

ALIBABA CLOUD

# 阿里云

视觉智能开放平台  
目标检测

文档版本：20220517

 阿里云

## 法律声明

阿里云提醒您在使用或阅读本文档之前仔细阅读、充分理解本法律声明各条款的内容。如果您阅读或使用本文档，您的阅读或使用行为将被视为对本声明全部内容的认可。

1. 您应当通过阿里云网站或阿里云提供的其他授权通道下载、获取本文档，且仅能用于自身的合法合规的业务活动。本文档的内容视为阿里云的保密信息，您应当严格遵守保密义务；未经阿里云事先书面同意，您不得向任何第三方披露本手册内容或提供给任何第三方使用。
2. 未经阿里云事先书面许可，任何单位、公司或个人不得擅自摘抄、翻译、复制本文档内容的部分或全部，不得以任何方式或途径进行传播和宣传。
3. 由于产品版本升级、调整或其他原因，本文档内容有可能变更。阿里云保留在没有任何通知或者提示下对本文档的内容进行修改的权利，并在阿里云授权通道中不时发布更新后的用户文档。您应当实时关注用户文档的版本变更并通过阿里云授权渠道下载、获取最新版的用户文档。
4. 本文档仅作为用户使用阿里云产品及服务的参考性指引，阿里云以产品及服务的“现状”、“有缺陷”和“当前功能”的状态提供本文档。阿里云在现有技术的基础上尽最大努力提供相应的介绍及操作指引，但阿里云在此明确声明对本文档内容的准确性、完整性、适用性、可靠性等不作任何明示或暗示的保证。任何单位、公司或个人因为下载、使用或信赖本文档而发生任何差错或经济损失的，阿里云不承担任何法律责任。在任何情况下，阿里云均不对任何间接性、后果性、惩戒性、偶然性、特殊性或刑罚性的损害，包括用户使用或信赖本文档而遭受的利润损失，承担责任（即使阿里云已被告知该等损失的可能性）。
5. 阿里云网站上所有内容，包括但不限于著作、产品、图片、档案、资讯、资料、网站架构、网站画面的安排、网页设计，均由阿里云和/或其关联公司依法拥有其知识产权，包括但不限于商标权、专利权、著作权、商业秘密等。非经阿里云和/或其关联公司书面同意，任何人不得擅自使用、修改、复制、公开传播、改变、散布、发行或公开发表阿里云网站、产品程序或内容。此外，未经阿里云事先书面同意，任何人不得为了任何营销、广告、促销或其他目的使用、公布或复制阿里云的名称（包括但不限于单独为或以组合形式包含“阿里云”、“Aliyun”、“万网”等阿里云和/或其关联公司品牌，上述品牌的附属标志及图案或任何类似公司名称、商号、商标、产品或服务名称、域名、图案标示、标志、标识或通过特定描述使第三方能够识别阿里云和/或其关联公司）。
6. 如若发现本文档存在任何错误，请与阿里云取得直接联系。

# 通用约定

格式	说明	样例
 危险	该类警示信息将导致系统重大变更甚至故障，或者导致人身伤害等结果。	 危险 重置操作将丢失用户配置数据。
 警告	该类警示信息可能会导致系统重大变更甚至故障，或者导致人身伤害等结果。	 警告 重启操作将导致业务中断，恢复业务时间约十分钟。
 注意	用于警示信息、补充说明等，是用户必须了解的内容。	 注意 权重设置为0，该服务器不会再接受新请求。
 说明	用于补充说明、最佳实践、窍门等，不是用户必须了解的内容。	 说明 您也可以通过按Ctrl+A选中全部文件。
>	多级菜单递进。	单击设置> 网络> 设置网络类型。
<b>粗体</b>	表示按键、菜单、页面名称等UI元素。	在结果确认页面，单击 <b>确定</b> 。
Courier字体	命令或代码。	执行 <code>cd /d C:/window</code> 命令，进入Windows系统文件夹。
斜体	表示参数、变量。	<code>bae log list --instanceid</code> <i>Instance_ID</i>
[ ] 或者 [a b]	表示可选项，至多选择一个。	<code>ipconfig [-all -t]</code>
{ } 或者 {a b}	表示必选项，至多选择一个。	<code>switch {active stand}</code>

# 目录

1.目标检测介绍	05
2.计费介绍	07
3.车辆检测	15
3.1. 机动车检测	15
3.2. 车辆部件识别	17
3.3. 车险图片分类	22
3.4. 车辆仪表盘识别	27
3.5. 车辆损伤识别	32
3.6. 车辆拥堵检测	40
3.7. 车辆违停检测	46
4.通用检测	55
4.1. 主体检测	55
4.2. 白底图检测	57
4.3. 透明图检测	59
4.4. 物体检测	61
4.5. IPC图像目标检测	69
4.6. IPC视频目标检测	72
4.7. 着装检测	76
4.8. 猫鼠识别	81
5.查询异步任务结果	86
6.公共信息	89
6.1. 请求结构	89
6.2. 公共请求参数	89
6.3. 公共响应参数	90
7.常见错误码	92

# 1.目标检测介绍

目标检测技术基于阿里云计算机视觉与深度学习技术，对通用或指定的目标进行检测、识别、打标。目标检测技术广泛应用于安防、交通等行业内图像场景的目标检测与跟踪。

## 服务开通

请单击[立即开通](#)在对应产品页面开通该服务，具体开通方式请参见[新手指南](#)。

## 能力介绍

目前阿里云视觉智能开放平台上线的目标检测能力包括：

类别	能力	说明
通用检测	主体检测	检测图像中的内容主体，返回该主体的区域位置/坐标信息。
	白底图检测	检测图片背景是否为白底。
	透明图检测	检测图片背景是否透明。
	物体检测	检测输入图像中的物体。
	IPC图像目标检测	可以检测到输入图像中的目标物体，例如人、车辆、宠物等。
	IPC视频目标检测	可以检测到输入视频中的目标物体，例如人、车辆、宠物等。
车辆检测	车辆损伤识别	针对常见小汽车车型，识别车辆外观受损部件及损伤类型，可识别数十种车辆部件，五大类外观损伤（刮擦、凹陷、开裂、褶皱、穿洞）。
	车辆部件识别	检测图片中车辆部件的位置以及名称。
	车辆仪表盘识别	识别车辆仪表盘上故障灯等信息。
	机动车检测	检测图像中的机动车主体，返回该机动车主体的区域位置/坐标信息。
	车险图片分类	对输入的车险图片进行分类。
	车辆拥堵检测	根据图片中的车辆，判断是否发生拥堵。
	车辆违停检测	可通过检测图片中目标区域内是否有车辆停放。
车辆维修	生成车辆维修方案	识别车辆受损照片，并根据受损情况生成车辆维修方案。
	查询车辆维修方案及费用	根据生成的维修方案，查询车辆具体的维修方案及维修费用。

## 应用场景

目标检测应用场景如下：

- 电商详情页装修

检测图像中商品主体的坐标信息，将其放置在商品详情页中适当的位置上，以完成详情页智能装修布局与智能设计。

- 路况分析

检测图像中道路、卡口的车辆位置和数量信息，分析道路交通状况，为交通疏导提供参考信息。

更多产品动态，请您持续关注[阿里云视觉智能开放平台](#)。

## 2. 计费介绍

本文主要介绍目标检测收费能力的计费方式及报价。其余未收费能力当前还处于公测阶段，可免费使用。

### 物体检测

物体检测能力支持按量付费和预付费资源包方式进行计费。

#### • 按量付费

调用量	按量付费
无限制	0.002元/次

#### 注意

- 如果您未购买资源包，系统将会按照按量付费的价格计费，请留意并及时支付账单，避免因欠费等原因影响您的业务。欠费24小时后会冻结调用该能力，支付欠费账单后可恢复使用。
- 调用失败不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

#### • 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您可根据实际情况按需[购买资源包](#)。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	25,000次	0.2点/次	50元	50元	首次享0.01元购买
10,000点	50,000次		100元	80元	无
50,000点	250,000次		500元	350元	无
500,000点	2,500,000次		5000元	3000元	无
5,000,000点	25,000,000次		50,000元	25,000元	无
20,000,000点	100,000,000次		200,000元	100,000元	无

#### 注意

- 资源包不可跨类目使用。例如，使用物体检测和高清人体分割两个能力，需要同时购买物体检测和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

### 车辆违停检测

车辆违停检测能力支持按量付费和预付费资源包方式进行计费。

#### ● 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 100,000	0.002元/次
100,000 < 调用量 (次/月) ≤ 5,000,000	0.0018元/次
5,000,000 < 调用量 (次/月) ≤ 30,000,000	0.0016元/次
调用量 (次/月) > 30,000,000	0.00136元/次

#### 🔔 注意

- 如果您未购买资源包，系统将会按照按量付费的价格计费，请留意并及时支付账单，避免因欠费等原因影响您的业务。欠费24小时后会冻结调用该能力，支付欠费账单后可恢复使用。
- 调用失败不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

#### ● 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您可根据实际情况按需[购买资源包](#)。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	25,000次	0.2点/次	50元	50元	首次享0.01元购买
10,000点	50,000次		100元	80元	无
50,000点	250,000次		500元	350元	无
500,000点	2,500,000次		5000元	3000元	无
5,000,000点	25,000,000次		50,000元	25,000元	无
20,000,000点	100,000,000次		200,000元	100,000元	无

#### 🔔 注意

- 资源包不可跨类目使用。例如，使用物体检测和高清人体分割两个能力，需要同时购买物体检测和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## 车辆拥堵检测

车辆拥堵检测能力支持按量付费和预付费资源包方式进行计费。

### ● 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 100,000	0.002元/次
100,000 < 调用量 (次/月) ≤ 5,000,000	0.0018元/次
5,000,000 < 调用量 (次/月) ≤ 30,000,000	0.0016元/次
调用量 (次/月) > 30,000,000	0.00136元/次

 注意

### ● 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您根据实际情况按需购买资源包。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	25,000次	0.2点/次	50元	50元	首次享0.01元购买
10,000点	50,000次		100元	80元	无
50,000点	250,000次		500元	350元	无
500,000点	2,500,000次		5000元	3000元	无
5,000,000点	25,000,000次		50,000元	25,000元	无
20,000,000点	100,000,000次		200,000元	100,000元	无

 注意

- 资源包不可跨类目使用。例如，使用物体检测和高清人体分割两个能力，需要同时购买物体检测和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## 白底图检测

白底图检测能力支持按量付费和预付费资源包方式进行计费。

### ● 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 100,000	0.002元/次

调用量	价格
100,000 < 调用量 (次/月) ≤ 5,000,000	0.0018元/次
5,000,000 < 调用量 (次/月) ≤ 30,000,000	0.0016元/次
调用量 (次/月) > 30,000,000	0.00136元/次

 注意

### ● 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照**按量付费**进行计费。您可根据实际情况按需**购买资源包**。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	25,000次	0.2点/次	50元	50元	首次享0.01元购买
10,000点	50,000次		100元	80元	无
50,000点	250,000次		500元	350元	无
500,000点	2,500,000次		5000元	3000元	无
5,000,000点	25,000,000次		50,000元	25,000元	无
20,000,000点	100,000,000次		200,000元	100,000元	无

 注意

- 资源包不可跨类目使用。例如，使用物体检测和高清人体分割两个能力，需要同时购买物体检测和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## IPC视频目标检测

IPC视频目标检测能力支持**按量付费**方式进行计费，价格如下所示。

规格	输入分辨率	分辨率取值范围	价格
SD	分辨率≤854×480像素	Max (宽, 高) ≤854像素, 且 Min (宽, 高) ≤480像素	0.0069元/分钟
HD	854×480 < 分辨率≤1280×720像素	854像素 < Max (宽, 高) ≤1280像素, 且 Min (宽, 高) ≤720像素; 或 480像素 < Min (宽, 高) ≤720像素, 且 Max (宽, 高) ≤1280像素	0.0138元/分钟

规格	输入分辨率	分辨率取值范围	价格
2K	1280×720 < 分辨率 ≤ 2560×1440 像素	1280像素 < Max (宽, 高) ≤ 2560像素, 且 Min (宽, 高) ≤ 1440像素; 或 720像素 < Min (宽, 高) ≤ 1440像素, 且 Max (宽, 高) ≤ 2560像素	0.0276元/分钟
4K	分辨率 > 2560×1440 像素	Max (宽, 高) > 2560像素, 且 Min (宽, 高) > 1440像素	0.0828元/分钟

#### 注意

- 对于输入视频图像最长边和最短边均需要满足对应规格的分辨率要求。例如，分辨率为1080×760像素的视频图像会被系统判断为2K规格。系统将根据输入图像分辨率自动判断规格，按照SD、HD、2K或4K各规格进行计费。
- 按照视频输入时长进行计费，时长最小计量单位为秒，时长不足1秒按照1秒进行计费。
- 该能力为异步能力，调用失败不计费，通过RequestId查询结果不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

## IPC图像目标检测

IPC图像目标检测能力支持按量付费和预付费资源包方式进行计费。

### 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 300万	0.00046元/次
300万 < 调用量 (次/月) ≤ 1500万	0.000368元/次
调用量 (次/月) > 1500万	0.000322元/次

#### 注意

- 如果您未购买资源包，系统将会按照按量付费的价格计费，请留意并及时支付账单，避免因欠费等原因影响您的业务。欠费24小时后会冻结调用该能力，支付欠费账单后可恢复使用。
- 调用失败不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

