
IMAGERECOGNITION 文档

2018-03-22



百度云
cloud.baidu.com

目录

1	API参考	1
1.1	简介	1
1.1.1	接口能力	1
1.1.2	请求格式	2
1.1.3	返回格式	2
1.1.4	请求限制	2
1.2	调用方式	3
1.2.1	调用方式一	3
1.2.2	调用方式二	3
1.3	细粒度图像识别——菜品识别	4
1.3.1	接口描述	4
1.3.2	请求说明	4
1.3.3	返回说明	5
1.4	细粒度图像识别——车型识别	7
1.4.1	接口描述	7
1.4.2	请求说明	7
1.4.3	返回说明	8
1.5	细粒度图像识别——logo商标识别	9
1.5.1	接口描述	9
1.5.2	logo商标识别	9
	请求说明	9

返回说明	10
1.5.3 logo商标识别—添加	12
返回说明	13
1.5.4 logo商标识别—删除	14
返回说明	15
1.6 细粒度图像识别—动物识别	16
1.6.1 接口描述	16
1.6.2 请求说明	16
1.6.3 返回说明	17
1.7 细粒度图像识别—植物识别	18
1.7.1 接口描述	18
1.7.2 请求说明	18
1.7.3 返回说明	19
1.8 通用图像分析——图像主体检测	20
1.8.1 接口描述	20
1.8.2 请求说明	20
1.8.3 返回说明	21
1.9 定制化图像识别——定制图像分类	22
1.9.1 接口描述	22
1.9.2 请求说明	22
1.9.3 返回说明	24
1.10 错误码	24
2 产品定价	28
2.1 计费简介	28
2.2 免费额度	28
2.3 免费/付费对比	28
2.4 价目表	29

2.4.1	图像主体检测	29
2.4.2	植物识别	29
2.4.3	动物识别	29
2.4.4	菜品识别	30
2.4.5	车型识别	30
2.5	费用举例	30
2.6	余额不足提醒与欠费处理	31
2.6.1	余额不足提醒	31
2.6.2	欠费处理	31
3	C# SDK文档	32
3.1	简介	32
3.1.1	接口能力	32
3.1.2	版本更新记录	32
3.2	快速入门	33
3.2.1	安装图像识别 C# SDK	33
	方法一：使用Nuget管理	33
	方法二：下载安装	33
3.2.2	新建交互类	33
3.3	接口说明	34
3.3.1	菜品识别	34
3.3.2	车辆识别	36
3.3.3	logo商标识别	37
3.3.4	logo商标识别—添加	39
3.3.5	logo商标识别—删除	40
3.3.6	动物识别	41
3.3.7	植物识别	43
3.3.8	图像主体检测	44

3.3.9	FAQ	46
	1. throw exception “fail to fetch token: 基础连接已关闭”	46
	2. SSL报错 “The authentication or decryption has failed”	46
3.4	错误信息	46
3.4.1	错误返回格式	46
3.4.2	错误码	46
4	Java SDK文档	49
4.1	简介	49
4.1.1	接口能力	49
4.1.2	版本更新记录	49
4.2	快速入门	50
4.2.1	安装ImageClassify Java SDK	50
4.2.2	新建AipImageClassifyClient	51
4.2.3	配置AipImageClassify	53
4.3	接口说明	53
4.3.1	菜品识别	53
4.3.2	车辆识别	55
4.3.3	logo商标识别	57
4.3.4	logo商标识别—添加	59
4.3.5	logo商标识别—删除	60
4.3.6	动物识别	61
4.3.7	植物识别	63
4.3.8	图像主体检测	64
4.4	错误信息	66
4.4.1	错误返回格式	66
4.4.2	错误码	67
5	Nodejs SDK文档	70

5.1	简介	70
5.1.1	接口能力	70
5.1.2	版本更新记录	70
5.2	快速入门	71
5.2.1	安装图像识别 Node SDK	71
5.2.2	新建AipImageClassifyClient	72
5.3	接口说明	73
5.3.1	菜品识别	73
5.3.2	车辆识别	75
5.3.3	logo商标识别	77
5.3.4	logo商标识别—添加	79
5.3.5	logo商标识别—删除	81
5.3.6	动物识别	82
5.3.7	植物识别	84
5.3.8	图像主体检测	85
5.4	错误信息	88
5.4.1	错误返回格式	88
5.4.2	错误码	88
6	PHP SDK文档	91
6.1	简介	91
6.1.1	接口能力	91
6.1.2	版本更新记录	91
6.2	快速入门	92
6.2.1	安装图像识别 PHP SDK	92
6.2.2	新建AipImageClassify	92
6.2.3	配置AipFace	93
6.3	接口说明	93

