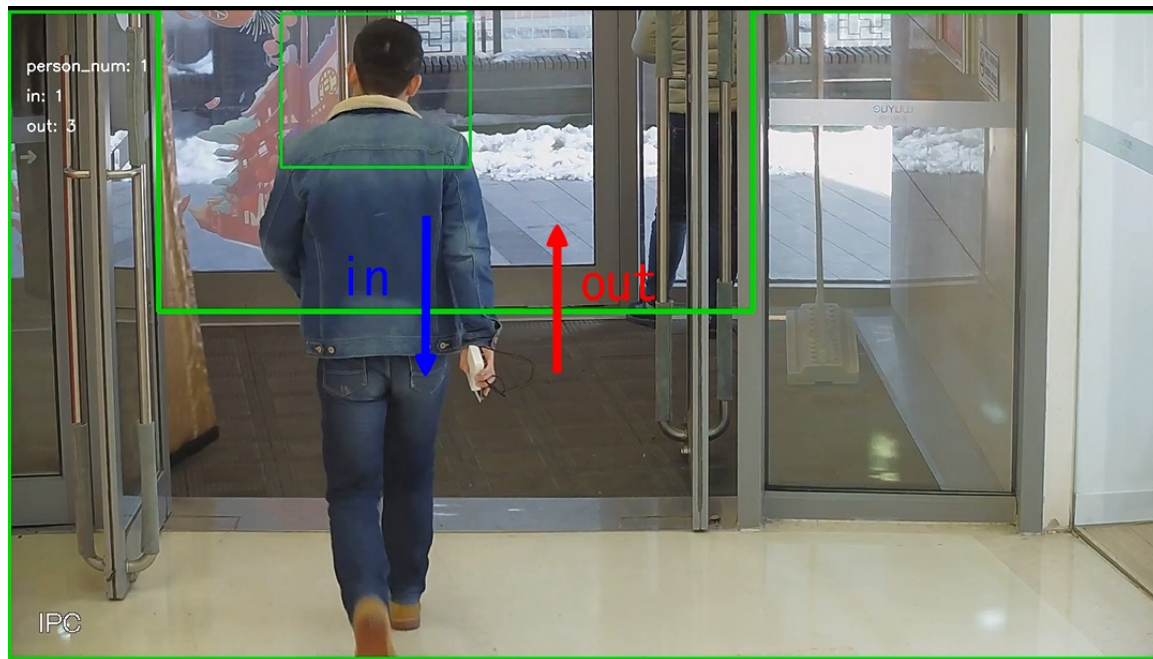
进出区域方向: 从区域外走到区域内就是in, 相反就是out, 详见下方示例。

**示例1:** 如下图, area区域框三条边贴着图像左方边缘, 从图像右方往左走到框里就是in, 从图像左方往右走出框就是out, 相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in, 就把框画在图像右半部分, 上、下、右三条边贴着图像边缘。



同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：



```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 错误码

### 错误返回格式

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。

### 错误码信息

SDK本地检测参数返回的错误码：



error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3:1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3:1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

### 服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队

216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## PHP SDK文档

### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对PHP开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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AI社区交流：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格文档](#)

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘等行为，可用于分析预警危险驾驶行为
人流量统计（动态版）	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、人机交互、AR特效等场景

## 版本更新记录

上线日期	版本号	更新内容
2019.12.19	2.2.19	新增手部关键点识别接口
2019.2.22	2.2.13	新增驾驶行为分析接口
2019.1.25	2.2.12	新增人流动态统计接口
2018.9.29	2.2.8	新增人像分割、手势识别接口
2018.6.8	2.2.5	新增人体分析服务

## 快速入门

### 安装人体分析 PHP SDK

#### 人体分析 PHP SDK目录结构

```

├── AipBodyAnalysis.php    //人体分析
├── lib
│   ├── AipHttpClient.php //内部HttpRequest类
│   ├── AipBCEUtil.php    //内部工具类
│   └── AipBase            //Aip基类

```

支持PHP版本：5.3+

使用PHP SDK开发步骤如下：

- 1.在[官方网站](#)下载php SDK压缩包。
- 2.将下载的aip-php-sdk-version.zip解压后，复制AipBodyAnalysis.php以及lib/\*到工程文件夹中。
- 3.引入AipBodyAnalysis.php

#### 新建AipBodyAnalysis

AipBodyAnalysis是人体分析的PHP SDK客户端，为使用人体分析的开发人员提供了一系列的交互方法。

参考如下代码新建一个AipBodyAnalysis：

```
require_once 'AipBodyAnalysis.php';

// 你的 APPID AK SK
const APP_ID = '你的 App ID';
const API_KEY = '你的 Api Key';
const SECRET_KEY = '你的 Secret Key';

$client = new AipBodyAnalysis(APP_ID, API_KEY, SECRET_KEY);
```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

**注意：**如您以前是百度云的老用户，其中API\_KEY对应百度云的“Access Key ID”，SECRET\_KEY对应百度云的“Access Key Secret”。

#### 配置AipBodyAnalysis

如果用户需要配置AipBodyAnalysis的网络请求参数(一般不需要配置)，可以在构造AipBodyAnalysis之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒）
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）

#### 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髌部、右髌部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：





多人场景：



```
$image = file_get_contents('example.jpg');
```

```
// 调用人体关键点识别
```

```
$client->bodyAnalysis($image);
```

人体关键点识别 请求参数详情

参数名称	是否必选	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

人体关键点识别 返回数据参数详情

接口除了返回人体框和每个关键点的坐标信息外，还会输出人体框和关键点的概率分数，实际应用中可以基于概率分数进行过滤，排除掉分数低的误识别“无效人体”，推荐的过滤方案：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.03时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息，包含21个关键点
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标

+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++v	是	float	v坐标

属性名	是否必填	数据类型	描述
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id, 用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，如果part被截断，则x、y坐标为part被截断的图片边界位置，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法**：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

人体关键点识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        },
        "top_head": {
          "y": 242.53125,
          "x": 620,
          "score": 0.87757384777069
        },
        "right_mouth_corner": {
          "y": 308.625,
          "x": 606.78125
        }
      }
    }
  ]
}
```



```
    "x": 606.78125,
    "score": 0.90121293067932
  },
  "neck": {
    "y": 335.0625,
    "x": 620,
    "score": 0.84662038087845
  },
  "left_shoulder": {
    "y": 361.5,
    "x": 699.3125,
    "score": 0.83550786972046
  },
  "left_knee": {
    "y": 731.625,
    "x": 699.3125,
    "score": 0.83575332164764
  },
  "left_ankle": {
    "y": 877.03125,
    "x": 725.75,
    "score": 0.85220056772232
  },
  "left_mouth_corner": {
    "y": 308.625,
    "x": 633.21875,
    "score": 0.91475087404251
  },
  "right_elbow": {
    "y": 348.28125,
    "x": 461.375,
    "score": 0.81766486167908
  },
  "right_ear": {
    "y": 282.1875,
    "x": 593.5625,
    "score": 0.86551451683044
  },
  "nose": {
    "y": 295.40625,
    "x": 620,
    "score": 0.90894532203674
  },
  "left_eye": {
    "y": 282.1875,
    "x": 633.21875,
    "score": 0.89628517627716
  },
  "right_eye": {
    "y": 282.1875,
    "x": 606.78125,
    "score": 0.89676940441132
  },
  "right_hip": {
    "y": 586.21875,
    "x": 593.5625,
    "score": 0.79803824424744
  },
  "left_wrist": {
    "y": 374.71875,
    "x": 884.375,
    "score": 0.89635348320007
  },
}
```

```
"left_ear": {
  "y": 295.40625,
  "x": 659.65625,
  "score": 0.86607384681702
},
"left_elbow": {
  "y": 361.5,
  "x": 791.84375,
  "score": 0.83910942077637
},
"right_shoulder": {
  "y": 348.28125,
  "x": 553.90625,
  "score": 0.85635334253311
},
"right_ankle": {
  "y": 890.25,
  "x": 580.34375,
  "score": 0.85149073600769
},
"right_knee": {
  "y": 744.84375,
  "x": 580.34375,
  "score": 0.83749794960022
},
"right_wrist": {
  "y": 348.28125,
  "x": 368.84375,
  "score": 0.83893859386444
}
},
"location": {
  "height": 703.20654296875,
  "width": 652.61810302734,
  "top": 221.92272949219,
  "score": 0.99269664287567,
  "left": 294.03039550781
}
},
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，**识别人体的静态属性和行为**，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、抽烟、使用手机、身体朝向等。

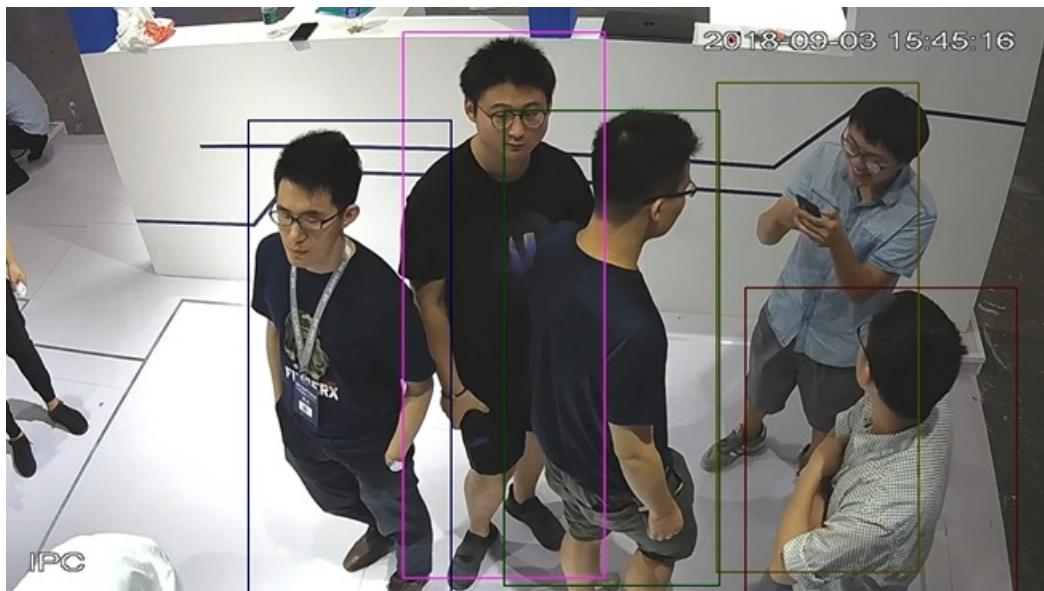
主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。

暂不适用夜间红外监控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：



非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰细分类	upper_wear_fg	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定

	属性	SK	
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚脐眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```
$image = file_get_contents('example.jpg');
```

```
// 调用人体检测与属性识别
$client->bodyAttr($image);
```

```
// 如果有可选参数
$options = array();
$options["type"] = "gender";
```

```
// 带参数调用人体检测与属性识别
$client->bodyAttr($image, $options);
```

人体检测与属性识别 请求参数详情

参数名称	是否必选	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
type	否	string	<p>1) 可选值说明:</p> <ul style="list-style-type: none"> <li>gender-性别,</li> <li>age-年龄阶段,</li> <li>lower_wear-下身服饰,</li> <li>upper_wear-上身服饰,</li> <li>headwear-是否戴帽子,</li> <li>face_mask-是否戴口罩,</li> <li>glasses-是否戴眼镜,</li> <li>upper_color-上身服饰颜色,</li> <li>lower_color-下身服饰颜色,</li> <li>cellphone-是否使用手机,</li> <li>upper_wear_fg-上身服饰细分类,</li> <li>upper_wear_texture-上身服饰纹理,</li> <li>orientation-身体朝向,</li> <li>umbrella-是否撑伞;</li> <li>bag-背包,</li> <li>smoke-是否吸烟,</li> <li>vehicle-交通工具,</li> <li>carrying_item-是否有手提物,</li> <li>upper_cut-上方截断,</li> <li>lower_cut-下方截断,</li> <li>occlusion-遮挡,</li> <li>is_human-是否正常人体</li> </ul> <p>2) type 参数值可以是可选值的组合, 用逗号分隔; 若无此参数默认输出全部22个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id, 用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数, 取值0-1, 越接近1, 代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段

+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数
++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定 (背面或者头部被截断的人体, 一般为不确定)
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定

+++score	否	float	对应概率分数
++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断, 有上方截断, 不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断, 有下方截断, 不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡, 轻度遮挡, 重度遮挡, 不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体, 非正常人体, 不确定
+++score	否	float	对应概率分数

### 人体检测与属性识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        }
      }
    }
  ]
}
```

```
"face_mask": {
  "score": 0.973924720287323,
  "name": "无口罩"
},
"gender": {
  "score": 0.8240132331848145,
  "name": "女性"
},
"age": {
  "score": 0.9495382308959961,
  "name": "青年"
},
"upper_cut": {
  "score": 0.9999241828918457,
  "name": "无上方截断"
},
"glasses": {
  "score": 0.9292197823524475,
  "name": "无眼镜"
},
"lower_color": {
  "score": 0.3373721539974213,
  "name": "黄"
},
"bag": {
  "score": 0.9770514369010925,
  "name": "无背包"
},
"upper_wear_texture": {
  "score": 0.7113875150680542,
  "name": "纯色"
},
"smoke": {
  "score": 0.9982085227966309,
  "name": "未吸烟"
},
"vehicle": {
  "score": 0.9907247424125671,
  "name": "无交通工具"
},
"lower_wear": {
  "score": 0.9739166498184204,
  "name": "短裤"
},
"carrying_item": {
  "score": 0.9735746383666992,
  "name": "无手提物"
},
"upper_wear": {
  "score": 0.7128473520278931,
  "name": "长袖"
},
"occlusion": {
  "score": 0.999269425868988,
  "name": "无遮挡"
},
"upper_color": {
  "score": 0.4870010614395142,
  "name": "白"
},
"is_human": {
  "score": 0.4870010614395142,
```



```
    "name": "正常人体"  
  }  
},  
"location": {  
  "height": 369,  
  "width": 316,  
  "top": 1,  
  "score": 0.9756121635437012,  
  "left": 58  
}  
}},  
"log_id": 6482262517986539034  
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。

适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

```

$image = file_get_contents('example.jpg');

// 调用人流量统计
$client->bodyNum($image);

// 如果有可选参数
$options = array();
$options["area"] = "x1,y1,x2,y2,x3,y3...xn,yn";
$options["show"] = "false";

// 带参数调用人流量统计
$client->bodyNum($image, $options);

```

### 人流量统计 请求参数详情

参数名称	是否可选	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
area	否	string	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p><b>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</b></p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p><b>area参数设置格式：</b></p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;..</p>
show	否	string	是否输出渲染的图片，默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false

### 人流量统计 返回数据参数详情

字段	是否可选	类型	说明
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

### 人流量统计 返回示例

```

{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}

```

## 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。

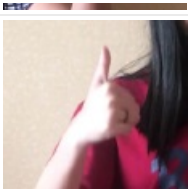

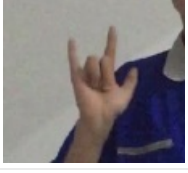
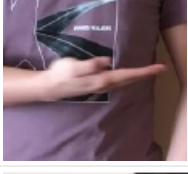


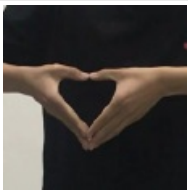



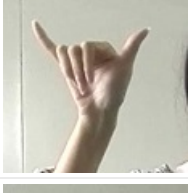
图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

注：

- 1) 上述24类以外的其他手势会划分到other类。
- 2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1（原食指）	One	
2	数字5（原掌心向前）	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
7	作别	Honour	
8	单手比心	Heart_single	

9	点赞	Thumb_up	
10	Diss	Thumb_down	
11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
16	数字2	two	
17	数字3	three	
18	数字4	four	
19	数字6	six	

20	数字7	seven	
21	数字8	eight	
22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

```
$image = file_get_contents('example.jpg');
```

```
// 调用手势识别
$client->gesture($image);
```

#### 手势识别 请求参数详情

参数名称	是否必选	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

#### 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标，手势、人脸
+classname	否	string	目标所属类别，24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id，用于问题定位

#### 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [{
    "probability": 0.9844077229499817,
    "top": 20,
    "height": 156,
    "classname": "Face",
    "width": 116,
    "left": 173
  },
  {
    "probability": 0.4679304957389832,
    "top": 157,
    "height": 106,
    "classname": "Heart_2",
    "width": 177,
    "left": 183
  }
  ]
}
```

## 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```

$image = file_get_contents('example.jpg');

// 调用人像分割
$client->bodySeg($image);

// 如果有可选参数
$options = array();
$options["type"] = "labelmap";

// 带参数调用人像分割
$client->bodySeg($image, $options);