### 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您可根据实际情况按需[购买资源包](#)。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	108,696次		50元	50元	首次享0.01元购买

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
10,000点	217,391次	0.046点/次	100元	80元	无
50,000点	1,086,957次		500元	350元	无
500,000点	10,869,565次		5000元	3000元	无
5,000,000点	108,695,652次		50,000元	25,000元	无
20,000,000点	434,782,609次		200,000元	100,000元	无

#### 注意

- 资源包不可跨类目使用。例如，使用人脸核身和高清人体分割两个能力，需要同时购买人脸核身和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## 猫鼠识别

猫鼠识别能力支持按量付费和预付费资源包方式进行计费。

### ● 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 300万	0.002元/次
300万 < 调用量 (次/月) ≤ 1500万	0.0016元/次
调用量 (次/月) > 1500万	0.0014元/次

#### 注意

- 如果您未购买资源包，系统将会按照按量付费的价格计费，请留意并及时支付账单，避免因欠费等原因影响您的业务。欠费24小时后会冻结调用该能力，支付欠费账单后可恢复使用。
- 调用失败不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

### ● 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您可根据实际情况按需[购买资源包](#)。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	25,000次		50元	50元	首次享0.01元购买

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
10,000点	50,000次	0.2点/次	100元	80元	无
50,000点	250,000次		500元	350元	无
500,000点	2,500,000次		5000元	3000元	无
5,000,000点	25,000,000次		50,000元	25,000元	无
20,000,000点	100,000,000次		200,000元	100,000元	无

#### 注意

- 资源包不可跨类目使用。例如，使用物体检测和高清人体分割两个能力，需要同时购买物体检测和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## 着装检测

着装检测能力支持按量付费和预付费资源包方式进行计费。

### ● 按量付费

调用量	价格
0 < 调用量 (次/月) ≤ 300万	0.005元/次
300万 < 调用量 (次/月) ≤ 1500万	0.004元/次
调用量 (次/月) > 1500万	0.0035元/次

#### 注意

- 如果您未购买资源包，系统将会按照按量付费的价格计费，请留意并及时支付账单，避免因欠费等原因影响您的业务。欠费24小时后会冻结调用该能力，支付欠费账单后可恢复使用。
- 调用失败不计费。
- 价格费用计算详情请参见[费用计算及购买说明](#)。

### ● 预付费资源包

阿里云视觉智能开放平台提供首次购买5000点规格享全单0.01元的优惠活动。资源包支持叠加购买，当资源包点数耗尽或过期后，将按照按量付费进行计费。您可根据实际情况按需[购买资源包](#)。资源包有效期1年，价格如下所示。

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
5000点	10,000次		50元	50元	首次享0.01元购买

资源包规格	可调用次数	抵扣系数	标准价格	优惠价格	优惠政策
10,000点	20,000次	0.5点/次	100元	80元	无
50,000点	100,000次		500元	350元	无
500,000点	1,000,000次		5000元	3000元	无
5,000,000点	10,000,000次		50,000元	25,000元	无
20,000,000点	40,000,000次		200,000元	100,000元	无

#### 注意

- 资源包不可跨类目使用。例如，使用人脸核身和高清人体分割两个能力，需要同时购买人脸核身和高清人体分割两个资源包。
- 价格费用计算详情请参见[费用计算及购买说明](#)。
- 如果因业务需求购买更多QPS，请[提交工单](#)或通过[咨询服务](#)联系我们。

## 咨询服务

如果您有任何购买问题需要咨询阿里云视觉智能开放平台，欢迎各位企业用户、开发商、服务商或者开发者通过钉钉搜索群号23109592，加入阿里云视觉智能开放平台咨询群，与平台取得联系。

## 3. 车辆检测

### 3.1. 机动车检测

本文介绍机动车检测DetectVehicle的语法及示例。

#### 功能描述

**机动车检测**能力可以检测图像中的机动车主体，返回该机动车主体的区域位置，坐标信息。

#### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

#### 输入限制

- 图像格式：JPEG、JPG、PNG、BMP。
- 图像大小：不超过3 MB。
- 图像分辨率：不限制图像分辨率，但图像分辨率过高容易导致请求超时。
- URL地址中不能包含中文字符。

#### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

#### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectVehicle	系统规定参数。取值： <b>DetectVehicle</b> 。
ImageUrl	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectVehicle/DetectVehicle1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

#### 返回数据

名称	类型	示例值	描述
RequestId	String	77BB1EB9-1707-4423-8391-CDAE6D29F9D5	请求ID。
Data	Object		返回的结果数据内容。

名称	类型	示例值	描述
DetectObjectInfoList	Array of DetectObjectInfo		机动车检测框的集合。
Type	String	vehicle	检测框对应的物体，值为 <code>vehicle</code> 代表机动车。
Boxes	Array of Integer	0	检测框坐标，格式为 <code>[left, top, right, bottom]</code> 。例如 <code>[877, 95, 1021, 177]</code> 。
Score	Float	0.897	检测框的分数，范围为0~1。
Id	Integer	1	检测框的ID。
Width	Integer	612	图像的宽度。
Height	Integer	436	图像的高度。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DetectVehicle
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectVehicle/DetectVehicle1.jpg
&公共请求参数
```

### 正常返回示例

`XML` 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectVehicleResponse>
  <RequestId>8A38BAB7-0B53-531E-86F7-70A7F0A57BF8</RequestId>
  <Data>
    <DetectObjectInfoList>
      <Score>0.94</Score>
      <Type>vehicle</Type>
      <Id>0</Id>
      <Boxes>33</Boxes>
      <Boxes>141</Boxes>
      <Boxes>759</Boxes>
      <Boxes>451</Boxes>
    </DetectObjectInfoList>
    <Height>533</Height>
    <Width>948</Width>
  </Data>
</DetectVehicleResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "8A38BAB7-0B53-531E-86F7-70A7F0A57BF8",
  "Data" : {
    "DetectObjectInfoList" : {
      "Score" : 0.94,
      "Type" : "vehicle",
      "Id" : 0,
      "Boxes" : [ 33, 141, 759, 451 ]
    },
    "Height" : 533,
    "Width" : 948
  }
}
```

## 错误码

关于机动车检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会系统自动清理删除。

## 3.2. 车辆部件识别

本文为您介绍车辆部件识别RecognizeVehicleParts的语法及示例。

## 功能描述

**车辆部件识别**能力用于检测图片中车辆部件的位置以及名称。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG。
- 图像大小：不超过10 MB。
- 图像分辨率：不限制图像分辨率，但图像分辨率过高容易导致请求超时。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RecognizeVehicleParts	系统规定参数。取值： <b>RecognizeVehicleParts</b> 。
ImageUrl	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/RecognizeVehicleParts/RecognizeVehicleParts1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	35771730-0FC5-4E7D-8C2F-2FADDFECAA39	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		各个子元素的识别结果。

名称	类型	示例值	描述
Type	String	hood	部件类型。具体请参见下表Type说明。
Boxes	Array of Integer	336,1,635,57	部件位置框信息, [左上角点x坐标, 左上角点y坐标, 右上角点x坐标, 右上角点y坐标]。
Score	Float	0.9929494857788086	部件类型对应的概率值。
OriginShapes	Array of Integer	480, 640	输入图像的原始尺寸[高度, 宽度]。

## Type说明

序号	中文名称	英文名称	序号	中文名称	英文名称
1	前保险杠	front_bumper	20	左尾灯	left_tail_light
2	后保险杠	rear_bumper	21	右尾灯	right_tail_light
3	左前大灯	left_light	22	左雾灯	left_foglight
4	右前大灯	right_light	23	右雾灯	right_foglight
5	中网	grille	24	格栅	grates
6	前机盖	hood	25	右前车窗	right_front_window
7	左前门	left_front_door	26	左前车窗	left_front_window
8	右前门	right_front_door	27	右后车窗	right_rear_window
9	左后门	left_rear_door	28	左后车窗	left_rear_window
10	右后门	right_rear_door	29	右底大边	right_doorsill

序号	中文名称	英文名称	序号	中文名称	英文名称
11	左前翼子板	left_front_wing	30	左底大边	left_doorsill
12	右前翼子板	right_front_wing	31	左前轮胎	left_front_tire
13	左后翼子板	left_rear_wing	32	左后轮胎	left_rear_tire
14	右后翼子板	right_rear_wing	33	右前轮胎	right_front_tire
15	后机盖	decklid	34	右后轮胎	right_rear_tire
16	前挡风玻璃	front_windshield	35	门把手	doorknob
17	后挡风玻璃	rear_windshield	36	轮眉	WheelBrow
18	左后视镜	left_mirror	37	车标	logo
19	右后视镜	right_mirror	38	车牌	plate

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=RecognizeVehicleParts
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/RecognizeVehicleParts/RecognizeVehicleParts1.jpg
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<RecognizeVehiclePartsResponse>
  <RequestId>66B7D01C-F28B-56EE-849B-ADA8B7219C52</RequestId>
  <Data>
    <OriginShapes>533</OriginShapes>
    <OriginShapes>948</OriginShapes>
    <Elements>
      <Score>0.9977518916130066</Score>
      <Type>hood</Type>
      <Boxes>5</Boxes>
      <Boxes>0</Boxes>
      <Boxes>588</Boxes>
      <Boxes>299</Boxes>
    </Elements>
    <Elements>
      <Score>0.9883555173873901</Score>
      <Type>left_light</Type>
      <Boxes>230</Boxes>
      <Boxes>182</Boxes>
      <Boxes>667</Boxes>
      <Boxes>372</Boxes>
    </Elements>
    <Elements>
      <Score>0.9499000310897827</Score>
      <Type>front_bumper</Type>
      <Boxes>98</Boxes>
      <Boxes>279</Boxes>
      <Boxes>713</Boxes>
      <Boxes>527</Boxes>
    </Elements>
    <Elements>
      <Score>0.9013092517852783</Score>
      <Type>left_front_wing</Type>
      <Boxes>543</Boxes>
      <Boxes>8</Boxes>
      <Boxes>723</Boxes>
      <Boxes>421</Boxes>
    </Elements>
    <Elements>
      <Score>0.786267101764679</Score>
      <Type>grille</Type>
      <Boxes>3</Boxes>
      <Boxes>277</Boxes>
      <Boxes>208</Boxes>
      <Boxes>533</Boxes>
    </Elements>
  </Data>
</RecognizeVehiclePartsResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "66B7D01C-F28B-56EE-849B-ADA8B7219C52",
  "Data" : {
    "OriginShapes" : [ 533, 948 ],
    "Elements" : [ {
      "Score" : 0.9977518916130066,
      "Type" : "hood",
      "Boxes" : [ 5, 0, 588, 299 ]
    }, {
      "Score" : 0.9883555173873901,
      "Type" : "left_light",
      "Boxes" : [ 230, 182, 667, 372 ]
    }, {
      "Score" : 0.9499000310897827,
      "Type" : "front_bumper",
      "Boxes" : [ 98, 279, 713, 527 ]
    }, {
      "Score" : 0.9013092517852783,
      "Type" : "left_front_wing",
      "Boxes" : [ 543, 8, 723, 421 ]
    }, {
      "Score" : 0.786267101764679,
      "Type" : "grille",
      "Boxes" : [ 3, 277, 208, 533 ]
    }
  ]
}
```

## 错误码

关于车辆部件识别的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

## 3.3. 车险图片分类

本文为您介绍车险图片分类ClassifyVehicleInsurance的语法及示例。

### 功能描述

[车险图片分类](#)能力对输入的车险图片进行分类。

### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG。
- 图像大小：不超过10 MB。
- 图像分辨率：不限制图像分辨率，但图像分辨率过高容易导致请求超时。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	ClassifyVehicleInsurance	系统规定参数。取值： <b>ClassifyVehicleInsurance</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domestic/objectdet/ClassifyVehicleInsurance/ClassifyVehicleInsurance1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	878EB16C-87C6-4C8B-897D-F68036F8AC97	请求ID。
Data	Object		返回的结果数据内容。
Labels	Array of Label		输出分类结果。
Score	Float	0.0023	分类结果对应的概率值。

名称	类型	示例值	描述
Name	String	others	label名称。枚举类型。枚举值包括： <ul style="list-style-type: none"> <li>• others：其他</li> <li>• detail：细节图</li> <li>• component：汽车部件图</li> <li>• vin：汽车vin码</li> <li>• people：人物</li> <li>• motor：发动机</li> <li>• semi-car：半车图</li> <li>• panoramic：全车图</li> <li>• license：行驶证</li> <li>• CT-scan：CT-扫描</li> <li>• truck：卡车</li> <li>• disassembly：拆解件</li> <li>• scene：现场图</li> </ul>
Threshold	Float	0	阈值。根据阈值控制输出label，取值范围0~1。

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=ClassifyVehicleInsurance
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/ClassifyVehicleInsurance/ClassifyVehicleInsurance1.jpg
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<ClassifyVehicleInsuranceResponse>
  <RequestId>878EB16C-87C6-4C8B-897D-F68036F8AC97</RequestId>
  <Data>
    <Labels>
      <Score>0.0023</Score>
      <Name>others</Name>
    </Labels>
    <Labels>
      <Score>0</Score>
      <Name>detail</Name>
    </Labels>
    <Labels>
      <Score>0.0001</Score>
      <Name>component</Name>
    </Labels>
```

```
<Labels>
  <Score>0</Score>
  <Name>vin</Name>
</Labels>
<Labels>
  <Score>0.0001</Score>
  <Name>people</Name>
</Labels>
<Labels>
  <Score>0</Score>
  <Name>motor</Name>
</Labels>
<Labels>
  <Score>0</Score>
  <Name>semi-car</Name>
</Labels>
<Labels>
  <Score>0.2215</Score>
  <Name>panoramic</Name>
</Labels>
<Labels>
  <Score>0</Score>
  <Name>license</Name>
</Labels>
<Labels>
  <Score>0</Score>
  <Name>CT-scan</Name>
</Labels>
<Labels>
  <Score>0.0001</Score>
  <Name>truck</Name>
</Labels>
<Labels>
  <Score>0.0001</Score>
  <Name>disassembly</Name>
</Labels>
<Labels>
  <Score>0.7759</Score>
  <Name>scene</Name>
</Labels>
<Threshold>0</Threshold>
</Data>
</ClassifyVehicleInsuranceResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "878EB16C-87C6-4C8B-897D-F68036F8AC97",
  "Data" : {
    "Labels" : [ {
      "Score" : 0.0023,
      "Name" : "others"
    }, {
      "Score" : 0,
      "Name" : "detail"
    }, {
      "Score" : 1.0E-4,
      "Name" : "component"
    }, {
      "Score" : 0,
      "Name" : "vin"
    }, {
      "Score" : 1.0E-4,
      "Name" : "people"
    }, {
      "Score" : 0,
      "Name" : "motor"
    }, {
      "Score" : 0,
      "Name" : "semi-car"
    }, {
      "Score" : 0.2215,
      "Name" : "panoramic"
    }, {
      "Score" : 0,
      "Name" : "license"
    }, {
      "Score" : 0,
      "Name" : "CT-scan"
    }, {
      "Score" : 1.0E-4,
      "Name" : "truck"
    }, {
      "Score" : 1.0E-4,
      "Name" : "disassembly"
    }, {
      "Score" : 0.7759,
      "Name" : "scene"
    } ],
    "Threshold" : 0
  }
}
```