6.3.1	菜品识别	93
6.3.2	车辆识别	95
6.3.3	logo商标识别	97
6.3.4	logo商标识别—添加	99
6.3.5	logo商标识别—删除	100
6.3.6	动物识别	101
6.3.7	植物识别	103
6.3.8	图像主体检测	104
6.4	错误信息	106
6.4.1	错误返回格式	106
6.4.2	错误码	106
7	Python SDK文档	109
7.1	简介	109
7.1.1	接口能力	109
7.1.2	版本更新记录	109
7.2	快速入门	110
7.2.1	安装图像识别 Python SDK	110
7.2.2	新建AipImageClassify	110
7.2.3	配置AipFace	111
7.3	接口说明	111
7.3.1	菜品识别	111
7.3.2	车辆识别	113
7.3.3	logo商标识别	115
7.3.4	logo商标识别—添加	117
7.3.5	logo商标识别—删除	119
7.3.6	动物识别	120
7.3.7	植物识别	122

7.3.8	图像主体检测	123
7.4	错误信息	125
7.4.1	错误返回格式	125
7.4.2	错误码	125
8	C++ SDK文档	128
8.1	简介	128
8.1.1	接口能力	128
8.1.2	版本更新记录	128
8.2	快速入门	129
8.2.1	安装图像识别 C++ SDK	129
8.2.2	新建client	129
8.3	接口说明	130
8.3.1	菜品识别	130
8.3.2	车辆识别	132
8.3.3	logo商标识别	134
8.3.4	logo商标识别—添加	136
8.3.5	logo商标识别—删除	137
8.3.6	动物识别	138
8.3.7	植物识别	140
8.3.8	图像主体检测	141
8.4	错误信息	143
8.4.1	错误返回格式	143
8.4.2	错误码	143

第1章 API参考

1.1 简介

Hi, 您好, 欢迎使用百度图像识别API服务。

本文档主要针对API开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171) :<http://ai.baidu.com/forum/topic/list/171>
- 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);

1.1.1 接口能力

产品功能	接口名称	接口能力简要描述
通用图像分析	图像主体检测	识别图像中的主体具体坐标位置。
细粒度图像识别	菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
	车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
	logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
	动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
	植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。

产品功能	接口名称	接口能力简要描述
定制化图像识别	定制图像分类	基于自定义训练出的图像分类模型，实现个性化图像识别。

1.1.2 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。

1.1.3 返回格式

JSON格式

1.1.4 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、JPEG、BMP、GIF**

接口名称	图片编码后大小限额
菜品识别	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px
车型识别	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px
logo商标识别	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px
图像主体检测	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px
动物识别	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px
植物识别	base64编码后大小不超过4M，最短边至少15px，最长边最大4096px

1.2 调用方式

调用AI服务相关的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。

1.2.1 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，需要每30天进行定期更换；

POST中参数按照API接口说明调用即可。

例如菜品识别API，使用HTTPS POST发送：

```
https://aip.baidubce.com/rest/2.0/image-classify/v2/dish?access_token=24.f9ba9c5241b67688bb4adbed82592000.1485570332.282335-8574074
```

获取access_token示例代码

```
{% AccessToken %}
```

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

1.2.2 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。

1.3 细粒度图像识别——菜品识别

1.3.1 接口描述

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

1.3.2 请求说明

请求示例

HTTP 方法：[POST](#)

请求URL：<https://aip.baidubce.com/rest/2.0/image-classify/v2/dish>

URL参数：

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下：

参数	值
Content-Type	application/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	-	Base64 编码字符串，以图片文件形式请求时必须填。(支持图片格式：jpg， bmp， png， jpeg)，图片大小不超过 4M。最短边至少 15px，最长边最大 4096px。
top_num	否	unit32	-	返回结果top n, 默认5.
filter_threshold	是	float	-	默认 0.95，可以通过该参数调节识别效果，降低非菜识别率.

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。

```
{% IMAGECLASSIFY-API-Dish %}
```

1.3.3 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
result_num	否	unit32	返回结果数目，及result数组中的元素个数
result	否	array()	菜品识别结果数组

字段	是否必选	类型	说明
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	float	卡路里，每100g的卡路里含量
+probability	否	float	识别结果中每一行的置信度值，0-1

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 2471272194,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤（串类）",
      "calorie": 333.33,
      "probability": 0.35874313116074
    },
    {
      "name": "鱿鱼",
      "calorie": 333.33,
      "probability": 0.20610593259335
    },
    {
      "name": "板筋",
      "calorie": 333.33,
      "probability": 0.15860831737518
    },
    {
      "name": "鸡脆骨",
      "calorie": 333.33,
      "probability": 0.077698558568954
    },
    {
      "name": "麻辣烫",
```

```
        "calorie": 333.33,  
        "probability": 0.041968926787376  
    }  
]  
  