```

### 人像分割 请求参数详情

参数名称	是否必选	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
type	否	string	可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费 1) 可选值说明： labelmap - 二值图像，需二次处理方能查看分割效果 scoremap - 人像前景灰度图 foreground - 人像前景抠图，透明背景 2) type 参数值可以是可选值的组合，用逗号分隔；如果无此参数默认输出全部3类结果图

### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id，用于问题定位
labelmap	否	string	分割结果图片，base64编码之后的二值图像，需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap，归一到0-255，不用进行二次处理，直接解码保存图片即可。 Base64编码后的灰度图文件，图片中每个像素点的灰度值 = 置信度 * 255，置信度为原图对应像素点位于人体轮廓内的置信度，取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图，透明背景，Base64编码后的png格式图片，不用进行二次处理，直接解码保存图片即可。将置信度大于0.5的像素抠出来，并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数，取值0-1

### 人像分割 返回示例



```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg"
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ],
}
```

## 驾驶行为分析

对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：



```
$image = file_get_contents('example.jpg');
```

```
// 调用驾驶行为分析
```

```
$client->driverBehavior($image);
```

```
// 如果有可选参数
```

```
$options = array();
```

```
$options["type"] = "smoke";
```

```
// 带参数调用驾驶行为分析
```

```
$client->driverBehavior($image, $options);
```

**驾驶行为分析** 请求参数详情

参数名称	是否必选	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
type	否	string	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

#### 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_	否	object	双手离开方向盘

leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
$image = file_get_contents('example.jpg');  
$dynamic = "true";  
  
// 调用人流量统计-动态版  
$client->bodyTracking($image, $dynamic);  
  
// 如果有可选参数  
$options = array();  
$options["case_id"] = 123;  
$options["case_init"] = "123";  
$options["show"] = "true";  
$options["area"] = "100";  
  
// 带参数调用人流量统计-动态版  
$client->bodyTracking($image, $dynamic, $options);
```

人流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	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
dynamic	是	string	true false		true：动态人流量统计，返回总人数、跟踪ID、区域进出人数； false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	string			任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	string	true false		每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
show	否	string	true false	false	否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	string			静态人数统计时，只统计区域内的人，缺省时为全图统计。 动态人流量统计时，进出区域的人流会被统计。 逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。 服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。

#### area参数设置说明

进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

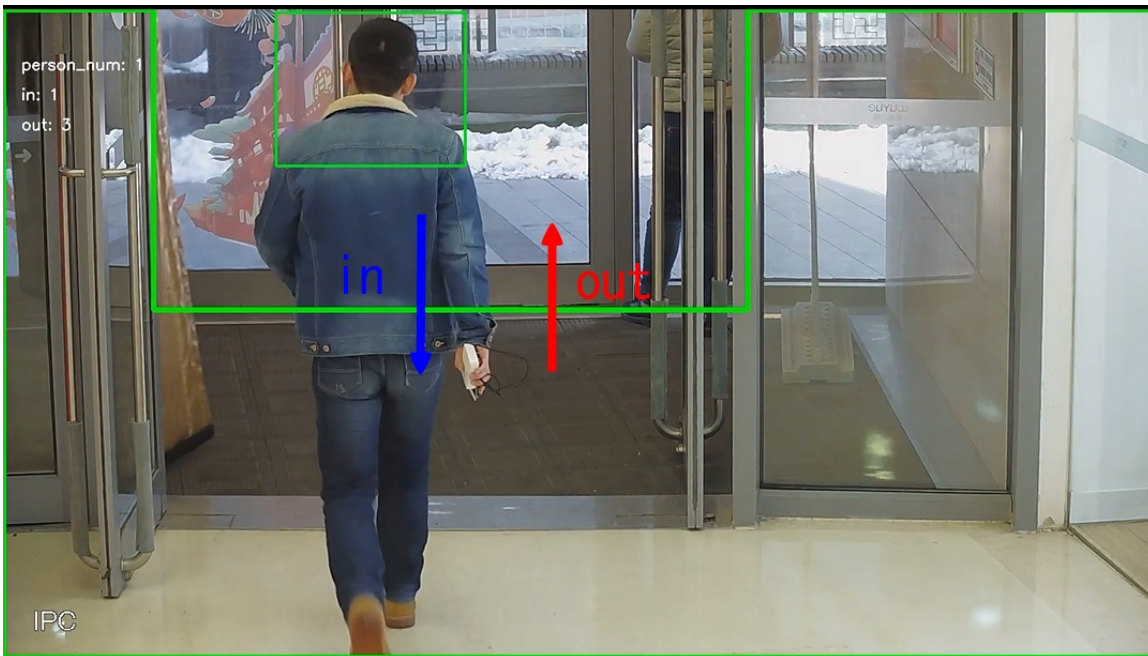
**示例1：**如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。





同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情



字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：

```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 手部关键点识别

### 🔗 手部关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

```
$image = file_get_contents('example.jpg');
```

```
// 调用手部关键点识别
$client->handAnalysis($image);
```

手部关键点识别 请求参数详情

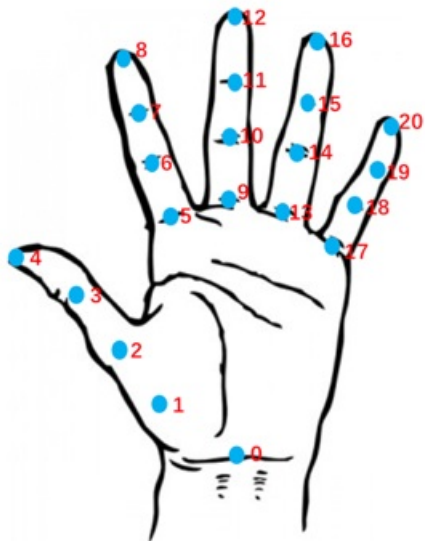
参数名称	是否必选	类型	说明
image	是	string	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，支持jpg/png/bmp格式

### 手部关键点识别 返回数据参数详情

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
+++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

### 手部关键点识别 返回示例

返回关键点对应位置示意图：



```
{
  "log_id": "46532731~~~~32348525916",
```

```
"hand_num": 1,
"hand_info": [
  {
    "hand_parts": {
      "0": {
        "y": 707,
        "x": 829,
        "score": 0.81601244211197
      },
      "1": {
        "y": 620,
        "x": 873,
        "score": 0.6850221157074
      },
      "2": {
        "y": 533,
        "x": 916,
        "score": 0.75532239675522
      },
      "3": {
        "y": 467,
        "x": 938,
        "score": 0.79282909631729
      },
      "4": {
        "y": 380,
        "x": 993,
        "score": 0.88114696741104
      },
      "5": {
        "y": 424,
        "x": 829,
        "score": 0.90723776817322
      },
      "6": {
        "y": 358,
        "x": 862,
        "score": 0.62833058834076
      },
      "7": {
        "y": 391,
        "x": 916,
        "score": 0.77525424957275
      },
      "8": {
        "y": 456,
        "x": 927,
        "score": 0.87964057922363
      },
      "9": {
        "y": 402,
        "x": 775,
        "score": 0.91389346122742
      },
      "10": {
        "y": 325,
        "x": 786,
        "score": 0.85947573184967
      },
      "11": {
        "y": 249,
        "x": 818,
```

```
"score": 0.88918441534042
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
  "y": 325,
  "x": 567,
  "score": 0.91110396385193
}
},~~~~
"location": {
  "height": 556,
  "width": 426,
  "top": 151,
  "score": 17.495880126953,
  "left": 567
}
}
]
}
```

## 错误码

[错误返回格式](#)

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。

#### 🔗 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit reached	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额

100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在
282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## Python SDK文档

### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对Python开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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AI社区交流：<http://ai.baidu.com/forum/topic/list/197>

- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格](#)文档

## 🔗 接口功能说明

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘等行为，可用于分析预警危险驾驶行为
人流量统计（动态版）	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列），进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、人机交互、AR特效等场景

## 🔗 版本更新记录

上线日期	版本号	更新内容
2019.12.19	2.2.19	新增手部关键点识别接口
2019.2.22	2.2.13	新增驾驶行为分析接口
2019.1.25	2.2.12	新增动态人流量统计接口
2018.9.29	2.2.8	新增人像分割、手势识别接口
2018.6.8	2.2.5	新增人体分析服务

## 快速入门

### 🔗 安装人体分析 Python SDK

#### 人体分析 Python SDK目录结构



```

├── README.md
├── aip          //SDK目录
│   ├── __init__.py //导出类
│   ├── base.py    //aip基类
│   ├── http.py   //http请求
│   └── bodyanalysis.py //人体分析
└── setup.py     //setuptools安装

```

支持Python版本：2.7.+ ,3.+

安装使用Python SDK有如下方式：

- 如果已安装pip，执行 `pip install baidu-aip` 即可。
- 如果已安装setuptools，执行 `python setup.py install` 即可。

#### 新建AipBodyAnalysis

AipBodyAnalysis是人体分析的Python SDK客户端，为使用人体分析的开发人员提供了一系列的交互方法。

参考如下代码新建一个AipBodyAnalysis：

```

from aip import AipBodyAnalysis

""" 你的 APPID AK SK """
APP_ID = '你的 App ID'
API_KEY = '你的 Api Key'
SECRET_KEY = '你的 Secret Key'

client = AipBodyAnalysis(APP_ID, API_KEY, SECRET_KEY)

```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

**注意：**如您以前是百度云的老用户，其中API\_KEY对应百度的“Access Key ID”，SECRET\_KEY对应百度的“Access Key Secret”。

#### 配置AipBodyAnalysis

如果用户需要配置AipBodyAnalysis的网络请求参数(一般不需要配置)，可以在构造AipBodyAnalysis之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒）
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）

#### 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髋部、右髋部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：



```
""" 读取图片 """  
def get_file_content(filePath):  
    with open(filePath, 'rb') as fp:  
        return fp.read()  
  
image = get_file_content('example.jpg')  
  
""" 调用人体关键点识别 """  
client.bodyAnalysis(image);
```

人体关键点识别 请求参数详情

参数名称	是否必选	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

### 人体关键点识别 返回数据参数详情

接口除了返回人体框和每个关键点的坐标信息外, 还会输出人体框和关键点的概率分数, 实际应用中可以基于概率分数进行过滤, 排除掉分数低的误识别“无效人体”, 推荐的过滤方案: 当关键点得分大于0.2的个数大于3, 且人体框的得分大于0.03时, 才认为是有效人体。

实际应用中, 可根据对误识别、漏识别的容忍程度, 调整阈值过滤方案, 灵活应用, 比如对误识别容忍低的应用场景, 人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息, 包含21个关键点
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标

+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部

+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id, 用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，**如果part被截断，则x、y坐标为part被截断的图片边界位置**，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体**。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

人体关键点识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        }
      }
    }
  ]
}
```



```
},
"top_head": {
  "y": 242.53125,
  "x": 620,
  "score": 0.87757384777069
},
"right_mouth_corner": {
  "y": 308.625,
  "x": 606.78125,
  "score": 0.90121293067932
},
"neck": {
  "y": 335.0625,
  "x": 620,
  "score": 0.84662038087845
},
"left_shoulder": {
  "y": 361.5,
  "x": 699.3125,
  "score": 0.83550786972046
},
"left_knee": {
  "y": 731.625,
  "x": 699.3125,
  "score": 0.83575332164764
},
"left_ankle": {
  "y": 877.03125,
  "x": 725.75,
  "score": 0.85220056772232
},
"left_mouth_corner": {
  "y": 308.625,
  "x": 633.21875,
  "score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 596.91875,
```

```
    "y": 586.21875,
    "x": 593.5625,
    "score": 0.79803824424744
  },
  "left_wrist": {
    "y": 374.71875,
    "x": 884.375,
    "score": 0.89635348320007
  },
  "left_ear": {
    "y": 295.40625,
    "x": 659.65625,
    "score": 0.86607384681702
  },
  "left_elbow": {
    "y": 361.5,
    "x": 791.84375,
    "score": 0.83910942077637
  },
  "right_shoulder": {
    "y": 348.28125,
    "x": 553.90625,
    "score": 0.85635334253311
  },
  "right_ankle": {
    "y": 890.25,
    "x": 580.34375,
    "score": 0.85149073600769
  },
  "right_knee": {
    "y": 744.84375,
    "x": 580.34375,
    "score": 0.83749794960022
  },
  "right_wrist": {
    "y": 348.28125,
    "x": 368.84375,
    "score": 0.83893859386444
  }
},
"location": {
  "height": 703.20654296875,
  "width": 652.61810302734,
  "top": 221.92272949219,
  "score": 0.99269664287567,
  "left": 294.03039550781
}
},
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、抽烟、使用手机、身体朝向等。

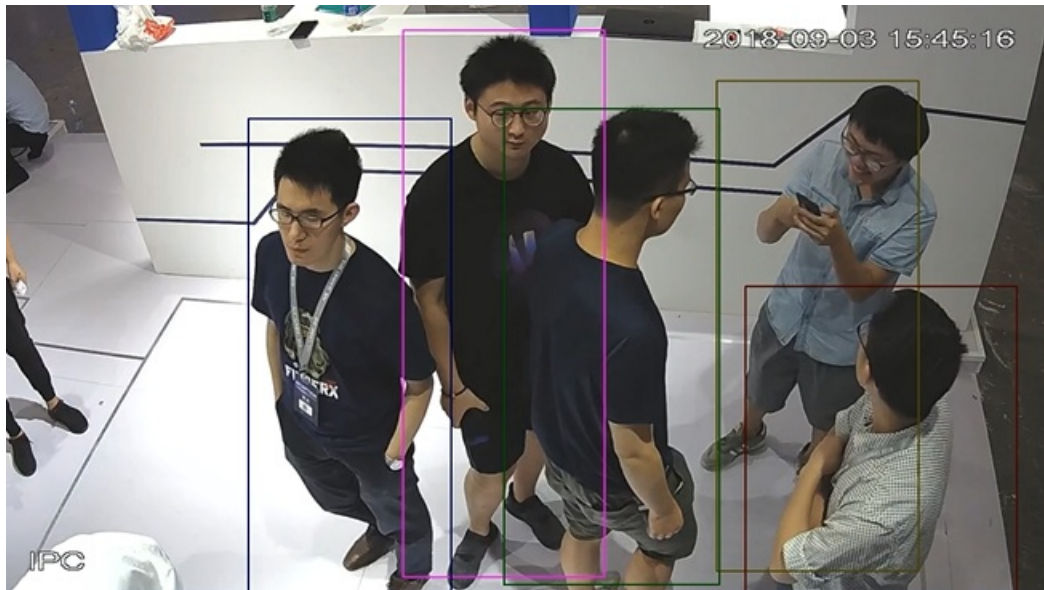
主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。

暂不适用夜间红外监控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：



非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰领口	upper_wear_neckline	圆领、V领、立领、高领、无领、开领、其他



9	饰细分类	ear_fg	l恤、九裤、衬衫、四袋、毛衣、夹克、羽绒服、风衣、外套
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚挤眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚挤眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用人体检测与属性识别 """
client.bodyAttr(image);

""" 如果有可选参数 """
options = {}
options["type"] = "gender"