## 错误码

关于车险图片分类的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会由系统自动清理删除。

# 3.4. 车辆仪表盘识别

本文为您介绍车辆仪表盘识别RecognizeVehicleDashboard的语法及示例。

## 功能描述

**车辆仪表盘识别**能力用于识别仪表盘上故障灯等信息。当前可以识别的故障灯类别包括：1（TA01 门窗未关闭）、2（TA02 安全带未系）、3（TA03 霜冻提示）、4（TG01 定速巡航）、5（TG02 雾灯打开）、6（TG03 远光打开）、7（TG04 坡道辅助）、8（TG05 玻璃加热）、9（TG06 制动提示灯）、10（FZ01 燃油不足）、11（FZ02 胎压过低）、12（FZ03 清洗液不足）、13（FZ04 电瓶缺电）、14（FW01 需保养）、15（FW02 水温异常）、16（FW03 机油不足）、17（FW04 冷却液不足）、18（FW05 刹车片磨损）、19（FW06 安全气囊故障）、20（FW07 发动机故障）、21（QT00 其他）。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG、PNG。
- 图像大小：不超过10 MB。
- 图像分辨率：不限制图像分辨率，但图像分辨率过高容易导致请求超时。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RecognizeVehicleDashboard	系统规定参数。取值： <b>RecognizeVehicleDashboard</b> 。

名称	类型	是否必选	示例值	描述
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domestic/objectdet/RecognizeVehicleDashboard/RecognizeVehicleDashboard1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	3BDA7297-2515-4C9C-99EA-C5950B906BCF	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		识别结果。
Boxes	Array of Float	"692.4900512695312", "423.0277099609375", "709.9772338867188", "436.11529541015625"	识别故障灯位置。格式为[x1,y1,x2,y2]，x1和y1为矩形框左上角点坐标，x2和y2为右下角点坐标。
Score	Float	0.9990643858909607	置信度。

名称	类型	示例值	描述
Label	String	13	<p>识别故障灯类别。例如13表示电瓶缺电。具体取值如下：</p> <ul style="list-style-type: none"> <li>• 1: TA01 门窗未关闭</li> <li>• 2: TA02 安全带未系</li> <li>• 3: TA03 霜冻提示</li> <li>• 4: TG01 定速巡航</li> <li>• 5: TG02 雾灯打开</li> <li>• 6: TG03 远光打开</li> <li>• 7: TG04 坡道辅助</li> <li>• 8: TG05 玻璃加热</li> <li>• 9: TG06 制动提示灯</li> <li>• 10: FZ01 燃油不足</li> <li>• 11: FZ02 胎压过低</li> <li>• 12: FZ03 清洗液不足</li> <li>• 13: FZ04 电瓶缺电</li> <li>• 14: FW01 需保养</li> <li>• 15: FW02 水温异常</li> <li>• 16: FW03 机油不足</li> <li>• 17: FW04 冷却液不足</li> <li>• 18: FW05 刹车片磨损</li> <li>• 19: FW06 安全气囊故障</li> <li>• 20: FW07 发动机故障</li> <li>• 21: QT00 其他</li> </ul>
ClassName	String	FZ04 电瓶缺电	识别故障灯名称，unicode编码。

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=RecognizeVehicleDashboard
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/Recogni
zeVehicleDashboard/RecognizeVehicleDashboard1.jpg
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<RecognizeVehicleDashboardResponse>
  <RequestId>3BDA7297-2515-4C9C-99EA-C5950B906BCF</RequestId>
  <Data>
    <Elements>
```

```
<Score>0.9990643858909607</Score>
<ClassName>FZ04 电瓶缺电</ClassName>
<Label>13</Label>
<Boxes>692.4900512695312</Boxes>
<Boxes>423.0277099609375</Boxes>
<Boxes>709.9772338867188</Boxes>
<Boxes>436.11529541015625</Boxes>
</Elements>
<Elements>
  <Score>0.9948421120643616</Score>
  <ClassName>FW07 发动机故障</ClassName>
  <Label>20</Label>
  <Boxes>262.644287109375</Boxes>
  <Boxes>419.08294677734375</Boxes>
  <Boxes>282.2294921875</Boxes>
  <Boxes>432.89495849609375</Boxes>
</Elements>
<Elements>
  <Score>0.9942097663879395</Score>
  <ClassName>TG06 制动提示灯</ClassName>
  <Label>9</Label>
  <Boxes>280.72662353515625</Boxes>
  <Boxes>405.62225341796875</Boxes>
  <Boxes>301.68280029296875</Boxes>
  <Boxes>421.94085693359375</Boxes>
</Elements>
<Elements>
  <Score>0.9878326654434204</Score>
  <ClassName>TA02 安全带未系</ClassName>
  <Label>2</Label>
  <Boxes>690.0284423828125</Boxes>
  <Boxes>374.8208923339844</Boxes>
  <Boxes>705.72021484375</Boxes>
  <Boxes>394.5099182128906</Boxes>
</Elements>
<Elements>
  <Score>0.957288920879364</Score>
  <ClassName>FZ01 燃油不足</ClassName>
  <Label>10</Label>
  <Boxes>716.0230712890625</Boxes>
  <Boxes>426.26654052734375</Boxes>
  <Boxes>730.5159912109375</Boxes>
  <Boxes>445.77960205078125</Boxes>
</Elements>
<Elements>
  <Score>0.9438304901123047</Score>
  <ClassName>TG04 坡道辅助</ClassName>
  <Label>7</Label>
  <Boxes>688.2109375</Boxes>
  <Boxes>353.2281188964844</Boxes>
  <Boxes>708.463134765625</Boxes>
  <Boxes>368.4597473144531</Boxes>
</Elements>
</Data>
```

```
</RecognizeVehicleDashboardResponse>
```

### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "3BDA7297-2515-4C9C-99EA-C5950B906BCF",
  "Data" : {
    "Elements" : [ {
      "Score" : 0.9990643858909607,
      "ClassName" : "FZ04 电瓶缺电",
      "Label" : 13,
      "Boxes" : [ 692.4900512695312, 423.0277099609375, 709.9772338867188, 436.115295410156
25 ]
    }, {
      "Score" : 0.9948421120643616,
      "ClassName" : "FW07 发动机故障",
      "Label" : 20,
      "Boxes" : [ 262.644287109375, 419.08294677734375, 282.2294921875, 432.89495849609375
]
    }, {
      "Score" : 0.9942097663879395,
      "ClassName" : "TG06 制动提示灯",
      "Label" : 9,
      "Boxes" : [ 280.72662353515625, 405.62225341796875, 301.68280029296875, 421.940856933
59375 ]
    }, {
      "Score" : 0.9878326654434204,
      "ClassName" : "TA02 安全带未系",
      "Label" : 2,
      "Boxes" : [ 690.0284423828125, 374.8208923339844, 705.72021484375, 394.5099182128906
]
    }, {
      "Score" : 0.957288920879364,
      "ClassName" : "FZ01 燃油不足",
      "Label" : 10,
      "Boxes" : [ 716.0230712890625, 426.26654052734375, 730.5159912109375, 445.77960205078
125 ]
    }, {
      "Score" : 0.9438304901123047,
      "ClassName" : "TG04 坡道辅助",
      "Label" : 7,
      "Boxes" : [ 688.2109375, 353.2281188964844, 708.463134765625, 368.4597473144531 ]
    } ]
  }
}
```

## 错误码

关于车辆仪表盘识别的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会系统自动清理删除。

## 3.5. 车辆损伤识别

本文介绍车辆损伤识别RecognizeVehicleDamage的语法及示例。

### 功能描述

**车辆损伤识别**能力可检测图像中车辆损伤的位置以及类型。

### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

### 输入限制

- 图像格式：JPEG、JPG。
- 图像大小：不超过10M。
- 图像分辨率：建议小于1280×720像素，图像分辨率过高容易导致请求超时。
- URL地址中不能包含中文字符。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	RecognizeVehicleDamage	系统规定参数。取值： <b>RecognizeVehicleDamage</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/RecognizeVehicleDamage/RecognizeVehicleDamage1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

### 返回数据

名称	类型	示例值	描述
RequestId	String	91FCB447-807D-4839-A5F0-E0A79D6B33C6	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		各个子元素的识别结果。
Type	String	1	<p>损伤类型ID。具体类型如下所示。</p> <ul style="list-style-type: none"> <li>• 1: 轻微刮擦</li> <li>• 2: 重度刮擦</li> <li>• 3: 轻度变形</li> <li>• 4: 中度变形</li> <li>• 5: 重度变形</li> <li>• 6: crack破损孔洞</li> <li>• 7: 翼子板和大灯缝隙</li> <li>• 8: 翼子板保险杠缝隙</li> <li>• 9: 大灯轻微刮擦</li> <li>• 10: 大灯重度刮擦</li> <li>• 11: 大灯破损</li> <li>• 12: 后视镜轻微刮擦</li> <li>• 13: 后视镜玻璃破损</li> <li>• 14: 后视镜脱落</li> <li>• 15: 挡风玻璃破损</li> </ul>
Scores	Array of Float	0.683465	对应所有15种损伤类型的损伤概率。
Boxes	Array of Integer	[173,178,277,259]	部件位置框信息，形式为[左上角点x坐标，右下角点y坐标，右上角点x坐标，左下角点y坐标]。
Score	Float	0.683465	损伤类型对应的概率值。

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=RecognizeVehicleDamage
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/Recogni
zeVehicleDamage/RecognizeVehicleDamage1.jpg
&<公共请求参数>
```

### 正常返回示例











```
"Elements" : [ {
  "Score" : 0.683465,
  "Type" : 1,
  "Scores" : [ 0.683465, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 173, 178, 277, 259 ]
}, {
  "Score" : 0.672104,
  "Type" : 1,
  "Scores" : [ 0.672104, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 269, 137, 432, 206 ]
}, {
  "Score" : 0.559934,
  "Type" : 1,
  "Scores" : [ 0.559934, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 788, 158, 884, 189 ]
}, {
  "Score" : 0.549272,
  "Type" : 1,
  "Scores" : [ 0.549272, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 667, 213, 846, 242 ]
}, {
  "Score" : 0.52864,
  "Type" : 1,
  "Scores" : [ 0.52864, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 487, 143, 670, 222 ]
}, {
  "Score" : 0.509307,
  "Type" : 1,
  "Scores" : [ 0.509307, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 692, 256, 761, 285 ]
}, {
  "Score" : 0.323798,
  "Type" : 1,
  "Scores" : [ 0.323798, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 684, 221, 827, 296 ]
}, {
  "Score" : 0.634771,
  "Type" : 2,
  "Scores" : [ 0, 0.634771, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 305, 351, 375, 391 ]
}, {
  "Score" : 0.61175,
  "Type" : 2,
  "Scores" : [ 0, 0.61175, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 562, 364, 621, 409 ]
}, {
  "Score" : 0.603638,
  "Type" : 2,
  "Scores" : [ 0, 0.603638, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 379, 302, 462, 352 ]
}, {
  "Score" : 0.578584,
  "Type" : 2,
  "Scores" : [ 0, 0.578584, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 ],
  "Boxes" : [ 481, 310, 546, 367 ]
}
```