}
```

1.4 细粒度图像识别—车型识别

1.4.1 接口描述

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

1.4.2 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/car>

URL参数：

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下：

参数	值
Content-Type	application/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	true	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
top_num	false	uint32	-	返 回 结 果 top n，默认5。

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。

```
{% IMAGECLASSIFY-API-Car %}
```

1.4.3 返回说明

返回参数

字段	是否必选	类型	说明
log_id	否	uint64	唯一的log id，用于问题定位
result	否	car-result()	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	double	置信度，示例：0.5321

返回示例

HTTP/1.1 200 OK

x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e

Cache-Control: no-cache

Server: BWS


```
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "\\u5b9d\\u9a6cX6",
      "score": 0.40009739995003
    },
    {
      "name": "\\u5b9d\\u9a6c|X5",
      "score": 0.3136293888092
    },
    {
      "name": "\\u5b9d\\u9a6c|X4",
      "score": 0.13881112635136
    }
  ]
}
```

1.5 细粒度图像识别—logo商标识别

1.5.1 接口描述

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（在[控制台](#)创建应用并申请建库）并通过调用logo入口接口完成自定义logo入库，提高识别效果。

1.5.2 logo商标识别

[请求说明](#) [请求示例](#)

HTTP 方法：[POST](#)

请求URL：<https://aip.baidubce.com/rest/2.0/image-classify/v2/logo>

URL参数：

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下：

参数	值
Content-Type	application/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，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.
custom_lib	否	boolean	-	是否只检索用户子库，true 则只检索用户子库，false(默认)为检索底库+用户子库

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。

{% IMAGECLASSIFY-API-Logo %}

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
result_num	是	uint32	识别结果数，标识返回结果数目

字段	是否必选	类型	说明
result	否	array()	返回结果数组，每一项为一个识别出的logo
+location	是	object	位置信息（左起像素位置、上起像素位置、像素宽、像素高）
++left	是	number	左起像素位置
++top	是	number	上起像素位置
++width	是	number	像素宽
++height	是	number	像素高
+name	是	string	识别的品牌名称
+probability	是	number	分类结果置信度（0-1.0）
+type	是	number	type=0 为 1 千种高优商标识别结果;type=1 为 2 万类logo库的结果；其它 type 为自定义logo库结果

返回示例

```

HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
      "type": 0,
      "name": "科颜氏",
      "probability": 0.99998807907104,
      "location": {
        "width": 296,

```

```
        "top": 20,  
        "height": 128,  
        "left": 23  
    }  
}  
]  
}
```

1.5.3 logo商标识别—添加

使用入库接口请先在[控制台](#)创建应用并申请建库，建库成功后方可正常使用。

请求示例

HTTP 方法: **POST**

请求URL: https://aip.baidubce.com/rest/2.0/realtime_search/v1/logo/add

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下:

参数	值
Content-Type	application/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，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.
brief	是	string	-	检索时带回。此处要传对应的name与code 字段，name长度小于100B，code 长 度 小 于 150B。示 例：“{” name \ “: \” abc\ “, \” code \ “: \” ddd\ “}”

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。

```
{% IMAGECLASSIFY-API-LogoAdd %}
```

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
cont_sign	是	string	输入图片签名，可用于删除

返回示例

HTTP/1.1 200 OK

```
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

1.5.4 logo商标识别—删除

使用删除接口请先在[控制台](#)创建应用并申请建库，建库成功后先调用入库接口完成logo图片入库，删除接口用户在已入库的logo图片中删除图片。

请求示例

HTTP 方法: [POST](#)

请求URL: https://aip.baidubce.com/rest/2.0/realtime_search/v1/logo/delete

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下:

参数	值
Content-Type	application/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	-	base64 编 码后的图片数据（和 cont_sign 二选一）。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。
cont_sign	是	string	-	图片签名（和 image 二选一，image 优先级更高）

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。

```
{% IMAGECLASSIFY-API-LogoDelete %}
```

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

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 50488417
}
```

1.6 细粒度图像识别—动物识别

1.6.1 接口描述

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出动物识别结果。

1.6.2 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/animal>

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下:

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下:

请求参数

参数名称	是否必选	类型	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
top_num	否	integer	6	返回预测得分 top结果数，默 认为6

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。{% IMAGECLASSIFY-API-Animal %}

1.6.3 返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
result	array(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	uint32	是	置 信 度，示 例：0.5321

返回示例

```
{
  "log_id": 304829369,
  "result": [
```

```
{
  "name": "蒙古马",
  "score": 0.41471540927887
},
{
  "name": "伊犁马",
  "score": 0.39988899230957
},
{
  "name": "哈萨克马",
  "score": 0.39988899230957
}
...
]
```

1.7 细粒度图像识别—植物识别

1.7.1 接口描述

该请求用于识别一张图片，即对于输入的一张图片（可正常解码，且长宽比较合适），输出植物识别结果。

1.7.2 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/plant>

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下:

参数	值
Content-Type	application/x-www-form-urlencoded

Body中放置请求参数，参数详情如下：

请求参数

| 参数名称 | 是否必选 | 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 | 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。 `{% IMAGECLASSIFY-API-Plant %}`

1.7.3 返回说明

返回参数

参数	类型	是否必须	说明
log_id	uint64	是	唯一的log id，用于问题定位
result	arry(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃
+score	uint32	是	置信度，示例：0.5321

返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
  ]
}
```

```
...  
]  
}
```