""" 带参数调用人体检测与属性识别 """
client.bodyAttr(image, options)

```

人体检测与属性识别 请求参数详情

参数名称	是否必选	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
type	否	string	<p>1) 可选值说明:</p> <ul style="list-style-type: none"> <li>gender-性别,</li> <li>age-年龄阶段,</li> <li>lower_wear-下身服饰,</li> <li>upper_wear-上身服饰,</li> <li>headwear-是否戴帽子,</li> <li>face_mask-是否戴口罩,</li> <li>glasses-是否戴眼镜,</li> <li>upper_color-上身服饰颜色,</li> <li>lower_color-下身服饰颜色,</li> <li>cellphone-是否使用手机,</li> <li>upper_wear_fg-上身服饰细分类,</li> <li>upper_wear_texture-上身服饰纹理,</li> <li>orientation-身体朝向,</li> <li>umbrella-是否撑伞,</li> <li>bag-背包,</li> <li>smoke-是否吸烟,</li> <li>vehicle-交通工具,</li> <li>carrying_item-是否有手提物,</li> <li>upper_cut-上方截断,</li> <li>lower_cut-下方截断,</li> <li>occlusion-遮挡,</li> <li>is_human-是否正常人体</li> </ul> <p>2) type 参数值可以是可选值的组合, 用逗号分隔; 若无此参数默认输出全部22个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id, 用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数, 取值0-1, 越接近1, 代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段

+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数
++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定 (背面或者头部被截断的人体, 一般为不确定)
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定

+++score	否	float	对应概率分数
++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断, 有上方截断, 不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断, 有下方截断, 不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡, 轻度遮挡, 重度遮挡, 不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体, 非正常人体, 不确定
+++score	否	float	对应概率分数

### 人体检测与属性识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        }
      }
    }
  ]
}
```

```
"face_mask": {
  "score": 0.99429154396057,
  "name": "无口罩"
},
"gender": {
  "score": 0.8240132331848145,
  "name": "女性"
},
"age": {
  "score": 0.9495382308959961,
  "name": "青年"
},
"upper_cut": {
  "score": 0.9999241828918457,
  "name": "无上方截断"
},
"glasses": {
  "score": 0.9292197823524475,
  "name": "无眼镜"
},
"lower_color": {
  "score": 0.3373721539974213,
  "name": "黄"
},
"bag": {
  "score": 0.9770514369010925,
  "name": "无背包"
},
"upper_wear_texture": {
  "score": 0.7113875150680542,
  "name": "纯色"
},
"smoke": {
  "score": 0.9982085227966309,
  "name": "未吸烟"
},
"vehicle": {
  "score": 0.9907247424125671,
  "name": "无交通工具"
},
"lower_wear": {
  "score": 0.9739166498184204,
  "name": "短裤"
},
"carrying_item": {
  "score": 0.9735746383666992,
  "name": "无手提物"
},
"upper_wear": {
  "score": 0.7128473520278931,
  "name": "长袖"
},
"occlusion": {
  "score": 0.999269425868988,
  "name": "无遮挡"
},
"upper_color": {
  "score": 0.4870010614395142,
  "name": "白"
},
"is_human": {
  "score": 0.4870010614395142,
  "name": "不是人类"
```

```
    "name": "正常人体"  
  }  
},  
"location": {  
  "height": 369,  
  "width": 316,  
  "top": 1,  
  "score": 0.9756121635437012,  
  "left": 58  
}  
}},  
"log_id": 6482262517986539034  
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。

适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用人流量统计 """
client.bodyNum(image);

""" 如果有可选参数 """
options = {}
options["area"] = "x1,y1,x2,y2,x3,y3...xn,yn"
options["show"] = "false"

""" 带参数调用人流量统计 """
client.bodyNum(image, options)

```

### 人流量统计 请求参数详情

参数名称	是否必选	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
area	否	string	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p><b>area参数设置格式：</b></p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;..</p>
show	否	string	是否输出渲染的图片，默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false

### 人流量统计 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

### 人流量统计 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

## 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。

图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。


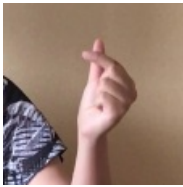


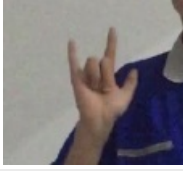
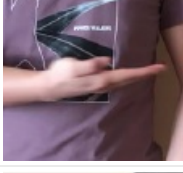


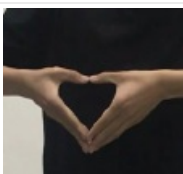


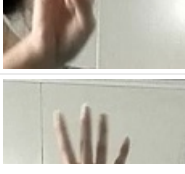
注：

- 1) 上述24类以外的其他手势会划分到other类。
- 2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1（原食指）	One	
2	数字5（原掌心向前）	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
			



7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
16	数字2	two	
17	数字3	three	
18	数字4	four	

18	数字5	five	
19	数字6	six	
20	数字7	seven	
21	数字8	eight	
22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

""" 读取图片 """

```
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()
```

```
image = get_file_content('example.jpg')
```

""" 调用手势识别 """

```
client.gesture(image);
```

#### 手势识别 请求参数详情

参数名称	是否必选	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

#### 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标，手势、人脸
+classname	否	string	目标所属类别，24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id，用于问题定位

### 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [
    {
      "probability": 0.9844077229499817,
      "top": 20,
      "height": 156,
      "classname": "Face",
      "width": 116,
      "left": 173
    },
    {
      "probability": 0.4679304957389832,
      "top": 157,
      "height": 106,
      "classname": "Heart_2",
      "width": 177,
      "left": 183
    }
  ]
}
```

### 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用人像分割 """
client.bodySeg(image);

""" 如果有可选参数 """
options = {}
options["type"] = "labelmap"

""" 带参数调用人像分割 """
client.bodySeg(image, options)

```

#### 人像分割 请求参数详情

参数名称	是否必选	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
type	否	string	可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费 1) 可选值说明： labelmap - 二值图像，需二次处理方能查看分割效果 scoremap - 人像前景灰度图 foreground - 人像前景抠图，透明背景 2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部3类结果图

#### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id，用于问题定位
labelmap	否	string	分割结果图片，base64编码之后的二值图像，需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap，归一到0-255，不用进行二次处理，直接解码保存图片即可。 Base64编码后的灰度图文件，图片中每个像素点的灰度值 = 置信度 * 255，置信度为原图对应像素点位于人体轮廓内的置信度，取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图，透明背景，Base64编码后的png格式图片，不用进行二次处理，直接解码保存图片即可。将置信度大于0.5的像素抠出来，并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数，取值0-1

### 人像分割 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg",
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ]
}
```

### 驾驶行为分析

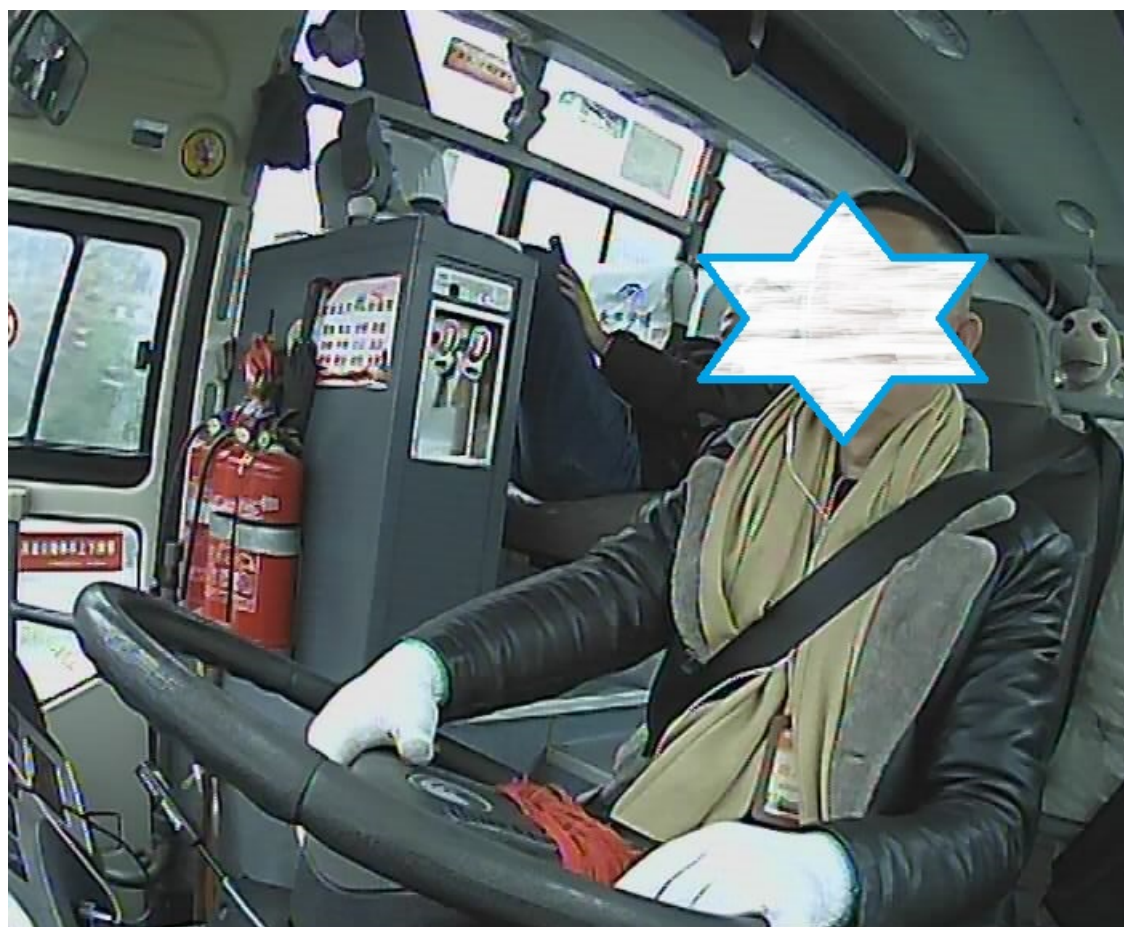
对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：





```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用驾驶行为分析 """
client.driverBehavior(image);

""" 如果有可选参数 """
options = {}
options["type"] = "smoke"

""" 带参数调用驾驶行为分析 """
client.driverBehavior(image, options)

```

### 驾驶行为分析 请求参数详情

参数名称	是否必选	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
type	否	string	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

### 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
+++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度



++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
dynamic = "true"

""" 调用人流量统计-动态版 """
client.bodyTracking(image, dynamic);

""" 如果有可选参数 """
options = {}
options["case_id"] = 123
options["case_init"] = "123"
options["show"] = "true"
options["area"] = "100"

""" 带参数调用人流量统计-动态版 """
client.bodyTracking(image, dynamic, options)
```

人流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	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
dynamic	是	string	true false		true：动态人流量统计，返回总人数、跟踪ID、区域进出人数； false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	string			任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	string	true false		每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
show	否	string	true false	false	否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	string			静态人数统计时，只统计区域内的人，缺省时为全图统计。 动态人流量统计时，进出区域的人流会被统计。 逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。 服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。

#### area参数设置说明

进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

**示例1：**如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。



同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面帧的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面帧的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：

```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 手部关键点识别

### 🔗 手部关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。



```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用手部关键点识别 """
client.handAnalysis(image);

```

### 手部关键点识别 请求参数详情

参数名称	是否必选	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50px，最长边最大4096px，支持jpg/png/bmp格式

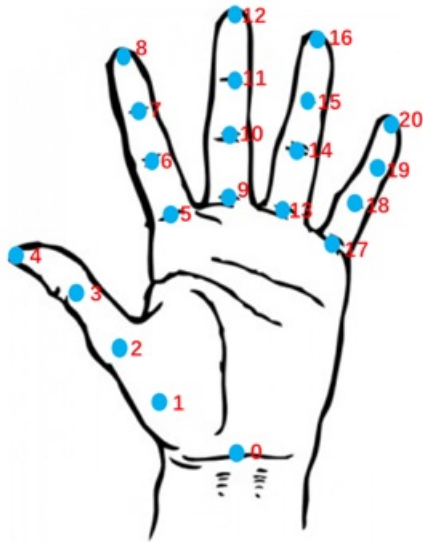
### 手部关键点识别 返回数据参数详情

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

### 手部关键点识别 返回示例

返回关键点对应位置示意图：





```
{
  "log_id": "46532731~~~~32348525916",
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        "2": {
          "y": 533,
          "x": 916,
          "score": 0.75532239675522
        },
        "3": {
          "y": 467,
          "x": 938,
          "score": 0.79282909631729
        },
        "4": {
          "y": 380,
          "x": 993,
          "score": 0.88114696741104
        },
        "5": {
          "y": 424,
          "x": 829,
          "score": 0.90723776817322
        },
        "6": {
          "y": 358,
          "x": 862,
          "score": 0.62833058834076
        },
        "7": {
          "y": 391,
          "x": 916,
```

```
"score": 0.77525424957275
},
"8": {
  "y": 456,
  "x": 927,
  "score": 0.87964057922363
},
"9": {
  "y": 402,
  "x": 775,
  "score": 0.91389346122742
},
"10": {
  "y": 325,
  "x": 786,
  "score": 0.85947573184967
},
"11": {
  "y": 249,
  "x": 818,
  "score": 0.88918441534042
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
```

```

    "z": {
      "y": 325,
      "x": 567,
      "score": 0.91110396385193
    }
  }, ~~~~
  "location": {
    "height": 556,
    "width": 426,
    "top": 151,
    "score": 17.495880126953,
    "left": 567
  }
}
]
}

```

## 错误码

### 错误返回格式

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。

### 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

### 服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交</a>

4	reached	<a href="#">工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在

282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## Node.js SDK文档

### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对Nodejs开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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[AI社区交流](#)：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格文档](#)

### 🔗 接口功能说明

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘、未佩戴口罩、闭眼、打哈欠等行为，可用于分析预警危险驾驶行为
人流量统计（动态版）	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、人机交互、AR特效等场景

## 🔗 版本更新记录

上线日期	版本号	更新内容
2019.12.19	2.4.5	新增手部关键点识别接口
2019.2.22	2.3.11	新增驾驶行为分析接口
2019.1.25	2.3.10	新增动态人流量统计接口
2018.9.29	2.3.6	新增人像分割、手势识别接口
2018.6.8	2.3.2	新增人体分析服务

## 快速入门

### 🔗 安装人体分析 Node SDK

#### 人体分析 Node SDK目录结构

```

├── src
│   ├── auth           //授权相关类
│   ├── http           //Http通信相关类
│   ├── client         //公用类
│   ├── util           //工具类
│   └── const          //常量类
├── AipBodyAnalysis.js //人体分析交互类
├── index.js           //入口文件
└── package.json      //npm包描述文件

```

#### 支持 node 版本 4.0+

#### 查看源码

Nodejs SDK代码已开源，您可以查看代码、或者在License范围内修改和编译SDK以适配您的环境。

github链接：<https://github.com/Baidu-AIP/nodejs-sdk>

#### 直接使用node开发包步骤如下：

- 1.在[官方网站](#)下载node SDK压缩包。
- 2.将下载的aip-node-sdk-version.zip解压后，复制到工程文件夹中。
- 3.进入目录，运行npm install安装sdk依赖库
- 4.把目录当做模块依赖

其中，`version`为版本号，添加完成后，用户就可以在工程中使用人体分析 Node SDK。

#### 直接使用npm安装依赖：

```
npm install baidu-aip-sdk
```

### 🔗 新建AipBodyAnalysisClient

AipBodyAnalysisClient是人体分析的node客户端，为使用人体分析的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个AipBodyAnalysisClient：

```
var AipBodyAnalysisClient = require("baidu-aip-sdk").bodyanalysis;

// 设置APPID/AK/SK
var APP_ID = "你的 App ID";
var API_KEY = "你的 Api Key";
var SECRET_KEY = "你的 Secret Key";

// 新建一个对象，建议只保存一个对象调用服务接口
var client = new AipBodyAnalysisClient(APP_ID, API_KEY, SECRET_KEY);
```

为了使开发者更灵活的控制请求，模块提供了设置全局参数和全局请求拦截器的方法；本库发送网络请求依赖的是[request模块](#)，因此参数格式与request模块的参数相同

更多参数细节您可以参考[request官方参数文档](#)。

```
var HttpClient = require("baidu-aip-sdk").HttpClient;

// 设置request库的一些参数，例如代理服务地址，超时时间等
// request参数请参考 https://github.com/request/request#requestoptions-callback
HttpClient.setRequestOptions({timeout: 5000});

// 也可以设置拦截每次请求（设置拦截后，调用的setRequestOptions设置的参数将不生效），
// 可以按需修改request参数（无论是否修改，必须返回函数调用参数）
// request参数请参考 https://github.com/request/request#requestoptions-callback
HttpClient.setRequestInterceptor(function(requestOptions) {
  // 查看参数
  console.log(requestOptions)
  // 修改参数
  requestOptions.timeout = 5000;
  // 返回参数
  return requestOptions;
});
```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

**注意：**如您以前是百度云的老用户，其中API\_KEY对应白云的“Access Key ID”，SECRET\_KEY对应白云的“Access Key Secret”。

## 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髋部、右髋部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：





多人场景：