```
        } ]
    }
}
```

## 错误码

关于车辆损伤识别的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 3.6. 车辆拥堵检测

本文为您介绍车辆拥堵检测DetectVehicleCongestion的语法及示例。

## 功能描述

**车辆拥堵检测**能力根据图片中的车辆，判断是否发生拥堵。

## 应用场景

- 停车场拥堵检测：通过停车场摄像头间隔时间内捕捉的两张图像，检测车辆以及相对位置，判断是否发生拥堵。
- 道路拥堵检测：通过道路监控间隔时间内捕捉的前后图像，检测车辆以及相对位置，判断是否发生拥堵。

## 特色优势

检测数据量少：无需较大数据量的视频，仅通过检测两张图片，即可判断是否发生拥堵。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG、PNG、BMP。
- 图像大小：不超过500 KB。
- 图像分辨率：大于20×20像素，小于800×800像素，长宽比小于等于2。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectVehicleCongestion	要执行的操作。取值： <b>DetectVehicleCongestion</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domestic/objectdet/DetectVehicleCongestion/DetectVehicleCongestion1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。
RoadRegions	Array	是		多个区域。格式为 <b>RoadRegion_1, RoadRegion_2, ..., RoadRegion_N</b> 。
RoadRegion	Array	是		任意四边形的四个顶点的坐标X和Y，格式为 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> 。
Point	Object	是		四边形的顶点坐标。
X	Long	是	400	坐标点的X值。
Y	Long	是	400	坐标点的Y值。
PreRegionIntersectFeatures	Array	否		前一帧画面与各个指定区域相交的车辆检测框对应的车辆特征。
Features	Array of String	否	AEAMQQA AAAA**** ABhC	Base64编码的字符串。

## 返回数据

名称	类型	示例值	描述
RequestId	String	4FC381BB-04F2-50F4-B54B-593042BCF3C6	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		车辆检测框的集合。

名称	类型	示例值	描述
Boxes	Array of Boxes		检测框坐标，格式为 [left, top, right, bottom]。
Left	Long	341	矩形区域的左上角横轴坐标。
Top	Long	434	矩形区域的左上角纵轴坐标。
Right	Long	589	矩形区域的右下角横轴坐标。
Bottom	Long	576	矩形区域的右下角纵轴坐标。
Score	Float	0.962890625	检测框的置信度，范围为 [0,1]。
TypeName	String	vehicle	检测框对应的物体。取值仅为 vehicle。
Id	Long	1	检测框ID。
RegionIntersect Features	Array of RegionIntersect Features		与各个指定区域相交的车辆检测框对应的车辆特征。
Features	Array of String	AEAMQAAAA****ABhC	此Base64编码用于匹配两帧之间的车辆是否为同一车辆。
RegionIntersect Matched	Array of RegionIntersect Matched		判断当前帧的RegionIntersectFeatures与PreRegionIntersectFeatures中哪一个特征是匹配的。

名称	类型	示例值	描述
Ids	Array of Long	[-1,-1,-1]	<p>当两帧之间的两个车辆是匹配时，说明该车辆在两帧之间的时间内存在滞留或拥堵现象。</p> <p><b>说明</b> -1代表没有匹配的特征，如果匹配将显示对应检测框ID。例如，车辆检测框ID为 [1,2,3]，RegionIntersectMatched显示为 [-1,-1,-1]，则表示三个区域均没有匹配，RegionIntersectMatched显示为 [-1,2,3]，则表示区域1未匹配，区域2和3匹配。</p>
RegionIntersects	Array of RegionIntersects		<p>与各个指定区域相交的车辆检测框对应的车辆特征。</p> <p>例如，返回结果为 {"Ids":1,3}, {"Ids":2,4}，其中{"Ids":1,3}是与区域1相交的车辆检测框的ID，{"Ids":2,4}是与区域2相交的车辆检测框的ID。当车辆出现在给定区域内，说明存在车辆占道的情况。</p> <p>除了RegionIntersects里包含的ID的检测框外，其它检测框为区域外的检测框。</p>
Ids	Array of Long	[1,2,3]	<p>与各个指定区域相交的车辆检测框的多个ID。</p> <p>例如，{"Ids":1,3}是与相应区域相交的车辆检测框的IDs。</p>

## 示例

### 请求示例

```

http(s)://[Endpoint]/?Action=DetectVehicleICongestion
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectVehicleICongestion/DetectVehicleICongestion1.jpg
&RoadRegions=[{"RoadRegion":{"Point":{"X":400,"Y":400}}}]
&PreRegionIntersectFeatures=[{"Features":["AEAMQQAAAA****ABhC"]}
&公共请求参数
    
```

### 正常返回示例

XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<DetectVehicleICongestionResponse>
  <RequestId>4FC381BB-04F2-50F4-B54B-593042BCF3C6</RequestId>
  <Data>
    
```

```
<Data>
  <RegionIntersectFeatures>
    <Features>AOAQQQBA***FRD</Features>
    <Features>AEAMQQA***KRC</Features>
    <Features>AEAMQQA***BhC</Features>
  </RegionIntersectFeatures>
  <RegionIntersects>
    <Ids>1</Ids>
    <Ids>2</Ids>
    <Ids>3</Ids>
  </RegionIntersects>
  <Elements>
    <Score>0.962890625</Score>
    <TypeName>vehicle</TypeName>
    <Id>0</Id>
    <Boxes>
      <Left>341</Left>
      <Top>434</Top>
      <Right>589</Right>
      <Bottom>576</Bottom>
    </Boxes>
  </Elements>
  <Elements>
    <Score>0.9619140625</Score>
    <TypeName>vehicle</TypeName>
    <Id>1</Id>
    <Boxes>
      <Left>304</Left>
      <Top>228</Top>
      <Right>525</Right>
      <Bottom>440</Bottom>
    </Boxes>
  </Elements>
  <Elements>
    <Score>0.83203125</Score>
    <TypeName>vehicle</TypeName>
    <Id>2</Id>
    <Boxes>
      <Left>0</Left>
      <Top>24</Top>
      <Right>68</Right>
      <Bottom>106</Bottom>
    </Boxes>
  </Elements>
  <Elements>
    <Score>0.53125</Score>
    <TypeName>vehicle</TypeName>
    <Id>3</Id>
    <Boxes>
      <Left>0</Left>
      <Top>0</Top>
      <Right>58</Right>
      <Bottom>38</Bottom>
    </Boxes>
  </Elements>
</Data>
```

```
<RegionIntersectMatched>
  <Ids>-1</Ids>
  <Ids>-1</Ids>
  <Ids>-1</Ids>
</RegionIntersectMatched>
</Data>
</DetectVehicleICongestionResponse>
```

## JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "4FC381BB-04F2-50F4-B54B-593042BCF3C6",
  "Data" : {
    "RegionIntersectFeatures" : {
      "Features" : [ "AOAQQQBA****FRD", "AEAMQQAA****KRC", "AEAMQQAA****BhC" ]
    },
    "RegionIntersects" : {
      "Ids" : [ 1, 2, 3 ]
    },
    "Elements" : [ {
      "Score" : 0.962890625,
      "TypeName" : "vehicle",
      "Id" : 0,
      "Boxes" : {
        "Left" : 341,
        "Top" : 434,
        "Right" : 589,
        "Bottom" : 576
      }
    }, {
      "Score" : 0.9619140625,
      "TypeName" : "vehicle",
      "Id" : 1,
      "Boxes" : {
        "Left" : 304,
        "Top" : 228,
        "Right" : 525,
        "Bottom" : 440
      }
    }, {
      "Score" : 0.83203125,
      "TypeName" : "vehicle",
      "Id" : 2,
      "Boxes" : {
        "Left" : 0,
        "Top" : 24,
        "Right" : 68,
        "Bottom" : 106
      }
    }, {
      "Score" : 0.53125,
```

```
"TypeName" : "vehicle",
  "Id" : 3,
  "Boxes" : {
    "Left" : 0,
    "Top" : 0,
    "Right" : 58,
    "Bottom" : 38
  }
} ],
"RegionIntersectMatched" : {
  "Ids" : [ -1, -1, -1 ]
}
}
```

## 错误码

关于车辆拥堵检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 3.7. 车辆违停检测

本文为您介绍车辆违停检测DetectVehicleIllegalParking的语法及示例。

## 功能描述

**车辆违停检测**能力可通过检测图片中目标区域内是否有车辆停放。

## 应用场景

- 消防通道占用：通过检测消防通道区域是否有车辆，可以判断是否有车辆占用了消防通道。
- 出入口违停：检测停车场、商场、小区、园区出入口指定区域是否有车辆违停。

## 特色优势

- 方便改造部署：仅提供图像即可检测判断。
- 支持多种角度：支持通过多种角度检测是否存在车辆违停。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG、PNG、BMP。
- 图像大小：不超过500 KB。

- 图像分辨率：大于20×20像素，小于800×800像素。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectVehicleIllegalParking	要执行的操作。取值： <b>DetectVehicleIllegalParking</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domestic/objectdet/DetectVehicleIllegalParking/DetectVehicleIllegalParking2.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。
RoadRegions	Array	是		多个区域。格式为 <b>RoadRegion_1, RoadRegion_2, ..., RoadRegion_N</b> 。
RoadRegion	Array	是		任意四边形的四个顶点的坐标X和Y，格式为 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> 。
Point	Object	是		四边形的顶点坐标。
X	Long	是	400	坐标点的X值。
Y	Long	是	400	坐标点的Y值。

## 返回数据

名称	类型	示例值	描述
RequestId	String	DB882EDD-991A-5A0C-A19B-CC7C4BA65E35	请求ID。
Data	Object		返回的结果数据内容。

名称	类型	示例值	描述
Elements	Array of Element		车辆检测框的集合。
Boxes	Array of Boxes		检测框坐标，格式为 <code>[left, top, right, bottom]</code> 。
Left	Long	413	矩形区域的左上角横轴坐标。
Top	Long	138	矩形区域的左上角纵轴坐标。
Right	Long	499	矩形区域的右下角横轴坐标。
Bottom	Long	268	矩形区域的右下角纵轴坐标。
Score	Float	0.9599609375	检测框的置信度，范围为 <code>[0,1]</code> 。
TypeName	String	vehicle	检测框对应的物体。取值仅为 <code>vehicle</code> 。
Id	Long	1	检测框ID。
RegionIntersects	Array of RegionIntersects		<p>与各个给定区域相交的车辆检测框对应的车辆特征。</p> <p>例如，返回结果为 <code>{"Ids": [1,3], "Ids": [2,4]}</code>，其中 <code>{"Ids": 1,3}</code>是与区域1相交的车辆检测框的ID，<code>{"Ids": 2,4}</code>是与区域2相交的车辆检测框的ID。当车辆出现在给定区域内，说明存在车辆占道的情况。</p> <p>除了RegionIntersects里包含的ID的检测框外，其它的检测框为区域外的检测框。</p>
Ids	Array of Long	[1,2,3]	<p>与各个指定区域相交的车辆检测框的多个ID。</p> <p>例如，<code>{"Ids": 1,3}</code>是与相应区域相交的车辆检测框的IDs。</p>

## 示例

请求示例