1.8 通用图像分析——图像主体检测

1.8.1 接口描述

用户向服务请求检测图像中的主体位置。

1.8.2 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/object_detect

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考“ Access Token获取 ”

Header如下:

参数	值
Content-Type	application/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	true	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
with_face	false	number	-	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。

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。

```
{% IMAGECLASSIFY-API-DETECT %}
```

1.8.3 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
result	否	watermark-location	裁剪结果

字段	是否必选	类型	说明
+left	否	uint32	表示定位位置的长方形左上顶点的水平坐标
+top	否	uint32	表示定位位置的长方形左上顶点的垂直坐标
+width	否	uint32	表示定位位置的长方形的宽度
+height	否	uint32	表示定位位置的长方形的高度

返回示例

```
HTTP/1.1 200 OK
x-bce-request-id: 73c4e74c-3101-4a00-bf44-fe246959c05e
Cache-Control: no-cache
Server: BWS
Date: Tue, 18 Oct 2016 02:21:01 GMT
Content-Type: application/json;charset=UTF-8
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

1.9 定制化图像识别——定制图像分类

1.9.1 接口描述

基于自定义训练出的图像分类模型，实现个性化图像识别。详情访问：[定制化图像训练平台](#)进行训练。更多训练模型过程中的常见问题查看 <http://ai.baidu.com/docs#/FAQ/top> 如有其它问题，欢迎加入QQ群185289236进行咨询。

1.9.2 请求说明

请求示例

HTTP 方法: [POST](#)

请求URL: 请首先在[定制化图像训练平台](#)进行自定义模型训练, 完成训练后可在服务列表中查看并获取url。

URL参数:

参数	值
access_token	通过 API Key 和 Secret Key 获取的 access_token,参考 “Access Token获取”

Header如下:

参数	值
Content-Type	application/json

Body请求示例:

```
{
  "image": "<base64数据>",
  "top_num": 5
}
```

Body中放置请求参数, 参数详情如下:

请求参数

参数	是否必选	类型	可选值范围	说明
image	true	string	-	图像数据, base64 编码, 要求 base64 编码后大小不超过 4M, 最短边至少 15px, 最长边最大 4096px, 支持 jpg/ png/ bmp 格式 **注意请去掉头部**
top_num	false	number	-	返回分类数量, 默认为6个

1.9.3 返回说明

返回参数

字段	是否必选	类型	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	分类结果数组
+name	否	string	分类名称
+score	否	number	置信度

1.10 错误码

若请求错误，服务器将返回的JSON文本包含以下参数：

- error_code：错误码。
- error_msg：错误描述信息，帮助理解 and 解决发生的错误。

例如Access Token失效返回：