```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用人体关键点识别
client.bodyAnalysis(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});
```

人体关键点识别 请求参数详情



参数名称	是否必选	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

### 人体关键点识别 返回数据参数详情

接口除了返回人体框和每个关键点的坐标信息外, 还会输出人体框和关键点的概率分数, 实际应用中可以基于概率分数进行过滤, 排除掉分数低的误识别“无效人体”, 推荐的过滤方案: 当关键点得分大于0.2的个数大于3, 且人体框的得分大于0.03时, 才认为是有效人体。

实际应用中, 可根据对误识别、漏识别的容忍程度, 调整阈值过滤方案, 灵活应用, 比如对误识别容忍低的应用场景, 人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息, 包含21个关键点
++top_head	是	object	头顶
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标

+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部

+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id, 用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x，y两个坐标，如果part被截断，则x、y坐标为part被截断的图片边界位置，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法**：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

人体关键点识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        }
      }
    }
  ]
}
```

```
},
"top_head": {
  "y": 242.53125,
  "x": 620,
  "score": 0.87757384777069
},
"right_mouth_corner": {
  "y": 308.625,
  "x": 606.78125,
  "score": 0.90121293067932
},
"neck": {
  "y": 335.0625,
  "x": 620,
  "score": 0.84662038087845
},
"left_shoulder": {
  "y": 361.5,
  "x": 699.3125,
  "score": 0.83550786972046
},
"left_knee": {
  "y": 731.625,
  "x": 699.3125,
  "score": 0.83575332164764
},
"left_ankle": {
  "y": 877.03125,
  "x": 725.75,
  "score": 0.85220056772232
},
"left_mouth_corner": {
  "y": 308.625,
  "x": 633.21875,
  "score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 500.04075,
```

```
    "y": 586.21875,
    "x": 593.5625,
    "score": 0.79803824424744
  },
  "left_wrist": {
    "y": 374.71875,
    "x": 884.375,
    "score": 0.89635348320007
  },
  "left_ear": {
    "y": 295.40625,
    "x": 659.65625,
    "score": 0.86607384681702
  },
  "left_elbow": {
    "y": 361.5,
    "x": 791.84375,
    "score": 0.83910942077637
  },
  "right_shoulder": {
    "y": 348.28125,
    "x": 553.90625,
    "score": 0.85635334253311
  },
  "right_ankle": {
    "y": 890.25,
    "x": 580.34375,
    "score": 0.85149073600769
  },
  "right_knee": {
    "y": 744.84375,
    "x": 580.34375,
    "score": 0.83749794960022
  },
  "right_wrist": {
    "y": 348.28125,
    "x": 368.84375,
    "score": 0.83893859386444
  }
},
"location": {
  "height": 703.20654296875,
  "width": 652.61810302734,
  "top": 221.92272949219,
  "score": 0.99269664287567,
  "left": 294.03039550781
}
},
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、抽烟、使用手机、身体朝向等。

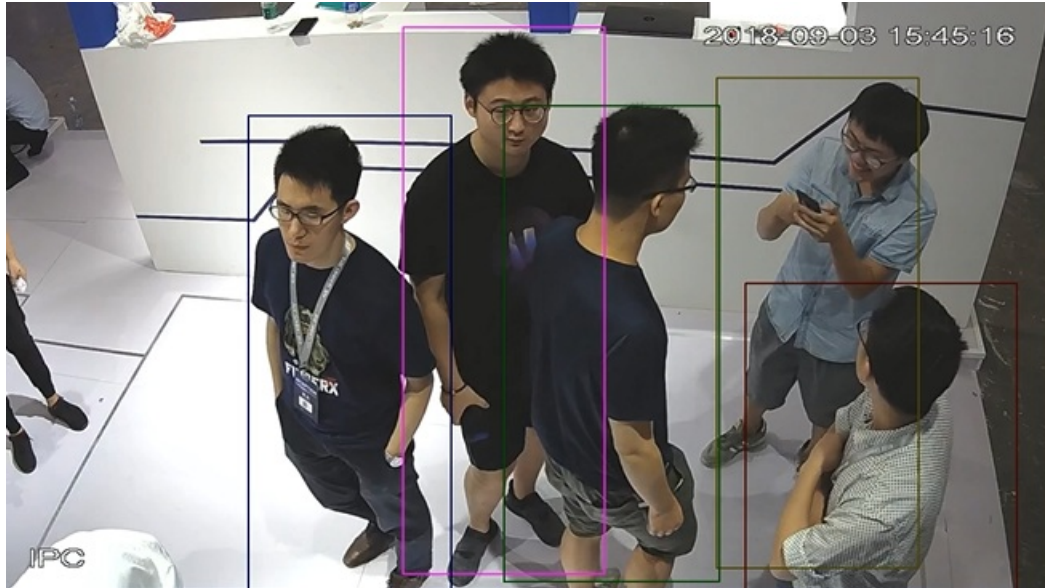
主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监

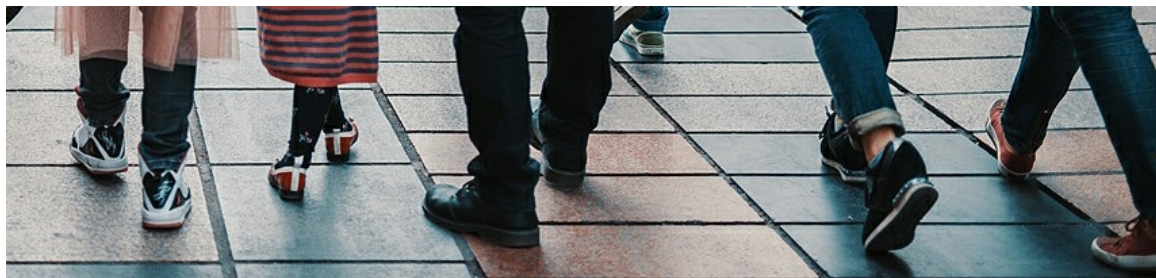
控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：



非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰细类	upper_wear	T恤 短袖 衬衫 西装 卫衣 夹克 羽绒服 风衣 外套

9	脚部刀类	ear_fg	刀、匕首、利刃、匕首、大刀、大刀、刀柄、刀柄、刀柄
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚脐眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用人体检测与属性识别
client.bodyAttr(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

// 如果有可选参数
var options = {};
options["type"] = "gender";

// 带参数调用人体检测与属性识别
client.bodyAttr(image, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});;
```

人体检测与属性识别 请求参数详情



参数名称	是否必选	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
type	否	string	<p>1) 可选值说明:</p> <ul style="list-style-type: none"> <li>gender-性别,</li> <li>age-年龄阶段,</li> <li>lower_wear-下身服饰,</li> <li>upper_wear-上身服饰,</li> <li>headwear-是否戴帽子,</li> <li>face_mask-是否戴口罩,</li> <li>glasses-是否戴眼镜,</li> <li>upper_color-上身服饰颜色,</li> <li>lower_color-下身服饰颜色,</li> <li>cellphone-是否使用手机,</li> <li>upper_wear_fg-上身服饰细分类,</li> <li>upper_wear_texture-上身服饰纹理,</li> <li>orientation-身体朝向,</li> <li>umbrella-是否撑伞;</li> <li>bag-背包,</li> <li>smoke-是否吸烟,</li> <li>vehicle-交通工具,</li> <li>carrying_item-是否有手提物,</li> <li>upper_cut-上方截断,</li> <li>lower_cut-下方截断,</li> <li>occlusion-遮挡,</li> <li>is_human-是否正常人体</li> </ul> <p>2) type 参数值可以是可选值的组合, 用逗号分隔; 如果无此参数默认输出全部22个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id, 用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数, 取值0-1, 越接近1, 代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段

+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数
++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定 (背面或者头部被截断的人体, 一般为不确定)
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定

+++score	否	float	对应概率分数
++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断, 有上方截断, 不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断, 有下方截断, 不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡, 轻度遮挡, 重度遮挡, 不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体, 非正常人体, 不确定
+++score	否	float	对应概率分数

### 人体检测与属性识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        }
      }
    }
  ]
}
```

```
"face_mask": {
  "score": 0.99429154396057,
  "name": "无口罩"
},
"gender": {
  "score": 0.8240132331848145,
  "name": "女性"
},
"age": {
  "score": 0.9495382308959961,
  "name": "青年"
},
"upper_cut": {
  "score": 0.9999241828918457,
  "name": "无上方截断"
},
"glasses": {
  "score": 0.9292197823524475,
  "name": "无眼镜"
},
"lower_color": {
  "score": 0.3373721539974213,
  "name": "黄"
},
"bag": {
  "score": 0.9770514369010925,
  "name": "无背包"
},
"upper_wear_texture": {
  "score": 0.7113875150680542,
  "name": "纯色"
},
"smoke": {
  "score": 0.9982085227966309,
  "name": "未吸烟"
},
"vehicle": {
  "score": 0.9907247424125671,
  "name": "无交通工具"
},
"lower_wear": {
  "score": 0.9739166498184204,
  "name": "短裤"
},
"carrying_item": {
  "score": 0.9735746383666992,
  "name": "无手提物"
},
"upper_wear": {
  "score": 0.7128473520278931,
  "name": "长袖"
},
"occlusion": {
  "score": 0.999269425868988,
  "name": "无遮挡"
},
"upper_color": {
  "score": 0.4870010614395142,
  "name": "白"
},
"is_human": {
  "score": 0.4870010614395142,
  "name": "不是人类"
```

```
    "name": "正常人体"
  }
},
"location": {
  "height": 369,
  "width": 316,
  "top": 1,
  "score": 0.9756121635437012,
  "left": 58
}
}},
"log_id": 6482262517986539034
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。

适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

```

var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用人流量统计
client.bodyNum(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

// 如果有可选参数
var options = {};
options["area"] = "x1,y1,x2,y2,x3,y3...xn,yn";
options["show"] = "false";

// 带参数调用人流量统计
client.bodyNum(image, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

```

#### 人流量统计 请求参数详情

参数名称	是否必选	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
area	否	string	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p><b>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</b></p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p><b>area参数设置格式：</b></p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;..</p>
show	否	string	是否输出渲染的图片，默认不返回， <b>选true时返回渲染后的图片(base64)</b> ，其它无效值或为空则默认false

#### 人流量统计 返回数据参数详情

字段	是否必选	类型	
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

### 人流量统计 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

### 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。



图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

注：

- 1) 上述24类以外的其他手势会划分到other类。
- 2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1（原食指）	One	
2	数字5（原掌心向前）	Five	
3	拳头	Fist	
4	OK	OK	

5	祈祷	Prayer	
6	作揖	Congratulation	
7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
			



16	数字2	two	
17	数字3	three	
18	数字4	four	
19	数字6	six	
20	数字7	seven	
21	数字8	eight	
22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用手势识别
client.gesture(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});
```

## 手势识别 请求参数详情

参数名称	是否必选	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

## 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标，手势、人脸
+classname	否	string	目标所属类别，24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id，用于问题定位

## 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [
    {
      "probability": 0.9844077229499817,
      "top": 20,
      "height": 156,
      "classname": "Face",
      "width": 116,
      "left": 173
    },
    {
      "probability": 0.4679304957389832,
      "top": 157,
      "height": 106,
      "classname": "Heart_2",
      "width": 177,
      "left": 183
    }
  ]
}
```

## 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用人像分割
client.bodySeg(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

// 如果有可选参数
var options = {};
options["type"] = "labelmap";

// 带参数调用人像分割
client.bodySeg(image, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});;
```

#### 人像分割 请求参数详情

参数名称	是否可选	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
type	否	string	可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费 1) 可选值说明： labelmap - 二值图像，需二次处理方能查看分割效果 scoremap - 人像前景灰度图 foreground - 人像前景抠图，透明背景 2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部3类结果图

#### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id，用于问题定位
labelmap	否	string	分割结果图片，base64编码之后的二值图像，需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap，归一到0-255，不用进行二次处理，直接解码保存图片即可。 Base64编码后的灰度图文件，图片中每个像素点的灰度值 = 置信度 * 255，置信度为原图对应像素点位于人体轮廓内的置信度，取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图，透明背景，Base64编码后的png格式图片，不用进行二次处理，直接解码保存图片即可。将置信度大于0.5的像素抠出来，并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数，取值0-1

### 人像分割 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg",
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ]
}
```

### 驾驶行为分析

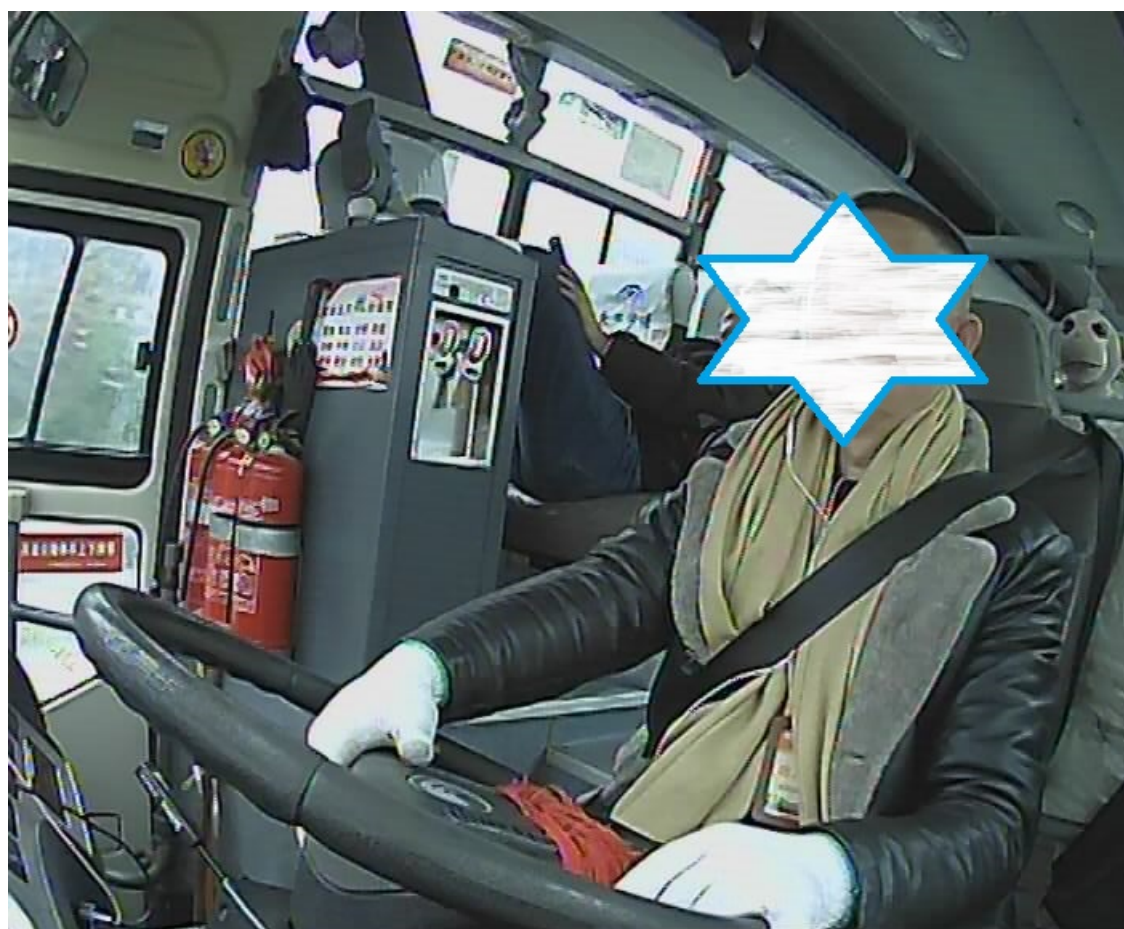
对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

#### 图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

#### 示例图参考：



```

var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用驾驶行为分析
client.driverBehavior(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

// 如果有可选参数
var options = {};
options["type"] = "smoke";

// 带参数调用驾驶行为分析
client.driverBehavior(image, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

```

#### 驾驶行为分析 请求参数详情

参数名称	是否必选	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
type	否	string	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

#### 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	检测到的属性行为信息，若检测到多个对象，则返回所有



person_info	是	object	驾驶员的属性行为信息；若不检测驾驶员，则该属性[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度
+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

返回示例



```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");
var dynamic = "true";

// 调用人流量统计-动态版
client.bodyTracking(image, dynamic).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

// 如果有可选参数
var options = {};
options["case_id"] = "123";
options["case_init"] = "123";
options["show"] = "true";
options["area"] = "100";

// 带参数调用人流量统计-动态版
client.bodyTracking(image, dynamic, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});
```

人流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	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
dynamic	是	string	true false		true：动态人流量统计，返回总人数、跟踪ID、区域进出人数； false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	string			任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	string	true false		每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
show	否	string	true false	false	否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	string			静态人数统计时，只统计区域内的人，缺省时为全图统计。 动态人流量统计时，进出区域的人流会被统计。 逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。 服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。