```
http(s)://[Endpoint]/?Action=DetectVehicleIllegalParking
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectV
ehicleIllegalParking/DetectVehicleIllegalParking2.jpg
&RoadRegions=[{"RoadRegion":[{"Point":{"X":400,"Y":400}}]}]
&公共请求参数
```

## 正常返回示例

### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectVehicleIllegalParkingResponse>
  <RequestId>DB882EDD-991A-5A0C-A19B-CC7C4BA65E35</RequestId>
  <Data>
    <RegionIntersects>
      <Ids>1</Ids>
      <Ids>2</Ids>
      <Ids>3</Ids>
      <Ids>4</Ids>
      <Ids>5</Ids>
      <Ids>6</Ids>
      <Ids>7</Ids>
      <Ids>8</Ids>
      <Ids>10</Ids>
      <Ids>11</Ids>
    </RegionIntersects>
    <Elements>
      <Score>0.9599609375</Score>
      <TypeName>vehicle</TypeName>
      <Id>0</Id>
      <Boxes>
        <Left>413</Left>
        <Top>138</Top>
        <Right>499</Right>
        <Bottom>268</Bottom>
      </Boxes>
    </Elements>
    <Elements>
      <Score>0.95751953125</Score>
      <TypeName>vehicle</TypeName>
      <Id>1</Id>
      <Boxes>
        <Left>146</Left>
        <Top>121</Top>
        <Right>371</Right>
        <Bottom>284</Bottom>
      </Boxes>
    </Elements>
    <Elements>
      <Score>0.93310546875</Score>
      <TypeName>vehicle</TypeName>
      <Id>2</Id>
      <Boxes>
```

```
        <Left>54</Left>
        <Top>92</Top>
        <Right>164</Right>
        <Bottom>213</Bottom>
    </Boxes>
</Elements>
<Elements>
    <Score>0.9326171875</Score>
    <TypeName>vehicle</TypeName>
    <Id>3</Id>
    <Boxes>
        <Left>94</Left>
        <Top>141</Top>
        <Right>171</Right>
        <Bottom>232</Bottom>
    </Boxes>
</Elements>
<Elements>
    <Score>0.9150390625</Score>
    <TypeName>vehicle</TypeName>
    <Id>4</Id>
    <Boxes>
        <Left>303</Left>
        <Top>114</Top>
        <Right>430</Right>
        <Bottom>195</Bottom>
    </Boxes>
</Elements>
<Elements>
    <Score>0.82080078125</Score>
    <TypeName>vehicle</TypeName>
    <Id>5</Id>
    <Boxes>
        <Left>14</Left>
        <Top>134</Top>
        <Right>32</Right>
        <Bottom>154</Bottom>
    </Boxes>
</Elements>
<Elements>
    <Score>0.767578125</Score>
    <TypeName>vehicle</TypeName>
    <Id>6</Id>
    <Boxes>
        <Left>0</Left>
        <Top>132</Top>
        <Right>17</Right>
        <Bottom>151</Bottom>
    </Boxes>
</Elements>
<Elements>
    <Score>0.767578125</Score>
    <TypeName>vehicle</TypeName>
    <Id>7</Id>
```

```
<Boxes>
  <Left>37</Left>
  <Top>133</Top>
  <Right>56</Right>
  <Bottom>162</Bottom>
</Boxes>
</Elements>
<Elements>
  <Score>0.765625</Score>
  <TypeName>vehicle</TypeName>
  <Id>8</Id>
  <Boxes>
    <Left>27</Left>
    <Top>131</Top>
    <Right>55</Right>
    <Bottom>158</Bottom>
  </Boxes>
</Elements>
<Elements>
  <Score>0.75341796875</Score>
  <TypeName>vehicle</TypeName>
  <Id>9</Id>
  <Boxes>
    <Left>408</Left>
    <Top>14</Top>
    <Right>499</Right>
    <Bottom>143</Bottom>
  </Boxes>
</Elements>
<Elements>
  <Score>0.517578125</Score>
  <TypeName>vehicle</TypeName>
  <Id>10</Id>
  <Boxes>
    <Left>164</Left>
    <Top>130</Top>
    <Right>183</Right>
    <Bottom>144</Bottom>
  </Boxes>
</Elements>
<Elements>
  <Score>0.355712890625</Score>
  <TypeName>vehicle</TypeName>
  <Id>11</Id>
  <Boxes>
    <Left>1</Left>
    <Top>133</Top>
    <Right>31</Right>
    <Bottom>153</Bottom>
  </Boxes>
</Elements>
</Data>
</DetectVehicleIllegalParkingResponse>
```

## JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "DB882EDD-991A-5A0C-A19B-CC7C4BA65E35",
  "Data" : {
    "RegionIntersects" : {
      "Ids" : [ 1, 2, 3, 4, 5, 6, 7, 8, 10, 11 ]
    },
    "Elements" : [ {
      "Score" : 0.9599609375,
      "TypeName" : "vehicle",
      "Id" : 0,
      "Boxes" : {
        "Left" : 413,
        "Top" : 138,
        "Right" : 499,
        "Bottom" : 268
      }
    }, {
      "Score" : 0.95751953125,
      "TypeName" : "vehicle",
      "Id" : 1,
      "Boxes" : {
        "Left" : 146,
        "Top" : 121,
        "Right" : 371,
        "Bottom" : 284
      }
    }, {
      "Score" : 0.93310546875,
      "TypeName" : "vehicle",
      "Id" : 2,
      "Boxes" : {
        "Left" : 54,
        "Top" : 92,
        "Right" : 164,
        "Bottom" : 213
      }
    }, {
      "Score" : 0.9326171875,
      "TypeName" : "vehicle",
      "Id" : 3,
      "Boxes" : {
        "Left" : 94,
        "Top" : 141,
        "Right" : 171,
        "Bottom" : 232
      }
    }, {
      "Score" : 0.9150390625,
      "TypeName" : "vehicle",
      "Id" : 4,
```

```
"Boxes" : {
  "Left" : 303,
  "Top" : 114,
  "Right" : 430,
  "Bottom" : 195
}
}, {
  "Score" : 0.82080078125,
  "TypeName" : "vehicle",
  "Id" : 5,
  "Boxes" : {
    "Left" : 14,
    "Top" : 134,
    "Right" : 32,
    "Bottom" : 154
  }
}, {
  "Score" : 0.767578125,
  "TypeName" : "vehicle",
  "Id" : 6,
  "Boxes" : {
    "Left" : 0,
    "Top" : 132,
    "Right" : 17,
    "Bottom" : 151
  }
}, {
  "Score" : 0.767578125,
  "TypeName" : "vehicle",
  "Id" : 7,
  "Boxes" : {
    "Left" : 37,
    "Top" : 133,
    "Right" : 56,
    "Bottom" : 162
  }
}, {
  "Score" : 0.765625,
  "TypeName" : "vehicle",
  "Id" : 8,
  "Boxes" : {
    "Left" : 27,
    "Top" : 131,
    "Right" : 55,
    "Bottom" : 158
  }
}, {
  "Score" : 0.75341796875,
  "TypeName" : "vehicle",
  "Id" : 9,
  "Boxes" : {
    "Left" : 408,
    "Top" : 14,
    "Right" : 499,
    "Bottom" : 142
```

```
    "Bottom" : 143
  }
}, {
  "Score" : 0.517578125,
  "TypeName" : "vehicle",
  "Id" : 10,
  "Boxes" : {
    "Left" : 164,
    "Top" : 130,
    "Right" : 183,
    "Bottom" : 144
  }
}, {
  "Score" : 0.355712890625,
  "TypeName" : "vehicle",
  "Id" : 11,
  "Boxes" : {
    "Left" : 1,
    "Top" : 133,
    "Right" : 31,
    "Bottom" : 153
  }
} ]
}
```

## 错误码

关于车辆违停检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

## 4.通用检测

### 4.1. 主体检测

本文介绍主体检测DetectMainBody的语法及示例。

#### 功能描述

**主体检测**能力对输入图片进行检测，输出主体定位信息。

#### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

#### 输入限制

- 图像格式：PNG（不支持8位、16位和64位的PNG）、JPG、JPEG、BMP。
- 图像大小：不超过3 MB。
- 图像分辨率：小于1280×1280像素。
- URL地址中不能包含中文字符。

#### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

#### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectMainBody	系统规定参数。取值： <b>DetectMainBody</b> 。
ImageUrl	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectMainBody/DetectMainBody1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

#### 返回数据

名称	类型	示例值	描述
RequestId	String	2E59C333-5480-4231-A8AB-BEE1001EA7FE	请求ID。

名称	类型	示例值	描述
Data	Object		返回的结果数据内容。
Location	Object		图像定位信息。
Width	Integer	583	元素宽度。
Height	Integer	320	元素高度。
Y	Integer	20	元素y坐标。
X	Integer	28	元素x坐标。

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=DetectMainBody
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectM
ainBody/DetectMainBody1.jpg
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectMainBodyResponse>
  <RequestId>2E59C333-5480-4231-A8AB-BEE1001EA7FE</RequestId>
  <Data>
    <Location>
      <X>28</X>
      <Y>20</Y>
      <Height>320</Height>
      <Width>583</Width>
    </Location>
  </Data>
</DetectMainBodyResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "2E59C333-5480-4231-A8AB-BEE1001EA7FE",
  "Data" : {
    "Location" : {
      "X" : 28,
      "Y" : 20,
      "Height" : 320,
      "Width" : 583
    }
  }
}
```

## 错误码

关于主体检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会由系统自动清理删除。

## 4.2. 白底图检测

本文介绍白底图检测DetectWhiteBaseImage的语法及示例。

### 功能描述

[白底图检测](#)能力用于检测图片背景是否为白底图。

### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

### 输入限制

- 图像格式：JPG、JPEG、PNG。
- 图像大小：不超过3 MB。
- 图像分辨率：大于32×32像素，小于1280×1280像素。
- URL地址中不能包含中文字符。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectWhiteBaseImage	系统规定参数。取值： <b>DetectWhiteBaseImage</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectWhiteBaseImage/DetectWhiteBaseImage1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	7A7F9EEB-44C4-4592-9089-A6185B222B23	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		返回判断结果。
WhiteBase	Integer	0	是否白底图。 <ul style="list-style-type: none"> <li>0: 非白底</li> <li>1: 白底</li> </ul>

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=DetectWhiteBaseImage
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectWhiteBaseImage/DetectWhiteBaseImage1.jpg
&<公共请求参数>
```

### 正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectWhiteBaseImageResponse>
  <RequestId>7A7F9EEB-44C4-4592-9089-A6185B222B23</RequestId>
  <Data>
    <Elements>
      <WhiteBase>0</WhiteBase>
    </Elements>
  </Data>
</DetectWhiteBaseImageResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "7A7F9EEB-44C4-4592-9089-A6185B222B23",
  "Data" : {
    "Elements" : {
      "WhiteBase" : 0
    }
  }
}
```

## 错误码

关于白底图检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

## 4.3. 透明图检测

本文介绍透明图检测DetectTransparentImage的语法及示例。

### 功能描述

**透明图检测**能力用于检测图片背景是否为透明图。

### 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

### 输入限制

- 图像格式：PNG（不支持8位、16位、64位PNG）、JPEG、JPG、BMP、WEBP。
- 图像大小：不超过3 MB。

- 图像分辨率：大于32×32像素，小于1280×1280像素。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectTransparentImage	系统规定参数。取值： <b>DetectTransparentImage</b> 。
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectTransparentImage/DetectTransparentImage1.png	图像URL地址。URL地址中不能包含中文字符。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	3F3F335E-640C-4860-A85F-774726FF56A7	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		返回判断结果。
TransparentImage	Integer	0	是否透明图。 <ul style="list-style-type: none"> <li>● 0: 非透明图</li> <li>● 1: 透明图</li> </ul>

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=DetectTransparentImage
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectTransparentImage/DetectTransparentImage1.png
&<公共请求参数>
```

## 正常返回示例

### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectTransparentImageResponse>
  <RequestId>3F3F335E-640C-4860-A85F-774726FF56A7</RequestId>
  <Data>
    <Elements>
      <TransparentImage>0</TransparentImage>
    </Elements>
  </Data>
</DetectTransparentImageResponse>
```

### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "3F3F335E-640C-4860-A85F-774726FF56A7",
  "Data" : {
    "Elements" : {
      "TransparentImage" : 0
    }
  }
}
```

## 错误码

关于透明图检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 4.4. 物体检测

本文介绍物体检测Detect Object的语法及示例。

## 功能描述

**物体检测**能力可以检测输入图像中的物体。

示例图如下。



## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG、PNG、BMP。
- 图像大小：不超过3 MB。
- 图像分辨率：大于20×20像素，小于4096×4096像素。
- URL地址中不能包含中文字符。

## 计费说明

关于物体检测的计费方式及报价，请参见[计费介绍](#)。

 **说明** 下方调试接口为付费接口，如需免费体验调试请前往[体验中心](#)。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectObject	系统规定参数。取值： <b>DetectObject</b> 。

名称	类型	是否必选	示例值	描述
ImageURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domestic/objectdet/DetectObject/DetectObject1.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	6EF97B44-2763-4EAD-8737-FB9F5EE25FE2	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		检测框数据集合。
Type	String	chair	检测框对应的物体名称。具体类型如下所示。
Boxes	Array of Integer	[468, 238, 531, 299]	检测框坐标，格式为 [left, top, right, bottom]。
Score	Float	0.266	检测到物体的置信度，取值范围为0~1。
Width	Integer	533	图片的宽度。
Height	Integer	300	图片的高度。

Type类型包括：

human (人体)、sneakers (胶底运动鞋)、chair (椅子)、hat (帽子)、lamp (灯)、cabinet/shelf (橱柜/架子)、car (汽车)、glasses (眼镜)、picture/frame (照片/图画)、street lights (街灯)、helmet (头盔)、pillow (枕头)、glove (手套)、potted plant (盆栽植物)、flower (花)、monitor (显示屏)、plants pot/vase (花盆)、boots (靴子)、umbrella (伞)、boat (小船)、flag (旗帜)、speaker (扬声器/话筒)、trash bin/can (垃圾桶)、backpack (双肩背包)、sofa (沙发)、belt (腰带)、carpet (地毯)、coffee table (咖啡桌/茶几)、tie (领带)、bed (床)、traffic light (红绿灯)、necklace (项链)、mirror (镜子)、bicycle (自行车)、watch (手表)、horse (马)、traffic sign (交通标志)、stuffed animal (填充玩具动物)、motorbike/motorcycle (摩托车)、wild bird (鸟)、laptop (笔记本电脑)、cow (奶牛)、clock (时钟)、bus (公共汽车)、nightstand (床头柜)、sheep (绵羊)、traffic cone (锥形交通路标)、keyboard (键盘)、hockey stick (曲棍球球棍)、fan (电扇)、dog (狗)、blackboard/whiteboard (白板/黑板)、mouse (鼠标)、telephone (电话)、airplane (飞机)、skis (滑雪板)、soccer (英式足球)、combine with glove (棒球手套)、train (火车)、tent (帐篷)、sailboat (帆船)、kite (风筝)、computer box (计算机主机机箱)、elephant (大象)、stroller (折叠式婴儿车)、baseball bat (棒球棒)、skateboard (溜冰板)、surfboard (冲浪板)、cat (猫)、zebra (斑马)、sports car (跑车)、giraffe (长颈鹿)、radiator (散热器)、tennis racket (网球拍)、skating and skiing shoes (溜冰鞋)、baseball (棒球)、american football (美式橄榄球)、basketball (篮球)、printer (打印机)、fire hydrant (消防栓)、projector (投影仪)、fire extinguisher (灭火器)、tennis ball (网球)、frisbee (飞盘)、fire truck (消防车)、helicopter (直升飞机)、carriage (四轮马车)、bear (熊)、globe (地球仪)、volleyball (排球)。

## 示例

### 请求示例