```
{
  "error_code": 110,
  "error_msg": "Access token invalid or no longer valid"
}
```

需要重新获取新的Access Token再次请求即可。

错误码	错误信息	描述
1	Unknown error	服务器内部错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
2	Service temporarily unavailable	服务暂不可用，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。

错误码	错误信息	描述
3	Unsupported openapi method	调用的API不存在，请检查后重新尝试
4	Open api request limit reached	集群超额
6	No permission to access data	无权限访问该用户数据
17	Open api daily request limit reached	每天请求量超额
18	Open api qps request limit reached	QPS超额
19	Open api total request limit reached	请求总量超额
100	Invalid parameter	无效的access_token参数，请检查后重新尝试
110	Access token invalid or no longer valid	access_token无效
111	Access token expired	access token过期
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（649285136）或工单联系技术支持团队。
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

错误码	错误信息	描述
216201	image format error	上传的图片格式错误，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
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（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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

错误码	错误信息	描述
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
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
283300	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（185289236）或工单联系技术支持团队
336001	Invalid Argument	入参格式有误，比如缺少必要参数、图片base64编码错误等等，可检查下图片编码、代码格式是否有误。有疑问请通过QQ群（185289236）或工单联系技术支持团队

第2章 产品定价

2.1 计费简介

图像识别下共有5项能力可在平台自助付费使用：图像主体检测、植物识别、动物识别、菜品识别、车型识别。

图像识别的计费方式为：按调用量计费，根据实际调用的次数，系统每小时会对您的百度云账户进行扣费。

2.2 免费额度

图像识别下各个能力都具有免费调用额度，开通付费后，每日免费调用额度仍保留，免费额度如下：

产品服务	免费调用额度
图像主体检测	500次/日
植物识别	500次/日
动物识别	500次/日
菜品识别	500次/日
车型识别	500次/日

注：成功调用与失败调用均消耗免费额度，未用完的免费额度不流转到下一日。

2.3 免费/付费对比

对于图像识别下的各个能力，免费使用和开通付费后使用的配置有较大差异，具体对比如下：

状态	免费调用额度	超过调用额度	QPS限制
免费状态	拥有	不响应请求	不保证并发
付费状态	拥有	可继续请求	保证10次并发

2.4 价目表

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

2.4.1 图像主体检测

每日500次免费调用额度，免费额度用尽后开始计费，价格如下：

月调用量（万次）	图像主体检测（元/千次）
0<月调用量≤5	0.70
5<月调用量≤10	0.60
10<月调用量≤20	0.50
20<月调用量≤50	0.40
50<月调用量≤100	0.35
100<月调用量	0.30

说明：调用失败不计费

2.4.2 植物识别

每日500次免费调用额度，免费额度用尽后开始计费，价格如下：

月调用量（万次）	植物识别（元/千次）
0<月调用量≤5	3.0
5<月调用量≤10	2.8
10<月调用量≤20	2.5
20<月调用量≤50	2.0
50<月调用量≤100	1.8
100<月调用量	1.5

说明：调用失败不计费

2.4.3 动物识别

每日500次免费调用额度，免费额度用尽后开始计费，价格如下：

月调用量（万次）	动物识别（元/千次）
0<月调用量≤5	1.0

5<月调用量≤10 | 0.9 |
10<月调用量≤20 | 0.8 |
20<月调用量≤50 | 0.7 |
50<月调用量≤100 | 0.6 |
100<月调用量 | 0.5 |

说明：调用失败不计费

2.4.4 菜品识别

每日500次免费调用额度，免费额度用尽后开始计费，价格如下：

月调用量（万次） | 菜品识别（元/千次） |
——— | —— |
0<月调用量≤5 | 0.70 |
5<月调用量≤10 | 0.60 |
10<月调用量≤20 | 0.50 |
20<月调用量≤50 | 0.40 |
50<月调用量≤100 | 0.35 |
100<月调用量 | 0.30 |

说明：调用失败不计费

2.4.5 车型识别

每日500次免费调用额度，免费额度用尽后开始计费，价格如下：

月调用量（万次） | 车型识别（元/千次） |
——— | —— |
0<月调用量≤5 | 3.0 |
5<月调用量≤10 | 2.8 |
10<月调用量≤20 | 2.5 |
20<月调用量≤50 | 2.0 |
50<月调用量≤100 | 1.8 |
100<月调用量 | 1.5 |

说明：调用失败不计费

2.5 费用举例

从2017-2-1至2017-2-28，本月图像主体检测的月调用量为15万次（已除去免费额度），费用如下：

前5万次落入0~5w阶梯，单价0.7元/千次，费用为35元
中间5万次落入5~10w阶梯，单价0.6元/千次，费用为30元
后5万次落入10~20w阶梯，单价0.5元/千次，费用为25元
本月费用共计：90元

2.6 余额不足提醒与欠费处理

2.6.1 余额不足提醒

- 根据您最近3天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来3天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近2天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来2天的费用，若不足以支付，系统发送续费提醒。
- 根据您最近1天的账单金额来判断您的账户余额（含可用代金券）是否足够支付未来1天的费用，若不足以支付，系统发送续费提醒。

2.6.2 欠费处理

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

第3章 C# SDK文档

3.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对C#开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

3.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

3.1.2 版本更新记录

上线日期	版本号	更新内容
2017.12.21	3.3.0	接口升级
2017.9.27	3.1.0	新增动物识别、植物识别

上线日期	版本号	更新内容
2017.9.7	3.0.0	新增图像识别服务

3.2 快速入门

3.2.1 安装图像识别 C# SDK

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

方法一：使用Nuget管理 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 [图像识别 C# SDK目录结构](#)

```
Baidu.Aip
├─ AipSdk.dll           // 百度AI服务 windows 动态库
├─ AipSdk.XML           // DLL注释
└─ thirdparty           // 第三方依赖
```

支持平台：[.Net Framework 3.5 及以上版本](#)

如果需要在 Unity / .net core 等平台使用，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 [thirdparty](#) 中的dll文件添加为引用。

3.2.2 新建交互类

Baidu.Aip.ImageClassify.ImageClassify是图像识别的交互类，为使用图像识别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.ImageClassify.ImageClassify(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”。

3.3 接口说明

3.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