#### area参数设置说明

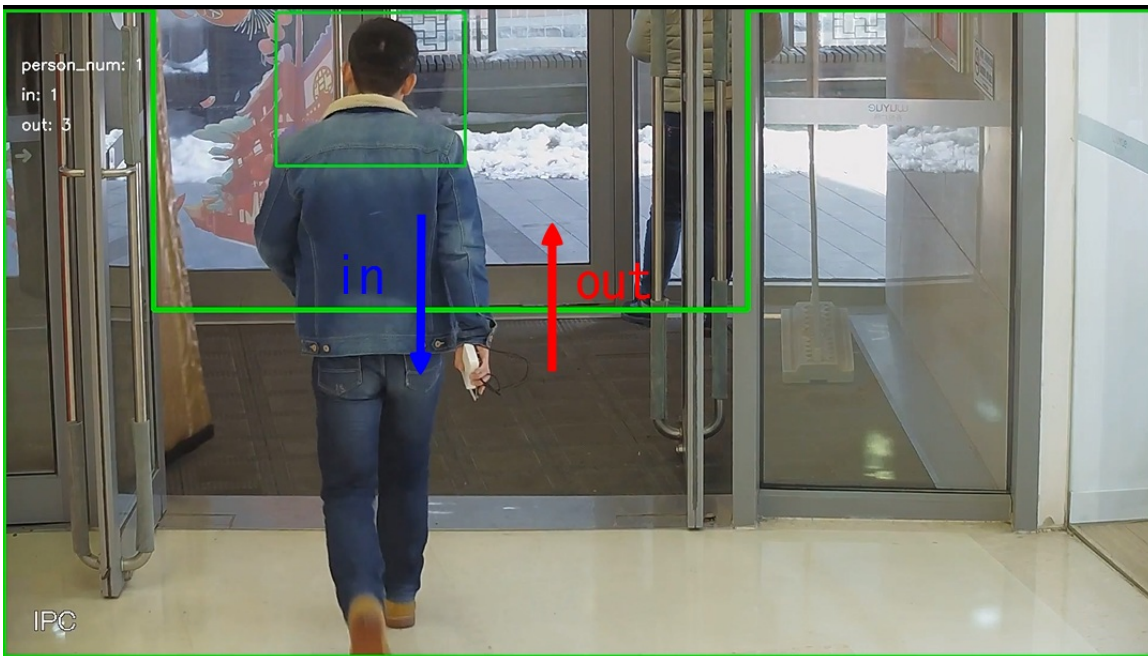
进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

**示例1：**如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。



同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面帧的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面帧的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：

```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 手部关键点识别

### 🔗 手部关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

```

var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用手部关键点识别
client.handAnalysis(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

```

### 手部关键点识别 请求参数详情

参数名称	是否必选	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50px，最长边最大4096px，支持jpg/png/bmp格式

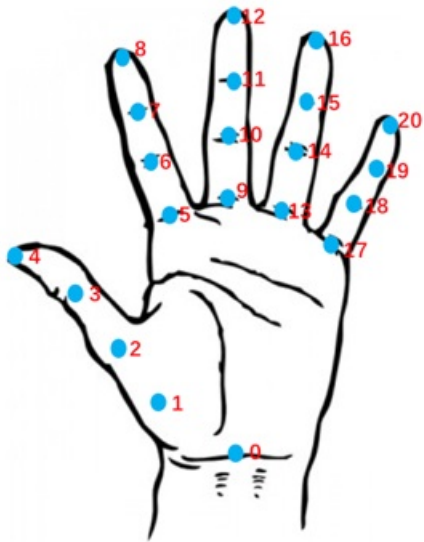
### 手部关键点识别 返回数据参数详情

字段	是否必选	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节点信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

### 手部关键点识别 返回示例

返回关键点对应位置示意图：





```
{
  "log_id": "46532731~~~~32348525916",
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        "2": {
          "y": 533,
          "x": 916,
          "score": 0.75532239675522
        },
        "3": {
          "y": 467,
          "x": 938,
          "score": 0.79282909631729
        },
        "4": {
          "y": 380,
          "x": 993,
          "score": 0.88114696741104
        },
        "5": {
          "y": 424,
          "x": 829,
          "score": 0.90723776817322
        },
        "6": {
          "y": 358,
          "x": 862,
          "score": 0.62833058834076
        },
        "7": {
          "y": 391,
          "x": 916,
```



```
"score": 0.77525424957275
},
"8": {
  "y": 456,
  "x": 927,
  "score": 0.87964057922363
},
"9": {
  "y": 402,
  "x": 775,
  "score": 0.91389346122742
},
"10": {
  "y": 325,
  "x": 786,
  "score": 0.85947573184967
},
"11": {
  "y": 249,
  "x": 818,
  "score": 0.88918441534042
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
```

```

    "z": {
      "y": 325,
      "x": 567,
      "score": 0.91110396385193
    }
  }, ~~~~
  "location": {
    "height": 556,
    "width": 426,
    "top": 151,
    "score": 17.495880126953,
    "left": 567
  }
}
]
}

```

## 错误码

### 错误返回格式

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。

### 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

### 服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交</a>

4	reached	<a href="#">工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在

282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误


## Java SDK文档

### 简介

Hi，您好，欢迎使用百度人体分析服务。

本文档主要针对Java开发者，描述百度人体分析接口服务的相关技术内容。如果您对文档内容有任何疑问，可以通过以下几种方式联系我们：

- 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AI社区交流：<http://ai.baidu.com/forum/topic/list/197>
- 具有免费调用额度的接口，超过每天的免费额度后会返回错误码：17，错误信息：Open api daily request limit reached（每天流量超限额），未上线计费功能的接口，可通过QQ群（860337848）联系群管、或[提交工单](#)手动提额
- 各个接口的免费调用额度（调用量和QPS并发），以及产品价格说明，请参考[产品价格](#)文档

 [接口功能说明](#)

接口名称	接口能力简要描述
人体关键点识别	检测图像中的所有人体并返回每个人体的矩形框位置，精准定位21个核心关键点，包含五官、四肢、脖颈等部位，更多关键点持续扩展中；支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景
人体检测与属性识别	检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持20余种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等；支持中低空俯拍视角、人体重叠、遮挡、背面、侧面、动作变化等复杂场景
人流量统计	识别和统计图像当中的人体个数（静态统计，不支持追踪和去重），适用于3米以上的中远距离俯拍，以人头为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片
手势识别	识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别22种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的手势列表：手指、掌心向前、拳头、OK、祈祷、作揖、作别、单手比心、点赞、diss、rock、掌心向上、双手比心（3种）、数字（7种）
人像分割	识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景；输入正常人像图片，返回分割后的二值结果图，返回的二值图像需要进行二次处理才可查看分割效果
驾驶行为分析	针对车载场景，检测图片中是否有驾驶员，并识别驾驶员是否有使用手机、抽烟、未系安全带、双手离开方向盘、未佩戴口罩、闭眼、打哈欠等行为，可用于分析预警危险驾驶行为
人流量统计	动态人数统计和跟踪，主要适用于低空俯拍、出入口场景，以人体头肩为主要识别目标，核心功能：传入监控视频（动态抓拍图片序列，进行人体追踪，根据目标轨迹判断进出区域行为，进行动态人数统计，返回区域进出人数。
手部关键点识别	检测图片中的所有人手，定位每只手的21个主要骨节点，可用于自定义手势识别、人机交互、AR特效等场景

## 🔗 版本更新记录

上线日期	版本号	更新内容
2019.12.19	4.13.0	新增手部关键点识别接口
2019.2.22	4.10.3	新增驾驶行为分析接口
2019.1.25	4.10.2	新增动态人流量统计接口
2018.9.29	4.7.0	新增人像分割、手势识别接口
2018.6.8	4.4.0	新增人体分析服务

## 快速入门

### 🔗 安装BodyAnalysis Java SDK

#### BodyAnalysis Java SDK目录结构

```
com.baidu.aip
├── auth           //签名相关类
├── http          //Http通信相关类
├── client        //公用类
├── exception     //exception类
├── bodyanalysis
│   └── AipBodyAnalysis //AipBodyAnalysis类
└── util         //工具类
```

支持 JAVA版本 : 1.7+

### 查看源码

Java SDK代码现已公开，您可以查看代码、或者在License范围内修改和编译SDK以适配您的环境。github链接：<https://github.com/Baidu-AIP/java-sdk>

### 使用maven依赖：

添加以下依赖即可。其中版本号可在[maven官网](#)查询

```
<dependency>
  <groupId>com.baidu.aip</groupId>
  <artifactId>java-sdk</artifactId>
  <version>${version}</version>
</dependency>
```

### 直接使用JAR包步骤如下：

- 1.在[官方网站](#)下载Java SDK压缩工具包。
- 2.将下载的aip-java-sdk-version.zip解压后，复制到工程文件夹中。
- 3.在Eclipse右键“工程 -> Properties -> Java Build Path -> Add JARs”。
- 4.添加SDK工具包aip-java-sdk-version.jar和第三方依赖工具包json-20160810.jar slf4j-api-1.7.25.jar slf4j-simple-1.7.25.jar（可选）。

其中，version为版本号，添加完成后，用户就可以在工程中使用BodyAnalysis Java SDK。

### 🔗 新建AipBodyAnalysis

AipBodyAnalysis是人体分析的Java客户端，为使用人体分析的开发人员提供了一系列的交互方法。

用户可以参考如下代码新建一个AipBodyAnalysis,初始化完成后建议**单例使用**,避免重复获取access\_token：

```

public class Sample {
    //设置APPID/AK/SK
    public static final String APP_ID = "你的 App ID";
    public static final String API_KEY = "你的 Api Key";
    public static final String SECRET_KEY = "你的 Secret Key";

    public static void main(String[] args) {
        // 初始化一个AipBodyAnalysis
        AipBodyAnalysis client = new AipBodyAnalysis(APP_ID, API_KEY, SECRET_KEY);

        // 可选：设置网络连接参数
        client.setConnectionTimeoutInMillis(2000);
        client.setSocketTimeoutInMillis(60000);

        // 可选：设置代理服务器地址, http和socket二选一，或者均不设置
        client.setHttpProxy("proxy_host", proxy_port); // 设置http代理
        client.setSocketProxy("proxy_host", proxy_port); // 设置socket代理

        // 调用接口
        String path = "test.jpg";
        JSONObject res = client.bodyAnalysis(path, new HashMap<String, String>());
        System.out.println(res.toString(2));
    }
}

```

在上面代码中，常量APP\_ID在百度云控制台中创建，常量API\_KEY与SECRET\_KEY是在创建完毕应用后，系统分配给用户的，均为字符串，用于标识用户，为访问做签名验证，可在AI服务控制台中的应用列表中查看。

**注意：**如您以前是百度云的老用户，其中API\_KEY对应百度的“Access Key ID”，SECRET\_KEY对应百度的“Access Key Secret”。

#### 🔗 配置AipBodyAnalysis

如果用户需要配置AipBodyAnalysis的一些细节参数，可以在构造AipBodyAnalysis之后调用接口设置参数，目前只支持以下参数：

接口	说明
setConnectionTimeoutInMillis	建立连接的超时时间（单位：毫秒）
setSocketTimeoutInMillis	通过打开的连接传输数据的超时时间（单位：毫秒）
setHttpProxy	设置http代理服务器
setSocketProxy	设置socket代理服务器（http和socket类型代理服务器只能二选一）

SDK默认使用slf4j-simple包进行日志输出，若用户需要使用自定义日志实现，可去除slf4j-simple依赖包，再额外添加相应的日志实现包即可。maven去除slf4j-simple依赖包示例：



```
<dependency>
  <groupId>com.baidu.aip</groupId>
  <artifactId>java-sdk</artifactId>
  <version>${version}</version>
  <exclusions>
    <exclusion>
      <groupId>org.slf4j</groupId>
      <artifactId>slf4j-simple</artifactId>
    </exclusion>
  </exclusions>
</dependency>
```

## 人体关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的所有人体，输出每个人体的21个主要关键点，包含头顶、五官、脖颈、四肢等部位，同时输出人体的坐标信息和数量。

支持多人检测、人体位置重叠、遮挡、背面、侧面、中低空俯拍、大动作等复杂场景。

21个关键点的位置：头顶、左耳、右耳、左眼、右眼、鼻子、左嘴角、右嘴角、脖子、左肩、右肩、左手肘、右手肘、左手腕、右手腕、左髌部、右髌部、左膝、右膝、左脚踝、右脚踝。示意图如下，正在持续扩展更多关键点，敬请期待。

单人场景：



多人场景：





```

public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.bodyAnalysis(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.bodyAnalysis(file, options);
    System.out.println(res.toString(2));
}

```

#### 人体关键点识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

#### 人体关键点识别 返回数据参数详情

接口除了返回人体框和每个关键点的坐标信息外，还会输出人体框和关键点的概率分数，实际应用中可以基于概率分数进行过滤，排除掉分数低的误识别“无效人体”，推荐的过滤方案：当关键点得分大于0.2的个数大于3，且人体框的得分大于0.03时，才认为是有效人体。

实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用，比如对误识别容忍低的应用场景，人体框的得分阈值可以提到0.05甚至更高。

字段	是否必选	类型	说明
person_num	是	uint32	人体数目
person_info	是	object[]	人体姿态信息
+body_parts	是	object	身体部位信息，包含21个关键点
++top_head	是	object	头顶

+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_eye	是	object	左眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_eye	是	object	右眼
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++nose	是	object	鼻子
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ear	是	object	左耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ear	是	object	右耳
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_mouth_corner	是	object	左嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_mouth_corner	是	object	右嘴角
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++neck	是	object	颈部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_shoulder	是	object	左肩
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_shoulder	是	object	右肩

+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_elbow	是	object	左手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_elbow	是	object	右手肘
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_wrist	是	object	左手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_wrist	是	object	右手腕
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_hip	是	object	左髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_hip	是	object	右髋部
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_knee	是	object	左膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_knee	是	object	右膝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++left_ankle	是	object	左脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
++right_ankle	是	object	右脚踝

++right_ankle	是	object	右脚踝
+++x	是	float	x坐标
+++y	是	float	y坐标
+++score	是	float	概率分数
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	是	float	人体框的概率分数
log_id	是	uint64	唯一的log id, 用于问题定位

说明：

1、body\_parts，一共21个part，每个part包含x, y两个坐标，如果part被截断，则x、y坐标为part被截断的图片边界位置，part顺序以实际返回顺序为准。

2、接口返回人体坐标框和每个关键点的置信度分数，在应用时可综合置信度score分数，过滤掉置信度低的“无效人体”，**建议过滤方法**：当关键点得分大于0.2的个数大于3，且人体框的分数大于0.03时，才认为是有效人体。实际应用中，可根据对误识别、漏识别的容忍程度，调整阈值过滤方案，灵活应用。

#### 人体关键点识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "body_parts": {
        "left_hip": {
          "y": 573,
          "x": 686.09375,
          "score": 0.78743487596512
        },
        "top_head": {
          "y": 242.53125,
          "x": 620,
          "score": 0.87757384777069
        },
        "right_mouth_corner": {
          "y": 308.625,
          "x": 606.78125,
          "score": 0.90121293067932
        },
        "neck": {
          "y": 335.0625,
          "x": 620,
          "score": 0.84662038087845
        },
        "left_shoulder": {
          "y": 361.5,
          "x": 699.3125,
          "score": 0.83550786972046
        },
        "left_knee": {
          "y": 731.625,
          "x": 699.3125,
```

```
"score": 0.83575332164764
},
"left_ankle": {
  "y": 877.03125,
  "x": 725.75,
  "score": 0.85220056772232
},
"left_mouth_corner": {
  "y": 308.625,
  "x": 633.21875,
  "score": 0.91475087404251
},
"right_elbow": {
  "y": 348.28125,
  "x": 461.375,
  "score": 0.81766486167908
},
"right_ear": {
  "y": 282.1875,
  "x": 593.5625,
  "score": 0.86551451683044
},
"nose": {
  "y": 295.40625,
  "x": 620,
  "score": 0.90894532203674
},
"left_eye": {
  "y": 282.1875,
  "x": 633.21875,
  "score": 0.89628517627716
},
"right_eye": {
  "y": 282.1875,
  "x": 606.78125,
  "score": 0.89676940441132
},
"right_hip": {
  "y": 586.21875,
  "x": 593.5625,
  "score": 0.79803824424744
},
"left_wrist": {
  "y": 374.71875,
  "x": 884.375,
  "score": 0.89635348320007
},
"left_ear": {
  "y": 295.40625,
  "x": 659.65625,
  "score": 0.86607384681702
},
"left_elbow": {
  "y": 361.5,
  "x": 791.84375,
  "score": 0.83910942077637
},
"right_shoulder": {
  "y": 348.28125,
  "x": 553.90625,
  "score": 0.85635334253311
},
```

```
"right_ankle": {
  "y": 890.25,
  "x": 580.34375,
  "score": 0.85149073600769
},
"right_knee": {
  "y": 744.84375,
  "x": 580.34375,
  "score": 0.83749794960022
},
"right_wrist": {
  "y": 348.28125,
  "x": 368.84375,
  "score": 0.83893859386444
}
},
"location": {
  "height": 703.20654296875,
  "width": 652.61810302734,
  "top": 221.92272949219,
  "score": 0.99269664287567,
  "left": 294.03039550781
}
},
},
"log_id": "6362401025381690607"
}
```