```
http(s)://objectdet.cn-shanghai.aliyuncs.com/?Action=DetectObject
&ImageURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectO
bject/DetectObject1.jpg
&<公共请求参数>
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectObjectResponse>
  <RequestId>6EF97B44-2763-4EAD-8737-FB9F5EE25FE2</RequestId>
  <Data>
    <Height>300</Height>
    <Elements>
      <Score>0.266</Score>
      <Type>chair</Type>
      <Boxes>468</Boxes>
      <Boxes>238</Boxes>
      <Boxes>531</Boxes>
      <Boxes>299</Boxes>
    </Elements>
    <Elements>
      <Score>0.213</Score>
      <Type>chair</Type>
      <Boxes>452</Boxes>
      <Boxes>168</Boxes>
      <Boxes>531</Boxes>
```

```
<Boxes>233</Boxes>
</Elements>
<Elements>
  <Score>0.308</Score>
  <Type>picture/frame</Type>
  <Boxes>487</Boxes>
  <Boxes>44</Boxes>
  <Boxes>501</Boxes>
  <Boxes>96</Boxes>
</Elements>
<Elements>
  <Score>0.257</Score>
  <Type>picture/frame</Type>
  <Boxes>477</Boxes>
  <Boxes>67</Boxes>
  <Boxes>484</Boxes>
  <Boxes>98</Boxes>
</Elements>
<Elements>
  <Score>0.201</Score>
  <Type>picture/frame</Type>
  <Boxes>503</Boxes>
  <Boxes>26</Boxes>
  <Boxes>524</Boxes>
  <Boxes>85</Boxes>
</Elements>
<Elements>
  <Score>0.515</Score>
  <Type>pillow</Type>
  <Boxes>161</Boxes>
  <Boxes>141</Boxes>
  <Boxes>181</Boxes>
  <Boxes>158</Boxes>
</Elements>
<Elements>
  <Score>0.293</Score>
  <Type>pillow</Type>
  <Boxes>473</Boxes>
  <Boxes>143</Boxes>
  <Boxes>496</Boxes>
  <Boxes>157</Boxes>
</Elements>
<Elements>
  <Score>0.214</Score>
  <Type>pillow</Type>
  <Boxes>241</Boxes>
  <Boxes>145</Boxes>
  <Boxes>253</Boxes>
  <Boxes>166</Boxes>
</Elements>
<Elements>
  <Score>0.374</Score>
  <Type>potted plant</Type>
  <Boxes>449</Boxes>
```

```
<Boxes>97</Boxes>
<Boxes>477</Boxes>
<Boxes>140</Boxes>
</Elements>
<Elements>
  <Score>0.412</Score>
  <Type>sofa</Type>
  <Boxes>145</Boxes>
  <Boxes>135</Boxes>
  <Boxes>304</Boxes>
  <Boxes>202</Boxes>
</Elements>
<Elements>
  <Score>0.269</Score>
  <Type>sofa</Type>
  <Boxes>443</Boxes>
  <Boxes>136</Boxes>
  <Boxes>531</Boxes>
  <Boxes>190</Boxes>
</Elements>
<Elements>
  <Score>0.262</Score>
  <Type>sofa</Type>
  <Boxes>301</Boxes>
  <Boxes>136</Boxes>
  <Boxes>352</Boxes>
  <Boxes>168</Boxes>
</Elements>
<Elements>
  <Score>0.242</Score>
  <Type>sofa</Type>
  <Boxes>452</Boxes>
  <Boxes>167</Boxes>
  <Boxes>531</Boxes>
  <Boxes>230</Boxes>
</Elements>
<Elements>
  <Score>0.219</Score>
  <Type>carpet</Type>
  <Boxes>0</Boxes>
  <Boxes>153</Boxes>
  <Boxes>272</Boxes>
  <Boxes>293</Boxes>
</Elements>
<Elements>
  <Score>0.458</Score>
  <Type>coffee table</Type>
  <Boxes>171</Boxes>
  <Boxes>172</Boxes>
  <Boxes>214</Boxes>
  <Boxes>214</Boxes>
</Elements>
<Elements>
  <Score>0.358</Score>
```

```
<Type>coffee table</Type>
<Boxes>337</Boxes>
<Boxes>185</Boxes>
<Boxes>420</Boxes>
<Boxes>274</Boxes>
</Elements>
<Elements>
  <Score>0.285</Score>
  <Type>coffee table</Type>
  <Boxes>386</Boxes>
  <Boxes>159</Boxes>
  <Boxes>428</Boxes>
  <Boxes>192</Boxes>
</Elements>
<Elements>
  <Score>0.214</Score>
  <Type>mirror</Type>
  <Boxes>503</Boxes>
  <Boxes>26</Boxes>
  <Boxes>524</Boxes>
  <Boxes>85</Boxes>
</Elements>
<Width>533</Width>
</Data>
</DetectObjectResponse>
```

### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "6EF97B44-2763-4EAD-8737-FB9F5EE25FE2",
  "Data" : {
    "Height" : 300,
    "Elements" : [ {
      "Score" : 0.266,
      "Type" : "chair",
      "Boxes" : [ 468, 238, 531, 299 ]
    }, {
      "Score" : 0.213,
      "Type" : "chair",
      "Boxes" : [ 452, 168, 531, 233 ]
    }, {
      "Score" : 0.308,
      "Type" : "picture/frame",
      "Boxes" : [ 487, 44, 501, 96 ]
    }, {
      "Score" : 0.257,
      "Type" : "picture/frame",
      "Boxes" : [ 477, 67, 484, 98 ]
    }, {
      "Score" : 0.201,
      "Type" : "picture/frame",
      "Boxes" : [ 503, 26, 524, 85 ]
    }
  ]
}
```

```
}, {
  "Score" : 0.515,
  "Type" : "pillow",
  "Boxes" : [ 161, 141, 181, 158 ]
}, {
  "Score" : 0.293,
  "Type" : "pillow",
  "Boxes" : [ 473, 143, 496, 157 ]
}, {
  "Score" : 0.214,
  "Type" : "pillow",
  "Boxes" : [ 241, 145, 253, 166 ]
}, {
  "Score" : 0.374,
  "Type" : "potted plant",
  "Boxes" : [ 449, 97, 477, 140 ]
}, {
  "Score" : 0.412,
  "Type" : "sofa",
  "Boxes" : [ 145, 135, 304, 202 ]
}, {
  "Score" : 0.269,
  "Type" : "sofa",
  "Boxes" : [ 443, 136, 531, 190 ]
}, {
  "Score" : 0.262,
  "Type" : "sofa",
  "Boxes" : [ 301, 136, 352, 168 ]
}, {
  "Score" : 0.242,
  "Type" : "sofa",
  "Boxes" : [ 452, 167, 531, 230 ]
}, {
  "Score" : 0.219,
  "Type" : "carpet",
  "Boxes" : [ 0, 153, 272, 293 ]
}, {
  "Score" : 0.458,
  "Type" : "coffee table",
  "Boxes" : [ 171, 172, 214, 214 ]
}, {
  "Score" : 0.358,
  "Type" : "coffee table",
  "Boxes" : [ 337, 185, 420, 274 ]
}, {
  "Score" : 0.285,
  "Type" : "coffee table",
  "Boxes" : [ 386, 159, 428, 192 ]
}, {
  "Score" : 0.214,
  "Type" : "mirror",
  "Boxes" : [ 503, 26, 524, 85 ]
} ] ,
"Width" : 533
```

```
}  
}
```

## 错误码

关于物体检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 4.5. IPC图像目标检测

本文为您介绍IPC图像目标检测DetectIPCObject的语法及示例。

## 功能描述

**IPC图像目标检测**能力可以检测到输入图像中的目标物体，例如人、车辆、宠物等。

## 应用场景

- 智能看家：家中有人或宠物移动时可生成高精度视频，实现检测提醒功能。且可以关注宠物在家中的活动状况。
- 智能看店：可对门店起到安防作用，尤其是在夜间，可及时知道有人进店。
- 智能看厂：安全是厂区非常重要的事情，对于人员在岗以及人员入侵可以起到良好的监管作用。

## 特色优势

- 适用场景多：适应多种广泛的日常场景的目标检测，室内、室外、家居、店铺、商场、白天、夜间等。
- 检测范围广：可检测的目标大小范围广，适用于大目标检测，也适用于较远摄像头下的小目标检测。
- 检测效率高：高并发处理，处理时间短。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPEG、JPG、PNG、BMP。
- 图像大小：不超过2 MB。
- 图像分辨率：大于20×20像素，小于1024×1024像素。并非越高清晰度越高，推荐值：最大边为640像素。
- URL地址中不能包含中文字符。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectIPCObject	系统规定参数。取值： <b>DetectIPCObject</b> 。
ImageURL	String	是	https://viapi-test.oss-cn-shanghai.aliyuncs.com/objectdet/detect-ipc-xxxx.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	7AE23740-A3E5-5607-8E10-895DCBD4C260	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of Element		检测到的目标集合，如果没有目标则为空。
Type	String	DOG	检测框对应的物体名称，取值如下： <ul style="list-style-type: none"> <li>PERSON：人物</li> <li>VEHICLE：车辆</li> <li>CAT：猫</li> <li>DOG：狗</li> </ul>
Score	Float	0.7138671875	检测框的置信度，取值范围[0.0,1.0]。
Box	Array of Long	[10,150,1587,1193]	检测框坐标，单位像素，格式为[left, top, right, bottom]。
TargetRate	Float	0.8566723958333333	目标检测框的面积大小和图像大小的比例。取值范围[0.0,1.0]，1.0表示全屏。
Width	Long	1600	图像的宽度，单位像素。
Height	Long	1200	图像的高度，单位像素。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DetectIPCObject
&ImageURL=https://viapi-test.oss-cn-shanghai.aliyuncs.com/objectdet/detect-ipc-xxxx.jpg
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectIPCObjectResponse>
  <RequestId>7AE23740-A3E5-5607-8E10-895DCBD4C260</RequestId>
  <Data>
    <Height>1200</Height>
    <Elements>
      <TargetRate>0.2129958333333333</TargetRate>
      <Type>CAT</Type>
      <Score>0.85107421875</Score>
      <Box>7</Box>
      <Box>665</Box>
      <Box>783</Box>
      <Box>1192</Box>
    </Elements>
    <Elements>
      <TargetRate>0.8566723958333333</TargetRate>
      <Type>DOG</Type>
      <Score>0.7138671875</Score>
      <Box>10</Box>
      <Box>150</Box>
      <Box>1587</Box>
      <Box>1193</Box>
    </Elements>
    <Width>1600</Width>
  </Data>
</DetectIPCObjectResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "7AE23740-A3E5-5607-8E10-895DCBD4C260",
  "Data" : {
    "Height" : 1200,
    "Elements" : [ {
      "TargetRate" : 0.21299583333333333,
      "Type" : "CAT",
      "Score" : 0.85107421875,
      "Box" : [ 7, 665, 783, 1192 ]
    }, {
      "TargetRate" : 0.8566723958333333,
      "Type" : "DOG",
      "Score" : 0.7138671875,
      "Box" : [ 10, 150, 1587, 1193 ]
    } ],
    "Width" : 1600
  }
}
```

## 错误码

关于IPC图像目标检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 4.6. IPC视频目标检测

本文为您介绍IPC视频目标检测DetectVideoIPCObject的语法及示例。

## 功能描述

**IPC视频目标检测**能力可以检测到输入视频中的目标物体，例如人、车辆、宠物等。

## 应用场景

- 智能看家：家中有人或宠物移动时可生成高精度视频，实现检测提醒功能。且可以关注宠物在家中的活动状况。
- 智能看店：可对门店起到安防作用，尤其是在夜间，可及时知道有人进店。
- 智能看厂：安全是厂区非常重要的事情，对于人员在岗以及人员入侵可以起到良好的监管作用。

## 特色优势

- 适用场景多：适应多种广泛的日常场景的目标检测，室内、室外、家居、店铺、商场、白天、夜间等。
- 检测范围广：可检测的目标大小范围广，适用于大目标检测，也适用于较远摄像头下的小目标检测。

- 检测效率高：高并发处理，处理时间短。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 视频格式：H264、H265。
- 视频分辨率：大于20×20像素，小于4096×4096像素。
- 视频帧率：要求在35帧以内。
- 视频大小：不超过10 MB。
- 视频时长：不超过10秒。
- URL地址中不能包含中文字符。

## 计费说明

关于IPC视频目标检测的计费方式及报价，请参见[计费介绍](#)。

 **说明** 下方调试接口为付费接口，如需免费体验调试请前往[体验中心](#)。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectVideoIPCObject	系统规定参数。取值： <a href="#">DetectVideoIPCObject</a> 。
VideoURL	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectVideoIPCObject/DetectVideoIPCObject1.mp4	视频文件URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。
StartTimestamp	Long	否	1629086400	视频实际开始时间的时间戳。例如视频开始时间是2021年8月16日12:00，对应时间戳为1629086400。

名称	类型	是否必选	示例值	描述
CallbackOnlyHasObject	Boolean	否	true	<p>是否回调消息过滤。</p> <ul style="list-style-type: none"> <li>• true: 如果输入的视频中检测到有目标, 才会执行回调, 否则不执行。</li> <li>• false: 无论是否检测到有目标, 都执行回调。</li> </ul>

## 返回数据

名称	类型	示例值	描述
RequestId	String	35B11E1B-800C-4598-B5AA-577E3BDBD917	请求ID。
Data	Object		返回的结果数据内容。
Width	Long	1280	视频宽度分辨率, 单位像素。
Height	Long	720	视频高度分辨率, 单位像素。
Frames	Array of Frame		视频帧的集合, 未检测到目标的帧不列出。
Time	Long	6124533574	时间戳, 单位秒。
Elements	Array of elements		检测到的目标集合, 如果没有目标则为空。
Type	String	PERSON	<p>检测框对应的物体名称。取值如下:</p> <ul style="list-style-type: none"> <li>• PERSON: 人</li> <li>• VEHICLE: 车辆</li> <li>• PET: 宠物</li> </ul>
X	Long	289	左上角X轴坐标, 单位像素。
Y	Long	271	左上角Y轴坐标, 单位像素。
Width	Long	100	检测框宽度, 单位像素。
Height	Long	156	检测框高度, 单位像素。

名称	类型	示例值	描述
Score	Float	0.7812	检测框的置信度，取值范围为[0.0,1.0]。
Input File	String	oss://viapi-test/viapi-3.0domepic/objectdet/DetectVideoIPCObject/DetectVideoIPCObject4.mp4	输入文件信息。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DetectVideoIPCObject
&VideoURL=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectVideoIPCObject/DetectVideoIPCObject1.mp4
&StartTimestamp=1629086400
&CallbackOnlyHasObject=true
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectVideoIPCObjectResponse>
  <RequestId>35B11E1B-800C-4598-B5AA-577E3BDBD917</RequestId>
  <Data>
    <Width>1280</Width>
    <Height>720</Height>
    <Frames>
      <Time>6124533574</Time>
      <Elements>
        <Type>PERSON</Type>
        <X>289</X>
        <Y>271</Y>
        <Width>100</Width>
        <Height>156</Height>
        <Score>0.7812</Score>
      </Elements>
    </Frames>
    <InputFile>oss://viapi-test/viapi-3.0domepic/objectdet/DetectVideoIPCObject/DetectVideoIPCObject4.mp4</InputFile>
  </Data>
</DetectVideoIPCObjectResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "35B11E1B-800C-4598-B5AA-577E3BDBD917",
  "Data" : {
    "Width" : 1280,
    "Height" : 720,
    "Frames" : [ {
      "Time" : 6124533574,
      "Elements" : [ {
        "Type" : "PERSON",
        "X" : 289,
        "Y" : 271,
        "Width" : 100,
        "Height" : 156,
        "Score" : 0.7812
      } ]
    } ],
    "InputFile" : "oss://viapi-test/viapi-3.0domepic/objectdet/DetectVideoIPCObject/DetectVideoIPCObject4.mp4"
  }
}
```