```
public void DishDetectDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    // 调用菜品识别，可能会抛出网络等异常，请使用try/catch捕获
    var result = client.DishDetect(image);
    Console.WriteLine(result);
    // 如果有可选参数
    var options = new Dictionary<string, object>{
        {"top_num", 3}
    };
    // 带参数调用菜品识别
    result = client.DishDetect(image, options);
    Console.WriteLine(result);
}
```

菜品识别 [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
top_num	否	string	返回预测得分top结果数，默认为5

菜品识别 [返回数据参数详情](#)

参数	类型	是否必须	说明
log_id	是	number	唯一的log id, 用于问题定位
result_num	否	number	返回结果数目, 及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名, 示例: 鱼香肉丝
+calorie	否	number	卡路里, 每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值, 0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤(串类)",
      "calorie": 83.12,
      "probability": 0.35874313116074
    },
    {
      "name": "鱿鱼",
      "calorie": 402.12,
      "probability": 0.20610593259335
    },
    {
      "name": "板筋",
      "calorie": 333.55,
      "probability": 0.15860831737518
    },
    {
      "name": "鸡脆骨",
      "calorie": 661.33,
      "probability": 0.077698558568954
    },
    {
```

```
        "name": "麻辣烫",
        "calorie": 273.33,
        "probability": 0.041968926787376
    }
]
}
```

3.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

```
public void CarDetectDemo() {
    var image = File.ReadAllBytes("图片文件路径");
    // 调用车辆识别，可能会抛出网络等异常，请使用try/catch捕获
    var result = client.CarDetect(image);
    Console.WriteLine(result);
    // 如果有可选参数
    var options = new Dictionary<string, object>{
        {"top_num", 3}
    };
    // 带参数调用车辆识别
    result = client.CarDetect(image, options);
    Console.WriteLine(result);
}
```

车辆识别 [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
top_num	否	string	返回预测得分top结果数，默认为5

车辆识别 [返回数据参数详情](#)

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6

参数	类型	是否必须	说明
+score	否	number	置信度，示例： 0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
    {
      "name": "宝马||X5",
      "score": 0.3136293888092
    },
    {
      "name": "宝马||X4",
      "score": 0.13881112635136
    }
  ]
}
```

3.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

```
public void LogoSearchDemo() {
  var image = File.ReadAllBytes("图片文件路径");
  // 调用logo商标识别，可能会抛出网络等异常，请使用try/catch捕获
  var result = client.LogoSearch(image);
  Console.WriteLine(result);
  // 如果有可选参数
  var options = new Dictionary<string, object>{
    {"custom_lib", "true"}
  };
  // 带参数调用logo商标识别
  result = client.LogoSearch(image, options);
}
```

```
Console.WriteLine(result);
}
```

logo商标识别 [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
custom_lib	否	string	true - 返回自定义库 false - 返回自定义库+默认库的识别结果	false	是否只使用自定义logo库的结果，默认false：返回自定义库+默认库的识别结果

logo商标识别 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东

参数	类型	是否必须	说明	示例
+probability	number	是	分类结果置信度（0-1.0）	0.8
+type	number	是	type=0为1千种高优商标识别结果;type=1为2万类logo库的结果；其它type为自定义logo库结果	1（2万类的结果）

logo商标识别 返回示例

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
      "type": 0,
      "name": "科颜氏",
      "probability": 0.99998807907104,
      "location": {
        "width": 296,
        "top": 20,
        "height": 128,
        "left": 23
      }
    }
  ]
}
```

3.3.4 logo商标识别—添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
public void LogoAddDemo() {
  var image = File.ReadAllBytes("图片文件路径");
  var brief = "{\\\"name\\\": \\\"宝马\\\",\\\"code\\\":\\\"666\\\"}";
}
```

```
// 调用logo商标识别-添加，可能会抛出网络等异常，请使用try/catch捕获
var result = client.LogoAdd(image, brief);
Console.WriteLine(result);
}
```

logo商标识别-添加 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
brief	是	string	brief，检索时带回。此处要传对应的 name 与 code 字段，name长度小于100B，code长度小于150B

logo商标识别-添加 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

logo商标识别-添加 返回示例

```
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

3.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
public void LogoDeleteByImageDemo() {
  var image = File.ReadAllBytes("图片文件路径");
}
```



```
// 调用删除logo商标，传入参数为图片，可能会抛出网络等异常，请使用try/catch捕获
var result = client.LogoDeleteByImage(image);
Console.WriteLine(result);
}

public void LogoDeleteBySignDemo() {
var contSign = "8cnn32frvrr2cd901";

// 调用删除logo商标，传入参数为图片签名，可能会抛出网络等异常，请使用try/catch捕获
var result = client.LogoDeleteBySign(contSign);
Console.WriteLine(result);
}
```

logo商标识别-删除 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
cont_sign	是	string	图片签名（和image二选一，image优先级更高）

logo商标识别-删除 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

logo商标识别-删除 返回示例

```
{
  "log_id": 50488417
}
```

3.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

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