## 人体检测与属性识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图像中的所有人体并返回每个人体的矩形框位置，识别人体的静态属性和行为，共支持22种属性，包括：性别、年龄阶段、衣着（含类别/颜色）、戴帽子（可区分安全帽/普通帽）、戴口罩、戴眼镜、背包、使用手机、抽烟、身体朝向等。

主要适用于监控场景的中低空斜拍视角，支持人体轻度重叠、轻度遮挡、背面、侧面、动作变化等复杂场景。

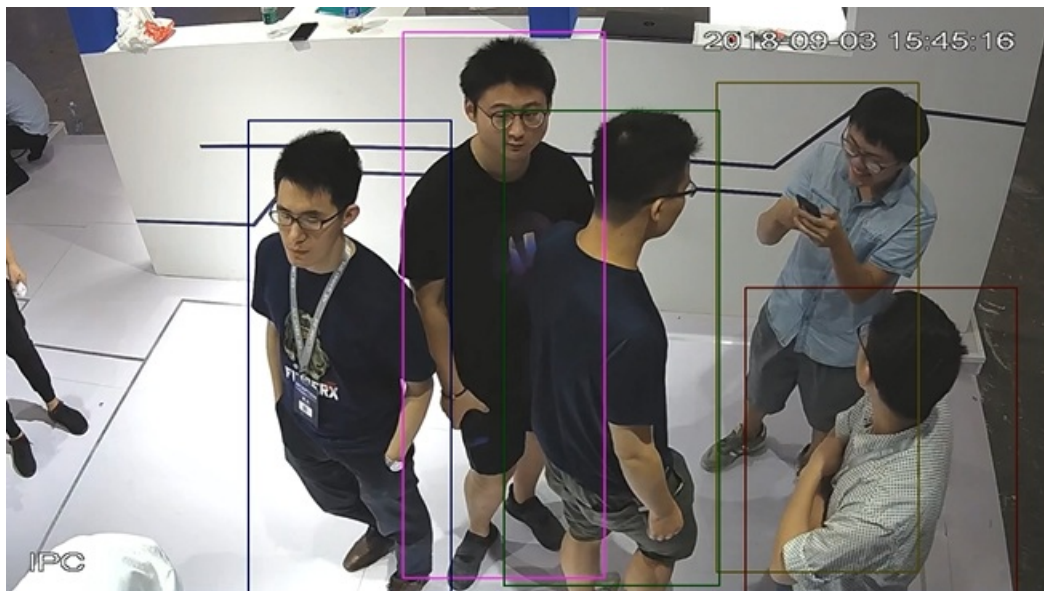
摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。

暂不适用夜间红外监控图片，后续会考虑扩展。

注：接口返回的属性信息包括人体的遮挡、截断情况，在应用时可基于此过滤掉“无效人体”，比如严重遮挡、严重截断的人体。

人体检测的效果示例如下：





非正常人体示例：严重截断



22种属性及其输出项如下，持续扩展更多属性，敬请期待。

序号	属性	接口字段	输出项说明
1	性别	gender	男性、女性
2	年龄阶段	age	幼儿、青少年、青年、中年、老年
3	上身服饰	upper_wear	长袖、短袖
4	下身服饰	lower_wear	长裤、短裤、长裙、短裙、不确定
5	上身服饰颜色	upper_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕
6	下身服饰颜色	lower_color	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
7	上身服饰纹理	upper_wear_texture	纯色、图案、碎花、条纹或格子
8	背包	bag	无背包、单肩包、双肩包、不确定
9	上身服饰细分类	upper_wear_fg	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套
10	是否戴帽子	headwear	无帽、普通帽、安全帽
11	是否戴口罩	face_mask	无口罩、戴口罩、不确定

	属性	SK	
12	是否戴眼镜	glasses	戴眼镜、戴墨镜、无眼镜、不确定
13	是否撑伞	umbrella	撑伞、无撑伞
14	是否使用手机	cellphone	未使用手机、使用手机、不确定
15	身体朝向	orientation	正面、背面、左侧面、右侧面
16	是否吸烟	smoke	吸烟、未吸烟、不确定
17	是否有手提物	carrying_item	无手提物、有手提物、不确定
18	交通工具	vehicle	无交通工具、骑摩托车、骑自行车、骑三轮车
19	上方截断	upper_cut	无上方截断、有上方截断
20	下方截断	lower_cut	无下方截断、有下方截断
21	遮挡	occlusion	无遮挡、轻度遮挡、重度遮挡
22	是否是正常人体	is_human	非正常人体、正常人体； <b>正常人体</b> ：身体露出大于二分之一的人体，一般以能看到腰部肚脐眼为标准； <b>非正常人体</b> ：严重截断、严重遮挡的人体，一般是看不到肚脐眼的，比如只有个脑袋、一条腿

注意：接口默认输出所有22个属性，如只需返回某几个特定属性，请将type 参数值设定属性可选值，用逗号分隔。

```
public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("type", "gender");

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.bodyAttr(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.bodyAttr(file, options);
    System.out.println(res.toString(2));
}
```

人体检测与属性识别 请求参数详情



参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
type	否	String	<p>1) 可选值说明：</p> <p>gender-性别， age-年龄阶段， lower_wear-下身服饰， upper_wear-上身服饰， headwear-是否戴帽子， face_mask-是否戴口罩， glasses-是否戴眼镜， upper_color-上身服饰颜色， lower_color-下身服饰颜色， cellphone-是否使用手机， upper_wear_fg-上身服饰细分类， upper_wear_texture-上身服饰纹理， orientation-身体朝向， umbrella-是否撑伞； bag-背包， smoke-是否吸烟， vehicle-交通工具， carrying_item-是否有手提物， upper_cut-上方截断， lower_cut-下方截断， occlusion-遮挡， is_human-是否是正常人体</p> <p>2) type 参数值可以是可选值的组合，用逗号分隔；如果无此参数默认输出全部22个属性</p>

#### 人体检测与属性识别 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	uint64	唯一的log id，用于问题定位
person_num	是	uint32	<b>人体数目</b>
person_info	是	object[]	人体姿态信息
+location	是	object	人体坐标信息
++height	是	float	人体区域的高度
++left	是	float	人体区域离左边界的距离
++top	是	float	人体区域离上边界的距离
++width	是	float	人体区域的宽度
++score	否	float	人体置信度分数，取值0-1，越接近1，代表人体的概率越大
+attributes	否	object	人体属性内容
++gender	否	object	性别
+++name	否	string	男性、女性、不确定
+++score	否	float	对应概率分数
++age	否	object	年龄阶段
+++name	否	string	幼儿、青少年、青年、中年、老年、不确定
+++score	否	float	对应概率分数

++upper_color	否	object	上半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++lower_color	否	object	下半身衣着颜色
+++name	否	string	红、橙、黄、绿、蓝、紫、粉、黑、白、灰、棕、不确定
+++score	否	float	对应概率分数
++cellphone	否	object	是否使用手机
+++name	否	string	未使用手机、使用手机、不确定
+++score	否	float	对应概率分数
++lower_wear	否	object	下半身服饰
+++name	否	string	长裤、短裤、长裙、短裙、不确定
+++score	否	float	对应概率分数
++upper_wear	否	object	上半身服饰
+++name	否	string	长袖、短袖、不确定
+++score	否	float	对应概率分数
++headwear	否	object	是否戴帽子
+++name	否	string	无帽、普通帽、安全帽、不确定
+++score	否	float	对应概率分数
++face_mask	否	object	是否戴口罩
+++name	否	string	无口罩、戴口罩、不确定（背面或者头部被截断的人体，一般为不确定）
+++score	否	float	对应概率分数
++glasses	否	object	是否戴眼镜
+++name	否	string	戴眼镜、戴墨镜、无眼镜、不确定
+++score	否	float	对应概率分数
++upper_wear_fg	否	object	上身服饰细分类
+++name	否	string	T恤、无袖、衬衫、西装、毛衣、夹克、羽绒服、风衣、外套、不确定
+++score	否	float	对应概率分数
++upper_wear_texture	否	object	上身服饰纹理
+++name	否	string	纯色、图案、碎花、条纹或格子、不确定
+++score	否	float	对应概率分数
++orientation	否	object	身体朝向
+++name	否	string	正面、背面、侧面、不确定
+++score	否	float	对应概率分数
++umbrella	否	object	是否撑伞
+++name	否	string	未撑伞、撑伞、不确定
+++score	否	float	对应概率分数
++bag	否	object	背包
+++name	否	string	无背包、单肩包、双肩包、不确定
+++score	否	float	对应概率分数
++smoke	不	object	是否吸烟

++smoke	否	object	是否吸烟
+++name	否	string	未吸烟、吸烟、不确定
+++score	否	float	对应概率分数
++vehicle	否	object	交通工具
+++name	否	string	无交通工具、骑摩托车、骑自行车、骑三轮车、不确定
+++score	否	float	对应概率分数
++upper_cut	否	object	上方截断
+++name	否	string	无上方截断，有上方截断，不确定
+++score	否	float	对应概率分数
++lower_cut	否	object	下方截断
+++name	否	string	无下方截断，有下方截断，不确定
+++score	否	float	对应概率分数
++occlusion	否	object	遮挡
+++name	否	string	无遮挡，轻度遮挡，重度遮挡，不确定
+++score	否	float	对应概率分数
++is_human	否	object	是否是正常人体
+++name	否	string	正常人体，非正常人体，不确定
+++score	否	float	对应概率分数

#### 人体检测与属性识别 返回示例

```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "upper_wear_fg": {
          "score": 0.6084373593330383,
          "name": "T恤"
        },
        "cellphone": {
          "score": 0.9986445307731628,
          "name": "未使用手机"
        },
        "lower_cut": {
          "score": 0.999593198299408,
          "name": "无下方截断"
        },
        "umbrella": {
          "score": 0.9999368190765381,
          "name": "未打伞"
        },
        "orientation": {
          "score": 0.6865364909172058,
          "name": "左侧面"
        },
        "headwear": {
          "score": 0.323924720287323,
          "name": "普通帽"
        },
        "gender": {
          "score": 0.8240132331848145,
          "name": "女性"
        }
      }
    }
  ]
}
```

```
    "name": "女性",
  },
  "age": {
    "score": 0.9495382308959961,
    "name": "青年"
  },
  "upper_cut": {
    "score": 0.9999241828918457,
    "name": "无上方截断"
  },
  "glasses": {
    "score": 0.9292197823524475,
    "name": "无眼镜"
  },
  "face_mask": {
    "score": 0.99429154396057,
    "name": "无口罩"
  },
  "lower_color": {
    "score": 0.3373721539974213,
    "name": "黄"
  },
  "bag": {
    "score": 0.9770514369010925,
    "name": "无背包"
  },
  "upper_wear_texture": {
    "score": 0.7113875150680542,
    "name": "纯色"
  },
  "smoke": {
    "score": 0.9982085227966309,
    "name": "未吸烟"
  },
  "vehicle": {
    "score": 0.9907247424125671,
    "name": "无交通工具"
  },
  "lower_wear": {
    "score": 0.9739166498184204,
    "name": "短裤"
  },
  "carrying_item": {
    "score": 0.9735746383666992,
    "name": "无手提物"
  },
  "upper_wear": {
    "score": 0.7128473520278931,
    "name": "长袖"
  },
  "occlusion": {
    "score": 0.999269425868988,
    "name": "无遮挡"
  },
  "upper_color": {
    "score": 0.4870010614395142,
    "name": "白"
  },
  "is_human": {
    "score": 0.4870010614395142,
    "name": "正常人体"
  }
}
```

```
"location": {  
  "height": 369,  
  "width": 316,  
  "top": 1,  
  "score": 0.9756121635437012,  
  "left": 58  
}  
},  
"log_id": 6482262517986539034  
}
```

## 人流量统计

对于输入的一张图片（可正常解码，且长宽比适宜），识别和统计图像当中的人体个数（静态统计，暂不支持追踪和去重）。适用于3米以上的中远距离俯拍，以头部为主要识别目标统计人数，无需正脸、全身照，适应各类人流密集场景（如：机场、车展、景区、广场等）；默认识别整图中的人数，支持指定不规则区域的人数统计，同时可输出渲染图片。

摄像头硬件选型无特殊要求，分辨率建议720p以上，更低分辨率的图片也能识别，只是效果可能有差异。暂不适用夜间红外监控图片，后续会考虑扩展。

渲染图示意如下：



注：接口默认返回整张图片中的人数，如需统计特定框选区域的人数，请使用area参数添加识别区域坐标信息。

```

public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("area", "x1,y1,x2,y2,x3,y3...xn,yn");
    options.put("show", "false");

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.bodyNum(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.bodyNum(file, options);
    System.out.println(res.toString(2));
}

```

### 人流量统计 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
area	否	String	<p>特定框选区域坐标，支持多个多边形区域，最多支持10个区域，如输入超过10个区域，截取前10个区域进行识别。</p> <p>此参数为空或无此参数、或area参数设置错误时，默认识别整个图片的人数。</p> <p>area参数设置错误的示例：某个坐标超过原图大小，x、y坐标未成对出现等；注意：设置了多个区域时，任意一个坐标设置错误，则认为area参数错误、失效。</p> <p>area参数设置格式：</p> <p>1) 多个区域用英文分号“;”分隔；</p> <p>2) 同一个区域内的坐标用英文逗号“,”分隔，默认尾点和首点相连做闭合。</p> <p>示例：</p> <p>1) 单个多边形区域：x1,y1,x2,y2,x3,y3...xn,yn</p> <p>2) 多个多边形区域：xa1,ya1,xa2,ya2,xa3,ya3...xan,yan;xb1,yb1,xb2,yb2,xb3,yb3...xbn,ybn;..</p>
show	否	String	是否输出渲染的图片，默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false

### 人流量统计 返回数据参数详情

字段	是否必选	类型	
person_num	是	uint64	识别出的人体数目；当未设置area参数时，返回的是全图人数；设置了有效的area参数时，返回的人数是所有区域的人数总和（所有区域求并集后的不规则区域覆盖的人数）
image	否	string	渲染后的图片，输入参数show=true时输出该字段
area_counts	否	array	每一个框选区域的人数，仅当请求中有area参数且参数有效时才会返回，否则该字段不返回；成功返回示例：[5,3,8]

### 人流量统计 返回示例

```
{
  "log_id": 716033439,
  "person_num": 16,
  "image": "/9j/4AAoFS2P/9k="
}
```

## 手势识别

识别图片中的手势类型，返回手势名称、手势矩形框、概率分数，可识别24种手势，支持动态手势识别，适用于手势特效、智能家居手势交互等场景；支持的24类手势列表：拳头、OK、祈祷、作揖、作别、单手比心、点赞、Diss、我爱你、掌心向上、双手比心（3种）、数字（9种）、Rock、竖中指。

主要适用于3米以内的自拍、他人拍摄，1米内识别效果最佳，拍摄距离太远时，手部目标太小，无法准确定位和识别。


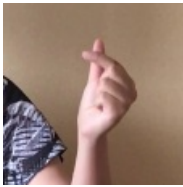


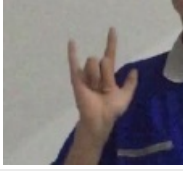
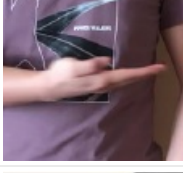


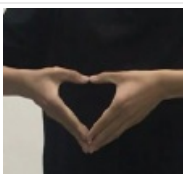


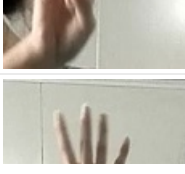
图片中有多个手势时，也能识别，但该情况下，单个手势的目标可能较小，且角度可能不好（例如存在倾斜、遮挡等），识别效果可能受影响。建议针对单个手势进行识别，效果最佳。