## 查询任务结果

该接口为异步接口，当前并未返回真实的请求结果，您需要通过返回的RequestId调用GetAsyncJobResult接口来获取该接口的真实请求结果。详情请参见[查询异步任务结果](#)。

## 错误码

关于IPC视频目标检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

# 4.7. 着装检测

本文为您介绍着装检测DetectWorkwear的语法及示例。

## 功能描述

**着装检测**能力基于视觉AI算法、物联网、大数据分析技术，检测指定场景范围内的人员帽子、口罩、工作服等穿戴情况，对不规范穿戴行为进行实时预警。

## 应用场景

工作人员规范着装识别：检测餐厅（后厨、前场）、服务大厅、工地、工厂、油田等场景下的工作人员着装（衣服、帽子、口罩等）是否合规，对不规范行为进行实时预警，并提醒进行相应整改。

## 特色优势

- 识别准确：基于大量场景样本的基础上，对不同环境、光线、角度、目标状态等进行针对性的训练学习，算法模型识别精度高。
- 适应性强：算法模型场景适应性强，可直接用于不同空间布局的场景、各类餐厅后厨及用餐前场。
- 自研算法：通过多端多场景风险发现互通，形成检测、预警、整改、优化全链路闭环，结合数据分析、规则提取实现反哺生产，进而优化模型性能。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPG、JPEG、PNG、BMP。
- 图像大小：不超过4 MB，长宽比建议为16:9。
- 图像分辨率：大于1080×720像素。
- URL地址中不能包含中文字符。
- 推荐的图像和检测目标的分辨率如下表所示。

图像分辨率 (单位像素)	1280×720		1920×1080		2560×1440		3840×2160	
检测目标分辨率 (单位像素)	目标宽	目标高	目标宽	目标高	目标宽	目标高	目标宽	目标高
	13~256	12~360	19~384	18~540	26~512	24~720	38~768	36~1080

## 计费说明

关于着装检测的计费方式及报价，请参见[计费介绍](#)。

 说明 下方调试接口为付费接口。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectWorkwear	系统规定参数。取值： <b>DetectWorkwear</b> 。
ImageUrl	String	否	https://viapi-test.oss-cn-shanghai.aliyuncs.com/test-team/zhangchao run/tiyan/xxxx.jpg	图像URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

名称	类型	是否必选	示例值	描述
Labels.N	String	否	hat	检测的属性，取值如下： <ul style="list-style-type: none"> <li>mask: 口罩</li> <li>hat: 帽子</li> <li>clothes: 衣服</li> </ul>

## 返回数据

名称	类型	示例值	描述
RequestId	String	16CA8094-D7BC-51D4-8D55-6AC59AB20E0B	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of 识别内容		检测结果。
Property	Array of 识别属性		检测结果属性。仅当Type返回0或1时，该字段显示返回结果，否则返回为空。
Label	String	hat	检测的属性。
Probability	Object		检测概率数据。
No	double	0.00036084422	确定未戴口罩、未穿工装、未戴帽子的概率。
Threshold	Long	0	系统推荐的阈值。
Unknown	double	0.0006906331	未知概率。
Yes	double	0.9989485	确定戴口罩、穿工装、戴帽子的概率。
Rectangles	Object		检测结果位置信息。
Height	Long	96	高度，单位像素。
Left	Long	1067	左上角Y坐标，单位像素。

名称	类型	示例值	描述
Top	Long	426	左上角X坐标，单位像素。
Width	Long	88	宽度，单位像素。
Score	double	0.63913465	检测结果概率值。
Type	String	1	检测结果类型。取值如下： <ul style="list-style-type: none"><li>• 0: 人体</li><li>• 1: 人头</li><li>• 2: 人脸</li><li>• 3: 手</li></ul>

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DetectWorkwear
&Clothes={"MaxNum":1,"Threshold":0.4}
&ImageUrl=https://viapi-test.oss-cn-shanghai.aliyuncs.com/test-team/zhangchaorun/tiyan/xxxx
.jpg
&Labels=["hat"]
&公共请求参数
```

### 正常返回示例

XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectWorkwearResponse>
  <RequestId>16CA8094-D7BC-51D4-8D55-6AC59AB20E0B</RequestId>
  <Data>
    <Elements>
      <Property>
        <Label>hat</Label>
        <Probability>
          <No>0.00036084422</No>
          <Threshold>0</Threshold>
          <Unknown>0.0006906331</Unknown>
          <Yes>0.9989485</Yes>
        </Probability>
      </Property>
      <Rectangles>
        <Height>96</Height>
        <Left>1067</Left>
        <Top>426</Top>
        <Width>88</Width>
      </Rectangles>
      <Score>0.63913465</Score>
      <Type>1</Type>
    </Elements>
  </Data>
</DetectWorkwearResponse>
```

JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "16CA8094-D7BC-51D4-8D55-6AC59AB20E0B",
  "Data" : {
    "Elements" : {
      "Property" : {
        "Label" : "hat",
        "Probability" : {
          "No" : 3.6084422E-4,
          "Threshold" : 0,
          "Unknown" : 6.906331E-4,
          "Yes" : 0.9989485
        }
      },
      "Rectangles" : {
        "Height" : 96,
        "Left" : 1067,
        "Top" : 426,
        "Width" : 88
      },
      "Score" : 0.63913465,
      "Type" : 1
    }
  }
}
```

## 错误码

关于着装检测的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会系统自动清理删除。

# 4.8. 猫鼠识别

本文为您介绍猫鼠识别DetectKitchenAnimals的语法及示例。

## 功能描述

**猫鼠识别**能力基于视觉AI算法、物联网、大数据分析技术，猫鼠识别能力可以检测出场景中是否有猫、老鼠等动物，实现实时预警。

## 应用场景

餐饮后厨或仓库：针对后厨操作间、食材存储仓库等场地，实时检测是否有猫、老鼠等动物，并进行实时预警，提醒工作人员进行相应处理防范。

## 特色优势

- 识别准确：基于大量场景样本的基础上，对不同环境、光线、角度、目标状态等进行针对性的训练学习，算法模型识别精度高。
- 适应性强：算法模型场景适应性强，可直接用于不同空间布局的场景、各类餐厅后厨及用餐前场。
- 自研算法：通过多端多场景风险发现互通，形成检测、预警、整改、优化全链路闭环，结合数据分析、规则提取实现反哺生产，进而优化模型性能。

## 前提条件

请确保您已开通[目标检测服务](#)，若未开通服务请[立即开通](#)。

## 输入限制

- 图像格式：JPG、JPEG、PNG、BMP。
- 图像大小：不超过4 MB，长宽比建议为16:9。
- 图像分辨率：大于1080×720像素。
- 输入连续图像尺寸需要保持一致。
- 需要采集摄像头视角的图像。
- 图像URL地址中不能包含中文字符。
- 推荐的图像和检测目标的分辨率如下表所示。

图像分辨率 (单位像素)	1280×720		1920×1080		2560×1440		3840×2160	
检测目标分辨率 (单位像素)	目标宽	目标高	目标宽	目标高	目标宽	目标高	目标宽	目标高
		40~640	40~640	60~960	60~960	80~1280	80~1280	120~1920

## 计费说明

关于猫鼠识别的计费方式及报价，请参见[计费介绍](#)。

 说明 下方调试接口为付费接口，如需免费体验调试请前往[体验中心](#)。

## 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

## 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	DetectKitchenAnimals	系统规定参数。取值： <b>DetectKitchenAnimals</b> 。

名称	类型	是否必选	示例值	描述
ImageURLA	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectKitchenAnimals/DetectKitchenAnimals-left1.png	图像A的URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。
ImageURLB	String	是	http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/DetectKitchenAnimals/DetectKitchenAnimals-right1.png	图像B的URL地址。当前仅支持上海地域的OSS链接，如何生成URL请参见 <a href="#">生成URL</a> 。

## 返回数据

名称	类型	示例值	描述
RequestId	String	C306F16F-30E1-54F4-93DF-A52CCF6664D1	请求ID。
Data	Object		返回的结果数据内容。
Elements	Array of elements		检测结果。
Type	String	mouse	检测结果类型。取值如下： <ul style="list-style-type: none"> <li>mouse：老鼠</li> <li>cat：猫</li> </ul>
Score	Float	0.75105053	检测结果概率值。
Rectangles	Object		检测目标位置。
Top	Long	1048	左上角X坐标，单位像素。
Left	Long	292	左上角Y坐标，单位像素。

名称	类型	示例值	描述
Height	Long	64	高度，单位像素。
Width	Long	64	宽度，单位像素。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?Action=DetectKitchenAnimals
&ImageURLA=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/Detect
KitchenAnimals/DetectKitchenAnimals-left1.png
&ImageURLB=http://viapi-test.oss-cn-shanghai.aliyuncs.com/viapi-3.0domepic/objectdet/Detect
KitchenAnimals/DetectKitchenAnimals-right1.png
&公共请求参数
```

### 正常返回示例

#### XML 格式

```
HTTP/1.1 200 OK
Content-Type:application/xml
<DetectKitchenAnimalsResponse>
  <RequestId>C306F16F-30E1-54F4-93DF-A52CCF6664D1</RequestId>
  <Data>
    <Elements>
      <Type>mouse</Type>
      <Score>0.75105053</Score>
      <Rectangles>
        <Left>292</Left>
        <Top>1048</Top>
        <Height>64</Height>
        <Width>64</Width>
      </Rectangles>
    </Elements>
    <Elements>
      <Type>cat</Type>
      <Score>0.4602436</Score>
      <Rectangles>
        <Left>1344</Left>
        <Top>208</Top>
        <Height>112</Height>
        <Width>164</Width>
      </Rectangles>
    </Elements>
  </Data>
</DetectKitchenAnimalsResponse>
```

#### JSON 格式

```
HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "C306F16F-30E1-54F4-93DF-A52CCF6664D1",
  "Data" : {
    "Elements" : [ {
      "Type" : "mouse",
      "Score" : 0.75105053,
      "Rectangles" : {
        "Left" : 292,
        "Top" : 1048,
        "Height" : 64,
        "Width" : 64
      }
    }, {
      "Type" : "cat",
      "Score" : 0.4602436,
      "Rectangles" : {
        "Left" : 1344,
        "Top" : 208,
        "Height" : 112,
        "Width" : 164
      }
    }
  ]
}
```