```
// 调用动物识别,可能会抛出网络等异常,请使用try/catch捕获
var result = client.AnimalDetect(image);
Console.WriteLine(result);
// 如果有可选参数
var options = new Dictionary<string, object>{
    {"top_num", 3}
};
// 带参数调用动物识别
result = client.AnimalDetect(image, options);
Console.WriteLine(result);
}
```

动物识别 请求参数详情

参数名称	是否必选	类型	默认值	说明
image	是	byte[]		二进制图像数据
top_num	否	string	6	返回预测得分top结果数,默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id,用于问题定位
result	array(object)	是	识别结果数组
+name	string	是	动物名称,示例:蒙古马
+score	number	是	置信度,示例:0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "蒙古马",
      "score": 0.41471540927887
    },
  ],
}
```

```
{
  "name": "伊犁马",
  "score": 0.39988899230957
},
{
  "name": "哈萨克马",
  "score": 0.39988899230957
}
...
]
```

3.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

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

植物识别 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

植物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	arry(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃
+score	number	是	置信度，示例：0.5321

[植物识别](#) [返回示例](#)

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

3.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。

```
public void ObjectDetectDemo() {
  var image = File.ReadAllBytes("图片文件路径");
  // 调用图像主体检测，可能会抛出网络等异常，请使用try/catch捕获
  var result = client.ObjectDetect(image);
  Console.WriteLine(result);
  // 如果有可选参数
  var options = new Dictionary<string, object>{
    {"with_face", 0}
  };
  // 带参数调用图像主体检测
  result = client.ObjectDetect(image, options);
  Console.WriteLine(result);
}
```

[图像主体检测](#) [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
with_face	否	string	0 - 不带人脸区域 1 - 带人脸区域	1	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id，用于问题定位	507499361
result	object	是	裁剪结果	-
+left	number	是	表示定位位置的长方形左上顶点的水平坐标	50
+top	number	是	表示定位位置的长方形左上顶点的垂直坐标	60
+width	number	是	表示定位位置的长方形的宽度	200
+height	number	是	表示定位位置的长方形的高度	200

图像主体检测 [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

3.3.9 FAQ

1. throw exception “fail to fetch token: 基础连接已关闭”

- 检查网络连接
- 检查网络是否有代理

2. SSL报错 “The authentication or decryption has failed” 可能由于网络代理等原因导致证书不正确，属于常见的网络问题，可以参考[这个答案](#)

3.4 错误信息

3.4.1 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 and 解决发生的错误。

3.4.2 错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用

错误码	错误信息	描述
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
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（649285136）或工单联系技术支持团队。
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片

错误码	错误信息	描述
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（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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
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

第4章 Java SDK文档

4.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对Java开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

4.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

4.1.2 版本更新记录

上线日期	版本号	更新内容
2017.12.22	4.0.0	接口统一升级
2017.10.18	3.2.1	使用proxy问题修复

上线日期	版本号	更新内容
2017.9.27	3.2.0	新增动物、植物识别接口
2017.9.7	3.1.0	新增图像识别服务

4.2 快速入门

4.2.1 安装ImageClassify Java SDK

ImageClassify 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
├── imageclassify
│   └── AipImageClassify //AipImageClassify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](#)[log4j-1.2.17.jar](#)。

其中，`version`为版本号，添加完成后，用户就可以在工程中使用ImageClassify Java SDK。

4.2.2 新建AipImageClassifyClient

AipImageClassifyClient是图像识别的Java客户端，为使用图像识别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AipImageClassifyClient,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_AppID";
    public static final String API_KEY = "你的_ApiKey";
    public static final String SECRET_KEY = "你的_SecretKey";

    public static void main(String[] args) {
        // 初始化一个AipImageClassifyClient
        AipImageClassifyClient client = new AipImageClassifyClient(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，和二选一，或者均不设置httpsocket
        client.setHttpProxy("proxy_host", proxy_port); // 设置代理http
        client.setSocketProxy("proxy_host", proxy_port); // 设置代理socket

        // 可选：设置日志输出格式，若不设置，则使用默认配置log4j
        // 也可以直接通过启动参数设置此环境变量jvm
        System.setProperty("aip.log4j.conf", "path/to/your/log4j.properties");

        // 调用接口
        String path = "test.jpg";
        JSONObject res = client.objectDetect(path, new HashMap<String, String>());
        System.out.println(res.toString(2));
    }
}
```

其中示例的log4j.properties文件内容如下：