注：

- 1) 上述24类以外的其他手势会划分到other类。
- 2) 除识别手势外，若图像中检测到人脸，会同时返回人脸框位置。

可识别的24种手势示意图如下：

序号	手势名称	classname	示例图
1	数字1（原食指）	One	
2	数字5（原掌心向前）	Five	
3	拳头	Fist	
4	OK	OK	
5	祈祷	Prayer	
6	作揖	Congratulation	
			

7	作别	Honour	
8	单手比心	Heart_single	
9	点赞	Thumb_up	
10	Diss	Thumb_down	
11	Rock	ILY	
12	掌心向上	Palm_up	
13	双手比心1	Heart_1	
14	双手比心2	Heart_2	
15	双手比心3	Heart_3	
16	数字2	two	
17	数字3	three	
18	数字4	four	



18	双丁手	two	
19	数字6	six	
20	数字7	seven	
21	数字8	eight	
22	数字9	nine	
23	Rock	Rock	
24	竖中指	Insult	

```

public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.gesture(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.gesture(file, options);
    System.out.println(res.toString(2));
}

```

#### 手势识别 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

#### 手势识别 返回数据参数详情

字段	是否必选	类型	说明
result_num	是	int	结果数量
result	是	object[]	检测到的目标，手势、人脸
+classname	否	string	目标所属类别，24种手势、other、face
+top	否	int	目标框上坐标
+width	否	int	目标框的宽
+left	否	int	目标框最左坐标
+height	否	int	目标框的高
+probability	否	float	目标属于该类别的概率
log_id	是	int64	唯一的log id，用于问题定位

### 手势识别 返回示例

```
{
  "log_id": 4466502370458351471,
  "result_num": 2,
  "result": [
    {
      "probability": 0.9844077229499817,
      "top": 20,
      "height": 156,
      "classname": "Face",
      "width": 116,
      "left": 173
    },
    {
      "probability": 0.4679304957389832,
      "top": 157,
      "height": 106,
      "classname": "Heart_2",
      "width": 177,
      "left": 183
    }
  ]
}
```

### 人像分割

识别人体的轮廓范围，与背景进行分离，适用于拍照背景替换、照片合成、身体特效等场景。输入正常人像图片，返回分割后的二值结果图、灰度图、透明背景的人像图（png格式）；并输出画面中的人数、人体坐标信息，可基于此对图片进行过滤、筛选，如筛选出大于x人的图片进行特殊处理。

分割效果示意图：

1) 原图



2) 二值图



3) 灰度图



4) 前景人像图 (透明背景)



注：返回的二值图像需要进行二次处理才可查看分割效果；灰度图和前景人像图不用处理，直接解码保存图片即可。

```
public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("type", "labelmap");

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.bodySeg(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.bodySeg(file, options);
    System.out.println(res.toString(2));
}
```

#### 人像分割 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
type	否	String	可以通过设置type参数，自主设置返回哪些结果图，避免造成带宽的浪费 1) 可选值说明： labelmap - 二值图像，需二次处理方能查看分割效果 scoremap - 人像前景灰度图 foreground - 人像前景抠图，透明背景 2) type 参数值可以是可选值的组合，用逗号分隔；若无此参数默认输出全部3类结果图

#### 人像分割 返回数据参数详情

字段	是否必选	类型	说明
log_id	是	int64	唯一的log id, 用于问题定位
labelmap	否	string	分割结果图片, base64编码之后的二值图像, 需二次处理方能查看分割效果
scoremap	否	string	分割后人像前景的scoremap, 归一到0-255, 不用进行二次处理, 直接解码保存图片即可。 Base64编码后的灰度图文件, 图片中每个像素点的灰度值 = 置信度 * 255, 置信度为原图对应像素点位于人体轮廓内的置信度, 取值范围[0, 1]
foreground	否	string	分割后的人像前景抠图, 透明背景, Base64编码后的png格式图片, 不用进行二次处理, 直接解码保存图片即可。将置信度大于0.5的像素抠出来, 并通过image matting技术消除锯齿
person_num	是	int32	检测到的人体框数目
person_info	否	object[]	人体框信息
+height	否	float	人体区域的高度
+left	否	float	人体区域离左边界的距离
+top	否	float	人体区域离上边界的距离
+width	否	float	人体区域的宽度
+score	否	float	人体框的概率分数, 取值0-1

### 人像分割 返回示例

```
{
  "log_id": 2451426617428943180,
  "labelmap": "iVBORwOKGg",
  "scoremap": "iVBORwOKGg",
  "foreground": "iVBORwOKGg",
  "person_num": 2,
  "person_info": [
    {
      "height": 420.9641110897064,
      "width": 365.67474365234375,
      "top": 3.704406976699829,
      "score": 0.9801973104476929,
      "left": 229.32940673828125
    },
    {
      "height": 371.6713676452637,
      "width": 167.91799926757812,
      "top": 49.91801834106445,
      "score": 0.4228516221046448,
      "left": 470.4878234863281
    }
  ]
}
```

### 驾驶行为分析

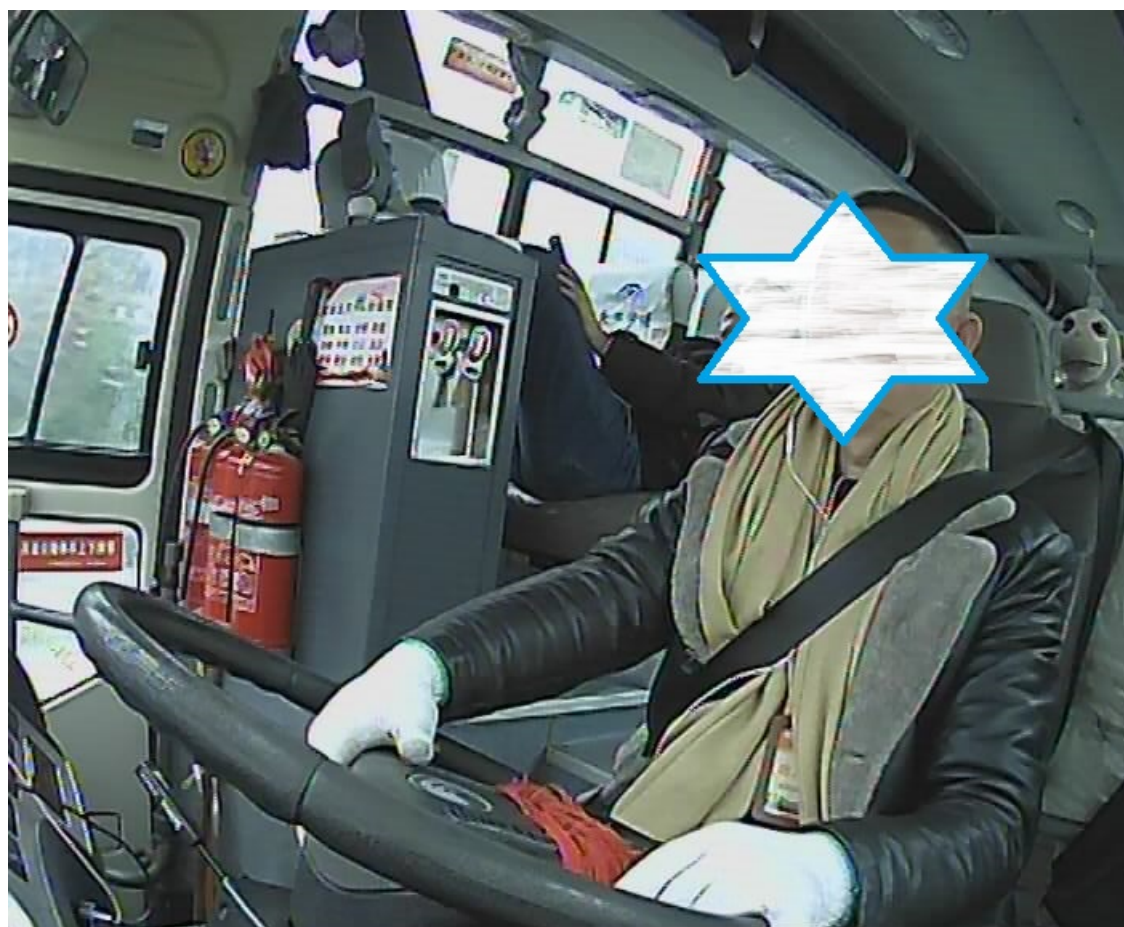
对于输入的一张车载驾驶员监控图片（可正常解码，且长宽比适宜），识别图像中是否有人体，若检测到至少1个人体，将目标最大的人体作为驾驶员，进一步识别驾驶员的属性行为，可识别使用手机、抽烟、未系安全带、双手离开方向盘、视线未朝前方、未佩戴口罩、闭眼、打哈欠、低头9种典型行为姿态。

注：若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员；针对香港、海外地区的右舵车，可通过请求参数里的wheel\_location字段，指定将左侧最大的人体作为驾驶员。

图片质量要求：

- 服务只适用于车载驾驶室监控场景，普通室内外监控场景，若要识别使用手机、抽烟等行为属性，请使用[人体检测与属性识别服务](#)。
- 车内摄像头硬件选型无特殊要求，分辨率建议720p以上，但更低分辨率的图片也能识别，只是效果可能有差异。
- 车内摄像头部署方案建议：尽可能拍全驾驶员的身体，并充分考虑背光、角度、方向盘遮挡等因素。
- 服务适用于夜间红外监控图片，识别效果跟可见光图片相比可能略微有差异。
- 图片主体内容清晰可见，模糊、驾驶员遮挡严重、光线暗等情况下，识别效果肯定不理想。

示例图参考：



```

public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("type", "smoke");

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.driverBehavior(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.driverBehavior(file, options);
    System.out.println(res.toString(2));
}

```

### 驾驶行为分析 请求参数详情

参数名称	是否必选	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据
type	否	String	识别的属性行为类别，英文逗号分隔，默认所有属性都识别； smoke //吸烟， cellphone //打手机， not_buckling_up // 未系安全带， both_hands_leaving_wheel // 双手离开方向盘， not_facing_front // 视角未看前方， no_face_mask // 未正确佩戴口罩， yawning // 打哈欠， eyes_closed // 闭眼， head_lowered // 低头
wheel_location	否	string	默认值"1"，表示左舵车（普遍适用于中国大陆地区，若图像中检测到多个大小相当的人体，默认取画面中右侧最大的人体作为驾驶员）； "0"表示右舵车（适用于香港等地区，若图像中检测到多个大小相当的人体，则取画面中左侧最大的人体作为驾驶员）； 其他输入值视为非法输入，直接使用默认值

### 驾驶行为分析 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	uint64	检测到的总人数（包括驾驶员和乘客），0代表未监测到驾驶员
driver_num	是	uint64	检测到的驾驶员数目。若大于1，则综合考虑人体框尺寸和位置，选取最佳驾驶员目标框进行属性分析，默认取画面中右侧最大的人体作为驾驶员（普遍适用于中国大陆地区）
person_info	是	object[]	驾驶员的属性行为信息；若未检测到驾驶员，则该项为[]
+location	否	object	检测出驾驶员的位置
++left	否	int	检测区域在原图的左起开始位置
++top	否	int	检测区域在原图的上起开始位置
++width	否	int	检测区域宽度
++height	否	int	检测区域高度

+attributes	否	object	驾驶员属性行为内容
++smoke	否	object	吸烟
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++cellphone	否	object	使用手机
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_buckling_up	否	object	未系安全带
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++both_hands_leaving_wheel	否	object	双手离开方向盘
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++not_facing_front	否	object	视角未朝前方
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++no_face_mask	否	object	未正确佩戴口罩，包含戴了口罩、但口鼻外露这类未戴好的情况
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++yawning	否	object	打哈欠，实际应用时，可结合闭眼综合判断疲劳，避免普通张嘴、说话等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++eyes_closed	否	object	闭眼，实际应用时，可结合打哈欠综合判断疲劳，避免正常眨眼等情况下被误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可
++head_lowered	否	object	低头，实际应用时，可结合闭眼、视角未朝前方综合判断分心、疲劳，避免单一属性引起误判
+++score	否	float	对应概率分数
+++threshold	否	float	建议阈值，仅作为参考，实际应用中根据测试情况选取合适的score阈值即可

驾驶行为分析 返回示例



```
{
  "person_num": 1,
  "person_info": [
    {
      "attributes": {
        "cellphone": {
          "threshold": 0.76,
          "score": 0.089325942099094
        },
        "yawning": {
          "threshold": 0.66,
          "score": 0.0007511890726164
        },
        "not_buckling_up": {
          "threshold": 0.58,
          "score": 0.81095975637436
        },
        "no_face_mask": {
          "threshold": 0.72,
          "score": 0.99875915050507
        },
        "both_hands_leaving_wheel": {
          "threshold": 0.3,
          "score": 0.9014720916748
        },
        "eyes_closed": {
          "threshold": 0.1,
          "score": 0.090511165559292
        },
        "head_lowered": {
          "threshold": 0.58,
          "score": 0.11450858414173
        },
        "smoke": {
          "threshold": 0.25,
          "score": 0.026156177744269
        },
        "not_facing_front": {
          "threshold": 0.53,
          "score": 0.68074524402618
        }
      },
      "location": {
        "width": 856,
        "top": 419,
        "score": 0.90945136547089,
        "left": 464,
        "height": 626
      }
    }
  ],
  "log_id": 2320165720061799596
}
```

## 人流量统计-动态版

统计图像中的人体个数和流动趋势，主要适用于**低空俯拍、出入口场景**，以**人体头肩**为主要识别目标，核心功能：

- **静态人数统计**：传入单帧图像，检测图片中的人体头肩，返回图中总人数。
- **动态人数统计和跟踪**：传入监控视频抓拍图片序列，进行人体追踪，返回每个人体框的坐标和所属ID；并根据目标轨迹判断

进出区域行为，进行动态人数统计，返回区域进出人数。同时可输出渲染结果图（含统计值和跟踪框渲染）。（注：抽帧频率需 > 2fps，否则无法有效跟踪，建议5fps，接口默认保证5qps，每天赠送5万次免费调用量，以便充分测试。）

渲染图示例：



```
public void sample(AipBodyAnalysis client) {  
    // 传入可选参数调用接口  
    HashMap<String, String> options = new HashMap<String, String>();  
    options.put("case_id", "123");  
    options.put("case_init", "true");  
    options.put("show", "true");  
    options.put("area", "100");  
  
    String dynamic = "true";  
  
    // 参数为本地路径  
    String image = "test.jpg";  
    JSONObject res = client.bodyTracking(image, dynamic, options);  
    System.out.println(res.toString(2));  
  
    // 参数为二进制数组  
    byte[] file = readFile("test.jpg");  
    res = client.bodyTracking(file, dynamic, options);  
    System.out.println(res.toString(2));  
}
```

人流量统计-动态版 请求参数详情

参数名称	是否必选	类型	可选值范围	默认值	说明
image	是	mixed			本地图片路径或者图片二进制数据
dynamic	是	String	true false		true：动态人流量统计，返回总人数、跟踪ID、区域进出人数； false：静态人数统计，返回总人数
case_id	当dynamic为True时，必填	String			任务ID（通过case_id区分不同视频流，自拟，不同序列间不可重复即可）
case_init	当dynamic为True时，必填	String	true false		每个case的初始化信号，为true时对该case下的跟踪算法进行初始化，为false时重载该case的跟踪状态。当为false且读取不到相应case的信息时，直接重新初始化
show	否	String	true false	false	否返回结果图（含统计值和跟踪框渲染），默认不返回，选true时返回渲染后的图片(base64)，其它无效值或为空则默认false
area	当dynamic为True时，必填	String			静态人数统计时，只统计区域内的人，缺省时为全图统计。 动态人流量统计时，进出区域的人流会被统计。 逗号分隔，如'x1,y1,x2,y2,x3,y3...xn,yn'，按顺序依次给出每个顶点的xy坐标（默认尾点和首点相连），形成闭合多边形区域。 服务会做范围（顶点左边需在图像范围内）及个数校验（数组长度必须为偶数，且大于3个顶点）。只支持单个多边形区域，建议设置矩形框，即4个顶点。坐标取值不能超过图像宽度和高度，比如1280的宽度，坐标值最小建议从1开始，最大到1279。