## 错误码

关于猫鼠识别的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会被系统自动清理删除。

## 5. 查询异步任务结果

本文介绍查询异步任务结果GetAsyncJobResult的语法及示例。

### 功能描述

针对异步接口，调用API接口后，返回的并不是真正的请求结果，您需要保存返回结果中的RequestId，然后调用GetAsyncJobResult来获取真正的请求结果。

 **说明** 异步任务的文件有过期时间，有效期是30分钟。如需长期使用，需及时将文件下载到本地服务器或存储在OSS中。有关OSS更多操作信息，请参见[上传文件](#)。

当前目标检测类目中IPC视频目标检测（DetectVideoIPCObject）为异步接口，需要调用GetAsyncJobResult来获取真实的请求结果。

### 调试

您可以在OpenAPI Explorer中直接运行该接口，免去您计算签名的困扰。运行成功后，OpenAPI Explorer可以自动生成SDK代码示例。

### 请求参数

名称	类型	是否必选	示例值	描述
Action	String	是	GetAsyncJobResult	系统规定参数。取值：GetAsyncJobResult。
JobId	String	是	35B11E1B-800C-4598-B5AA-577E3BDBD917	异步接口返回的RequestId，输入后可以查询异步接口的真实请求结果。

### 返回数据

名称	类型	示例值	描述
RequestId	String	87FC80D2-2571-4BBD-BD61-AFF7912C556D	请求ID。
Data	Object		返回的结果数据内容。
Status	String	PROCESS_SUCCESS	异步任务状态。包括： <ul style="list-style-type: none"><li>• QUEUING：任务排队中</li><li>• PROCESSING：异步处理中</li><li>• PROCESS_SUCCESS：处理成功</li><li>• PROCESS_FAILED：处理失败</li><li>• TIMEOUT_FAILED：任务超时未处理完成</li><li>• LIMIT_RETRY_FAILED：超过最大重试次数</li></ul>

名称	类型	示例值	描述
ErrorMessage	String	paramsillegal	异步任务错误码信息。
Result	String	{\inputFile\": \"oss://public-vigen-video/guotian.xgt/test_images/test_video\", \"width\": 1280, \"height\": 720, \"frames\": [{\time\": 6124533574, \"elements\": [{\type\": \"PERSON\", \"score\": 0.7812, \"x\": 289, \"y\": 271, \"width\": 100, \"height\": 156}, {\type\": \"PERSON\", \"score\": 0.4377, \"x\": 917, \"y\": 267, \"width\": 34, \"height\": 51}]]}}	异步任务返回的真实请求结果。
ErrorCode	String	InvalidParameter	异步任务错误码。
JobId	String	35B11E1B-800C-4598-B5AA-577E3BDBD917	异步任务ID。

## 示例

### 请求示例

```
http(s)://[Endpoint]/?JobId=35B11E1B-800C-4598-B5AA-577E3BDBD917
&公共请求参数
```

### 正常返回示例

XML 格式

```

HTTP/1.1 200 OK
Content-Type:application/xml
<RequestId>87FC80D2-2571-4BBD-BD61-AFF7912C556D</RequestId>
<Data>
  <Status>PROCESS_SUCCESS</Status>
  <ErrorMessage>paramsIllegal</ErrorMessage>
  <Result>{\ "inputFile\":"oss://public-vigen-video/guotian.xgt/test_images/test_video\",
  \ "width\":1280,\ "height\":720,\ "frames\":[{\ "time\":6124533574,\ "elements\":[{\ "type\":"PERSON\",
  \ "score\":0.7812,\ "x\":289,\ "y\":271,\ "width\":100,\ "height\":156},{\ "type\":"PERSON\",
  \ "score\":0.4377,\ "x\":917,\ "y\":267,\ "width\":34,\ "height\":51}]}]}</Result>
  <ErrorCode>InvalidParameter</ErrorCode>
  <JobId>35B11E1B-800C-4598-B5AA-577E3BDBD917</JobId>
</Data>

```

### JSON 格式

```

HTTP/1.1 200 OK
Content-Type:application/json
{
  "RequestId" : "87FC80D2-2571-4BBD-BD61-AFF7912C556D",
  "Data" : {
    "Status" : "PROCESS_SUCCESS",
    "ErrorMessage" : "paramsIllegal",
    "Result" : "{\\\\"inputFile\\\\":\\\\"oss://public-vigen-video/guotian.xgt/test_images/tes
t_video\\\\" ,\\\\"width\\\\":1280,\\\\"height\\\\":720,\\\\"frames\\\\":[{\\\\"time\\\\":6124533574,
\\\\"elements\\\\":[{\\\\"type\\\\":\\\\"PERSON\\\\" ,\\\\"score\\\\":0.7812,\\\\"x\\\\":289,\\\\"y\\\\"
:271,\\\\"width\\\\":100,\\\\"height\\\\":156},{\\\\"type\\\\":\\\\"PERSON\\\\" ,\\\\"score\\\\":0.437
7,\\\\"x\\\\":917,\\\\"y\\\\":267,\\\\"width\\\\":34,\\\\"height\\\\":51}]}]}\\\\"",
    "ErrorCode" : "InvalidParameter",
    "JobId" : "35B11E1B-800C-4598-B5AA-577E3BDBD917"
  }
}

```

## 错误码

关于查询异步任务结果的错误码，详情请参见[常见错误码](#)。

## SDK参考

阿里云视觉AI能力支持使用SDK调用，具体可参见[SDK总览](#)下载安装。

## 安全声明

- 请确保上传的图片或文件来源符合相应的法律法规。
- 通过体验调试上传的临时文件有效期为1小时，在24小时后会系统自动清理删除。

## 6. 公共信息

### 6.1. 请求结构

目标检测服务支持基于URL发送HTTP/HTTPS请求。请求参数需要包含在URL中，请求及返回结果都使用UTF-8 字符集编码。

以下为一条未编码的URL请求示例：

```
https://objectdet.cn-shanghai.aliyuncs.com/?Action=DetectMainBody&<公共请求参数>
```

- `https`：指定了请求通信协议。
- `objectdet.cn-shanghai.aliyuncs.com`：指定了目标检测服务的接入地址（Endpoint）。
- `cn-shanghai`：区域（RegionId），请参见[访问域名查看支持的域名](#)。
- `Action=DetectMainBody`：指定了要调用的API。
- `<公共请求参数>`：API接口中使用了公共请求头（Common Request Headers），该内容可以被所有的视觉智能服务请求使用。详细说明请参见[公共请求参数](#)。

### 6.2. 公共请求参数

阿里云视觉智能开放平台是RPC API类型接口，每个API都必须提供这些公共请求参数。

#### 请求示例

```
http://objectdet.cn-shanghai.aliyuncs.com/?Signature=NAxw11W9ROkidJfGeZrsKUXw%2BQ****
&AccessKeyId=testId
&Action=DetectMainBody
&Format=JSON
&RegionId=cn-shanghai
&SignatureMethod=HMAC-SHA1
&SignatureNonce=313ef0fb-0393-464f-a6e2-59d9ca2585b1
&SignatureVersion=1.0
&Timestamp=2019-10-12T08%3A18%3A18Z
&Version=2019-12-30
```

#### 请求参数

名称	类型	是否必填	说明
Signature	String	是	请求签名，即最终生成的签名结果值。如何生成请求签名，请参见 <a href="#">请求签名</a> 。
AccessKeyId	String	是	访问服务使用的密钥ID。具体请参见 <a href="#">创建AccessKey</a>
Action	String	是	API接口名称。
Format	String	否	返回消息的格式。取值：JSON（默认值）或XML。

名称	类型	是否必填	说明
RegionId	String	否	API支持的RegionID。例如：cn-shanghai。
SignatureMethod	String	是	签名方式。取值：HMAC-SHA1。
SignatureNonce	String	是	签名唯一随机数。用于防止网络重放攻击，建议您在不同请求使用不同的随机数。JAVA语言建议用 <code>java.util.UUID.randomUUID()</code> 生成。
SignatureVersion	String	是	签名算法版本。取值：1.0。
Timestamp	String	是	请求时间戳，为日期格式。使用UTC时间按照ISO8601标准，格式为YYYY-MM-DDThh:mm:ssZ。示例：2019-10-13T10:00:00Z 表示北京时间 2019 年 10 月 13 日 18 点 00 分 00 秒。
Version	String	是	API 的版本号，格式为 YYYY-MM-DD。取值：2019-12-30。

## 6.3. 公共响应参数

阿里云视觉智能开放平台是RPC API类型接口，所有的API响应均提供一组公共响应头。

您每次发送的接口调用请求，无论成功与否，系统都会返回一个唯一识别码RequestId。

### 请求示例

```
https://objectdet.cn-shanghai.aliyuncs.com/
?Format=xml
&Version=2019-12-30
&Signature=Pc5WB8gokVn0xfeu%2FZV%2BiNM1dg****
&SignatureMethod=HMAC-SHA1
&SignatureNonce=15215528852396
&SignatureVersion=1.0
&AccessKeyId=key-test
&Timestamp=2014-10-10T12:00:00Z
```

### 成功结果示例

调用API服务后，系统采用统一格式返回数据。返回2xx的HTTP状态码表示调用成功，调用成功返回的数据格式主要有XML和JSON两种。外部系统可以在请求时传入参数来制定返回的数据格式，默认为XML格式。为了便于查看，本文档中的返回示例做了格式化处理，实际返回结果没有进行换行、缩进等处理。

#### ● JSON格式

```
{
  "RequestId": "4C467B38-3910-447D-87BC-AC049166F216",
  /* 返回结果数据 */
}
```

- XML格式

```
<?xml version="1.0" encoding="utf-8"?>
<!--结果的根结点-->
<接口名称+Response>
  <!--返回请求标签-->
  <RequestId>4C467B38-3910-447D-87BC-AC049166F216</RequestId>
  <!--返回结果数据-->
</接口名称+Response>
```

## 错误结果示例

当调用出错时，HTTP请求会返回一个4xx或5xx的HTTP状态码。返回的消息体中包含具体的错误码及错误信息，还包含一个全局唯一的请求ID（RequestId）和一个该次请求访问的站点ID（HostId）。您可以根据返回的错误码来定位错误原因。若无法在调用方定位错误原因，请联系阿里云客服人员，并提供该HostId和RequestId，以便能尽快帮您解决问题。

- JSON格式

```
{
  "RequestId": "7463B73D-35CC-4D19-A010-6B8D65D242EF",
  "HostId": "objectdet.cn-shanghai.aliyuncs.com",
  "Code": "UnsupportedOperation",
  "Message": "The specified action is not supported."
}
```

- XML格式

```
<?xml version="1.0" encoding="UTF-8"?>
<Error>
  <RequestId>8906582E-6722-409A-A6C4-0E7863B733A5</RequestId>
  <HostId>objectdet.cn-shanghai.aliyuncs.com</HostId>
  <Code>UnsupportedOperation</Code>
  <Message>The specified action is not supported.</Message>
</Error>
```

## 7. 常见错误码

当API请求发生错误时，服务端会返回错误信息。本文档为您介绍API返回的错误码信息。

### 问题诊断中心

使用[问题诊断中心](#)可帮助开发者更高效地诊断API使用过程中的问题。

### 机动车检测错误码

状态码	错误码	描述
400	InvalidImage.Download	图片下载失败。
400	InvalidImage.Timeout	图片下载超时。
400	InvalidImage.URL	参数错误：无效的图像URL。
500	InternalServerError.Decode	图像解码失败。
500	InternalServerError.Process	推理过程异常。

### 主体检测错误码

状态码	错误码	描述
400	MissingParameter	入参缺失。
400	InvalidParameter	入参错误。
400	InvalidParameter.Unsuitable	数据为空或不适合被处理。
500	InternalServerError	调用算法抛错。
500	InternalServerError.Env	初始化环境失败。
500	InternalServerError.Model	模型加载错误。
500	InternalServerError.Convert	输入参数转换失败。
500	InternalServerError.Algo	算法服务出错。
500	InternalServerError.Transfer	下载或上传数据失败。

### 公共错误码

API的公共错误码请参见[API错误中心](#)。

### 处理建议

请求服务出现错误码提示，请参见错误码描述查看具体错误原因，并进行错误处理。针对一些常见高频的错误类型，此处提供一些错误处理建议供您参考。

- URL类错误码

此类错误码一般都是文件URL不符合要求，例如URL不安全、无效。建议确认当前URL是否正确，或者您可以重新生成URL，具体请参见[生成URL](#)。

- 图像加载错误码

此类错误也可以看做是图像URL错误，例如图像下载错误、地域错误。当前系统仅支持 华东2（上海） 地域，如果您通过OSS生成URL，那么您在OSS上创建的Bucket所属地域应该为 华东2（上海） 。

- 参数类错误码

此类错误码一般是输入参数不符合要求，例如参数缺失、参数类型错误等。建议您检查输入参数是否符合要求，您可以参考对应API文档中的请求参数表格进行检查确认。

- 系统服务类错误码

如果出现此类错误码，建议您重新发起服务请求，如果问题依然存在，请在钉钉群组（23109592）中反馈问题。