#### area参数设置说明

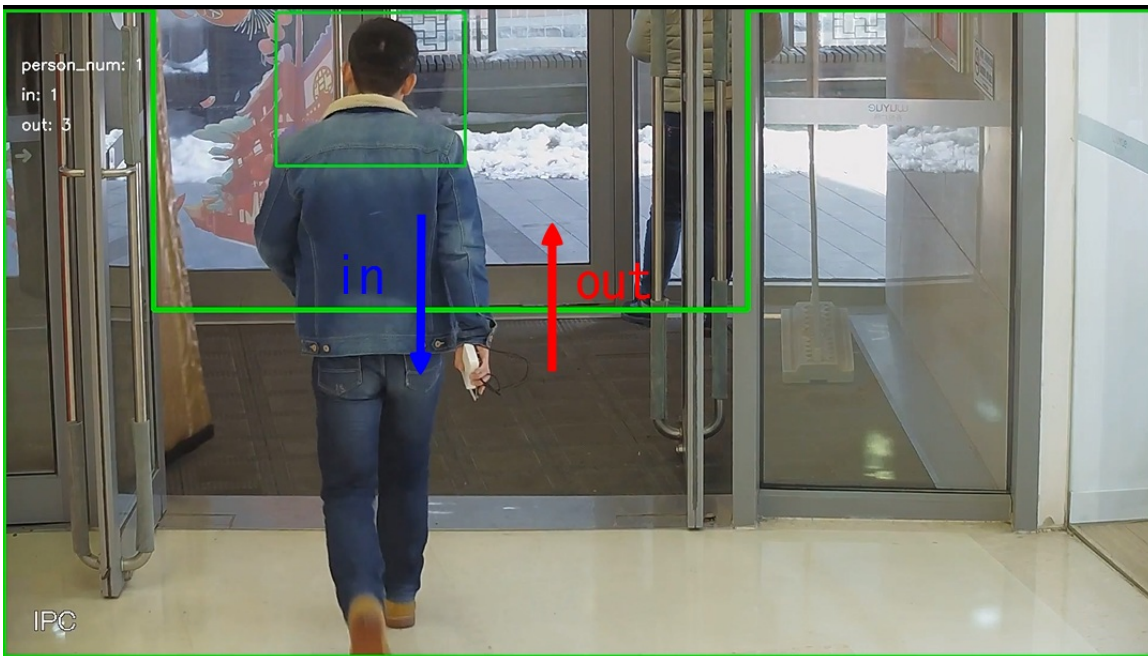
进出区域方向：从区域外走到区域内就是in，相反就是out，详见下方示例。

**示例1：**如下图，area区域框三条边贴着图像左方边缘，从图像右方往左走到框里就是in，从图像左方往右走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像左方向右走是in，就把框画在图像右半部分，上、下、右三条边贴着图像边缘。



同理，上下方向，如果area区域框三条边贴着图像下方边缘，从图像上方往下走到框里就是in，从图像下方往上走出框就是out，相当于只有图像中间那条线起作用。如果想要从图像下方向上走是in，就把框画在图像上半部分，上、左、右三条边贴着图像边缘。

示例2：如下图，area区域是一个不规则多边形，将画面中门口以外的部分都框起来了，蓝色箭头的方向代表in，人从门外走进区域框里，红色箭头的方向代表out，人走出区域框，走向门外。



人流量统计-动态版 返回数据参数详情

字段	是否必选	类型	说明
person_num	是	int	检测到的人体框数目
person_info	否	object[]	每个框的具体信息
+location	否	object	跟踪到的人体框位置
++left	否	int	人体框左坐标
++top	否	int	人体框顶坐标
++width	否	int	人体框宽度
++height	否	int	人体框高度
+ID	否	int	人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续
person_count	否	object	进出区域的人流统计
+in	否	int	当前帧进入区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全进入area区域框时，该画面帧的in计数1；如要计算某一段时间内进入区域的累计人数，可基于连续帧图片的返回结果计算得到
+out	否	int	当前帧离开区域的瞬时人数，一般情况下，当人体头肩检测框刚好完全离开area区域框时，该画面帧的out计数1；如要计算某一段时间内离开区域的累计人数，可基于连续帧图片的返回结果计算得到
image	否	string	结果图，含跟踪框和统计值（渲染jpg图片byte内容的base64编码，得到后先做base64解码再以字节流形式imdecode）

### 渲染结果图说明

画面里刚出现的人体头肩检测框都是红色，被跟踪锁定之后会变成其他颜色（颜色随机，不同颜色没有特定规律），模型根据同颜色框的运动轨迹来判断进出移动方向；人体被跟踪锁定后，检测框上方会出现人体的ID编号，ID的取值逻辑为：每个case从1开始，不同人体向上递增但不一定连续。

### 人流量统计-动态版 返回示例

未检测到任何人：

```
{
  "person_num":0,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人，无轨迹，无人进出区域：

```
{
  "person_num":2,
  "person_info":[]
  "person_count":
  {
    "in":0,
    "out":0
  }
}
```

检测到2个人和2条轨迹，1人离开区域：

```
{
  "person_num":2,
  "person_info":
  [
    {
      "ID":3
      "location":
      {
        "left": 100,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
    {
      "ID": 5
      "location":
      {
        "left": 400,
        "top": 200,
        "width": 200,
        "height": 400,
      }
    }
  ]
  "person_count":
  {
    "in":0,
    "out":1
  }
}
```

## 手部关键点识别

### 🔗 手部关键点识别

对于输入的一张图片（可正常解码，且长宽比适宜），检测图片中的手部，输出每只手的坐标框、21个骨节点坐标信息。

当前接口主要适用于图片中单个手部的情况，图片中同时存在多个手部时，识别效果可能欠佳。

```

public void sample(AipBodyAnalysis client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地路径
    String image = "test.jpg";
    JSONObject res = client.handAnalysis(image, options);
    System.out.println(res.toString(2));

    // 参数为二进制数组
    byte[] file = readFile("test.jpg");
    res = client.handAnalysis(file, options);
    System.out.println(res.toString(2));
}

```

#### 手部关键点识别 请求参数详情

参数名称	是否必填	类型	说明
image	是	mixed	本地图片路径或者图片二进制数据

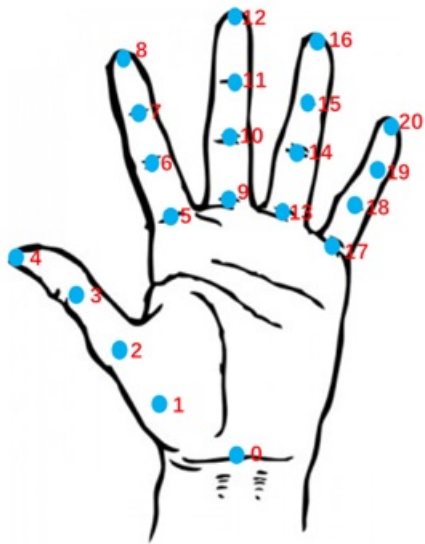
#### 手部关键点识别 返回数据参数详情

字段	是否必填	类型	说明
hand_num	是	uint32	检测到的人手数目
hand_info	否	object[]	手部关节信息
+location	否	object	手部坐标信息
++left	否	int	手部区域离左边界的距离
++top	否	int	手部区域离上边界的距离
++width	否	int	手部区域的宽度
++height	否	int	手部区域的高度
+hand_parts	是	object	手部骨节点信息，包含21个关键点
++0	是	object	关键点的标签，0-20
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数
.....	.....	.....	.....
++20	是	object	第20号关键点
+++x	是	uint32	x坐标
+++y	是	uint32	y坐标
+++score	是	float	置信度分数

#### 手部关键点识别 返回示例

返回关键点对应位置示意图：





```
{
  "log_id": "46532731~~~~32348525916",
  "hand_num": 1,
  "hand_info": [
    {
      "hand_parts": {
        "0": {
          "y": 707,
          "x": 829,
          "score": 0.81601244211197
        },
        "1": {
          "y": 620,
          "x": 873,
          "score": 0.6850221157074
        },
        "2": {
          "y": 533,
          "x": 916,
          "score": 0.75532239675522
        },
        "3": {
          "y": 467,
          "x": 938,
          "score": 0.79282909631729
        },
        "4": {
          "y": 380,
          "x": 993,
          "score": 0.88114696741104
        },
        "5": {
          "y": 424,
          "x": 829,
          "score": 0.90723776817322
        },
        "6": {
          "y": 358,
          "x": 862,
          "score": 0.62833058834076
        },
        "7": {
          "y": 391,
          "x": 916,
```



```
"score": 0.77525424957275
},
"8": {
  "y": 456,
  "x": 927,
  "score": 0.87964057922363
},
"9": {
  "y": 402,
  "x": 775,
  "score": 0.91389346122742
},
"10": {
  "y": 325,
  "x": 786,
  "score": 0.85947573184967
},
"11": {
  "y": 249,
  "x": 818,
  "score": 0.88918441534042
},
"12": {
  "y": 151,
  "x": 840,
  "score": 0.85277211666107
},
"13": {
  "y": 424,
  "x": 720,
  "score": 0.81941932439804
},
"14": {
  "y": 347,
  "x": 720,
  "score": 0.83918035030365
},
"15": {
  "y": 282,
  "x": 720,
  "score": 0.84135395288467
},
"16": {
  "y": 194,
  "x": 731,
  "score": 0.85115605592728
},
"17": {
  "y": 489,
  "x": 687,
  "score": 0.84704375267029
},
"18": {
  "y": 434,
  "x": 644,
  "score": 0.88782823085785
},
"19": {
  "y": 391,
  "x": 622,
  "score": 0.86267304420471
},
"20": {
```

```

    "z": {
      "y": 325,
      "x": 567,
      "score": 0.91110396385193
    }
  }, ~~~~
  "location": {
    "height": 556,
    "width": 426,
    "top": 151,
    "score": 17.495880126953,
    "left": 567
  }
}
]
}

```

## 错误码

### 错误返回格式

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。

### 错误码信息

SDK本地检测参数返回的错误码：

error_code	error_msg	备注
SDK100	image size error	图片大小超限，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，建议长宽比3：1以内，图片请求格式支持：PNG、JPG、BMP
SDK101	image length error	图片边长不符合要求，最短边至少50px，最长边最大4096px，建议长宽比3：1以内
SDK102	read image file error	读取图片文件错误
SDK108	connection or read data time out	连接超时或读取数据超时，请检查本地网络设置、文件读取设置
SDK109	unsupported image format	不支持的图片格式，当前支持以下几类图片：PNG、JPG、BMP

### 服务端返回的错误码

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（860337848）或 <a href="#">提交工单</a> 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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队。
3	Unsupported openapi method	调用的API不存在，请检查请求URL后重新尝试，一般为URL中有非英文字符，如“-”，可手动输入重试
4	Open api request limit	集群超限额，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交</a>

4	reached	<a href="#">工单</a> 联系技术支持团队。
6	No permission to access data	无权限访问该用户数据，创建应用时未勾选相关接口，请登录百度云控制台，找到对应的应用，编辑应用，勾选上相关接口，然后重试调用
13	Get service token failed	获取token失败
14	IAM Certification failed	IAM 鉴权失败，建议参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
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，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
18	Open api qps request limit reached	QPS超限额，未上线计费的接口，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
19	Open api total request limit reached	请求总量超限额，可通过QQ群（860337848）联系群管、 <a href="#">提交工单</a> 申请提升限额
100	Invalid parameter	无效的access_token参数，token拉取失败，可以参考 <a href="#">Access Token获取</a> 重新获取
110	Access token invalid or no longer valid	access_token无效，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
111	Access token expired	access token过期，token有效期为30天，注意需要定期更换，也可以每次请求都拉取新token
282000	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（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216100	invalid param	请求中包含非法参数，请检查后重新尝试
216101	not enough param	缺少必须的参数，请检查参数是否有遗漏
216102	service not support	请求了不支持的服务，请检查调用的url
216103	param too long	请求中某些参数过长，请检查后重新尝试
216110	appid not exist	appid不存在，请重新核对信息是否为后台应用列表中的appid
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，现阶段支持的图片格式为：PNG、JPG、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
216203	image size error	上传的图片base64编码有误，请校验base64编码方式，并重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
216634	detect error	检测错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
282003	missing parameters: {参数名}	请求参数缺失
282005	batch processing error	处理批量任务时发生部分或全部错误，请根据具体错误码排查
282006	batch task limit reached	批量任务处理数量超出限制，请将任务数量减少到10或10以下
282114	url size error	URL长度超过1024字节或为0
282808	request id: xxxxx not exist	request id xxxxx 不存在

282809	result type error	返回结果请求错误（不属于excel或json）
282810	image recognize error	图像识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（860337848）或 <a href="#">提交工单</a> 联系技术支持团队
283300	Invalid argument	入参格式有误，可检查下图片编码、代码格式是否有误

## 人像分割-证件照版（邀测）

### 接口描述

针对自拍类单人图片，先基于人脸检测、人体关键点定位，裁剪出符合证件照场景的人像图片（头肩上半身），对裁剪后的图片进行发丝级精细化分割；接口返回裁剪后的原图、分割结果灰度图、合成的黑底效果图，并输出人脸框坐标、关键点坐标、人像姿态（正脸/侧脸/正肩/侧肩等）等辅助信息，便于筛选结果、二次开发。

注：邀测的接口，暂未封装服务端SDK，只能通过API调用。

分割效果示意图：



### 请求说明

#### 请求示例

HTTP 方法：POST

请求URL：https://aip.baidubce.com/rest/2.0/image-classify/v1/body\_seg\_photo

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	说明
image	是	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。图片的base64编码不包含图片头，如(data:image/jpg;base64,)，支持图片格式：jpg、bmp、png，最短边至少50px，最长边最大4096px。 <b>异常情况</b> ：当图片中的人体数目 >=2时，或人像处于倒立状态，会报错

[返回说明](#)

### 返回参数

字段	是否必选	类型	说明
log_id	是	unit64	请求唯一的log id，用于问题定位
seg_info	是	object	人像精细化分割结果
+scoremap	是	string	分割后人像前景灰度图，归一到0-255，不用进行二次处理，直接解码保存图片即可。Base64编码后的灰度图文件
+demo	是	string	合成的证件照效果图（黑底）
+cut	是	string	按照证件照头肩比例裁剪后的原图
face_location	是	object	人脸框位置信息
+height	是	int32	人脸框的高度
+left	是	int32	人脸框离左边界的距离
+top	是	int32	人脸框离上边界的距离
+width	是	int32	人脸框的宽度
+score	是	float	人脸框的置信度分数，取值0-1
pose_info	是	object	人体姿态信息，关键点的坐标
+facepose	是	int32	人脸姿态，一共5种类别，1：水平正脸，2：倾斜正脸，3：水平侧脸，4：倾斜侧脸，-1：未知姿态
+bodypose	是	int32	肩膀姿态，一共7种类别，1：水平正肩，2：水平微侧肩，3：水平侧肩，4：倾斜正肩，5：倾斜微侧肩，6：倾斜侧肩，-1：未知姿态
+body_parts	是	object[]	人体关键点坐标信息，包含头肩部位的9个关键点
++left_ear	是	object	左耳
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++right_ear	是	object	右耳
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++left_eye	是	object	左眼
+++x	是	int32	x坐标

+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++right_eye	是	object	右眼
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++++nose_tip	是	object	鼻尖
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++left_mouth	是	object	左嘴角
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++right_mouth	是	object	右嘴角
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++left_shoulder	是	object	左肩
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++right_shoulder	是	object	右肩
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1
++neck	是	object	脖子
+++x	是	int32	x坐标
+++y	是	int32	y坐标
+++score	是	float	关键点置信度分数，取值0-1

### 返回示例

```
{
  log_id: 123,
  seg_info:{
    cut:"xxxx"
    scoremap:"yyyy"
```

```
scoremap: xxx
demo:"xxx"
},
face_location:{
  score:0.9999945,
  top:116,
  left:80,
  width:173,
  height:226
},
pose_info:{
  body_parts:[
    {
      left_ear:{
        score:0.8760631,
        x:80,
        y:202
      }
    },
    {
      right_ear:{
        score:0.64241165,
        x:252,
        y:206
      }
    },
    {
      left_eye:{
        score:0.95858794,
        x:125,
        y:214
      }
    },
    {
      right_eye:{
        score:0.9648065,
        x:199,
        y:216
      }
    },
    {
      nose_tip:{
        score:0.88654065,
        x:160,
        y:265
      }
    },
    {
      left_mouth:{
        score:0.92905945,
        x:135,
        y:300
      }
    },
    {
      right_mouth:{
        score:0.96665746,
        x:189,
        y:301
      }
    },
    {
      right_shoulder:{
```

```
    score: 0.25711855,  
    x:344,  
    y:432  
  }  
},  
{  
  left_shoulder:{  
    score:0.4777191,  
    x:10,  
    y:431  
  }  
},  
{  
  neck:{  
    score:0.7303297,  
    x:173,  
    y:369  
  }  
}  
],  
facepose:1,  
bodypose:2  
}  
}
```