```
## 可以设置级别：debug>info>error
```

```
## debug: 显示debug、info、error
```

```
## info: 显示info、error

## error: 只error
log4j.rootLogger=debug,appender1

## log4j.rootLogger=info,appender1

## log4j.rootLogger=error,appender1


## 输出到控制台
log4j.appender.appender1=org.apache.log4j.ConsoleAppender

## 样式为TTCCLayout
log4j.appender.appender1.layout=org.apache.log4j.PatternLayout

## 自定义样式

## %r 时间 0

## %t 方法名 main

## %p 优先级 DEBUG/INFO/ERROR

## %c 所属类的全名(包括包名)

## %l 发生的位置,在某个类的某行

## %m 输出代码中指定的讯息,如log(message)中的message

## %n 输出一个换行

log4j.appender.appender1.layout.ConversionPattern=[%d{yy/MM/dd HH:mm:ss:SSS}][%t]
[%p] -%l %m%n
```

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”。

4.2.3 配置AipImageClassify

如果用户需要配置AipImageClassify的一些细节参数，可以在构造AipImageClassify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）

4.3 接口说明

4.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("top_num", "3");

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.dishDetect(image, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.dishDetect(file, options);
    System.out.println(res.toString(2));
}
```

[菜品识别 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
top_num	否	String	返回预测得分top结果数，默认为5

菜品识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	是	number	唯一的log id，用于问题定位
result_num	否	number	返回结果数目，及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	number	卡路里，每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值，0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤（串类）",
      "calorie": 83.12,
      "probability": 0.35874313116074
    },
    {
      "name": "鱿鱼",
      "calorie": 402.12,
      "probability": 0.20610593259335
    },
    {
      "name": "板筋",
```

```
        "calorie": 333.55,
        "probability": 0.15860831737518
    },
    {
        "name": "鸡脆骨",
        "calorie": 661.33,
        "probability": 0.077698558568954
    },
    {
        "name": "麻辣烫",
        "calorie": 273.33,
        "probability": 0.041968926787376
    }
]
}
```

4.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("top_num", "3");

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.carDetect(image, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.carDetect(file, options);
    System.out.println(res.toString(2));
}
```

[车辆识别 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
top_num	否	String	返回预测得分top结果数，默认为5

车辆识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	number	置信度，示例：0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
    {
      "name": "宝马||X5",
      "score": 0.3136293888092
    },
    {
      "name": "宝马||X4",
      "score": 0.13881112635136
    }
  ]
}
```


4.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("custom_lib", "false");

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.logoSearch(image, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.logoSearch(file, options);
    System.out.println(res.toString(2));
}
```

logo商标识别 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
custom_lib	否	String	true - 返回自定义库 false - 返回自定义库+默认库的识别结果	false	是否只使用自定义logo库的结果，默认false：返回自定义库+默认库的识别结果

logo商标识别 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东
+probability	number	是	分类结果置信度（0-1.0）	0.8
+type	number	是	type=0为1千种高优商标识别结果；type=1为2万类logo库的结果；其它type为自定义logo库结果	1（2万类的结果）

logo商标识别 返回示例

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
```

```
        "type": 0,
        "name": "科颜氏",
        "probability": 0.99998807907104,
        "location": {
            "width": 296,
            "top": 20,
            "height": 128,
            "left": 23
        }
    }
}
```

4.3.4 logo商标识别—添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    String brief = "{\\\"name\\\": \\\"宝马\\\",\\\"code\\\":\\\"666\\\"}";

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.logoAdd(image, brief, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.logoAdd(file, brief, options);
    System.out.println(res.toString(2));
}
```

logo商标识别—添加 [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

参数名称	是否必选	类型	说明
brief	是	String	brief，检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B

logo商标识别-添加 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

logo商标识别-添加 返回示例

```
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

4.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.logoDeleteByImage(image, options);
    System.out.println(res.toString(2));
}
```

```
// 参数为本地图片二进制数组
byte[] file = readImageFile(image);
res = client.logoDeleteByImage(file, options);
System.out.println(res.toString(2));

// 删除logo商标，传入参数为图片签名
JSONObject res = client.logoDeleteBySign(contSign, options);
System.out.println(res.toString(2));

}
```

logo商标识别-删除 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
cont_sign	是	String	图片签名（和image二选一，image优先级更高）

logo商标识别-删除 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

logo商标识别-删除 返回示例

```
{
  "log_id": 50488417
}
```

4.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();
    options.put("top_num", "3");

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.animalDetect(image, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.animalDetect(file, options);
    System.out.println(res.toString(2));
}
```

动物识别 请求参数详情

参数名称	是否必选	类型	默认值	说明
image	是	mixed		本地图片路径或者图片二进制数据
top_num	否	String	6	返回预测得分top结果数，默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	number	是	置信度，示例：0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "蒙古马",
      "score": 0.41471540927887
    },
    {
      "name": "伊犁马",
      "score": 0.39988899230957
    },
    {
      "name": "哈萨克马",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

4.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

```
public void sample(AipImageClassify client) {
    // 传入可选参数调用接口
    HashMap<String, String> options = new HashMap<String, String>();

    // 参数为本地图片路径
    String image = "test.jpg";
    JSONObject res = client.plantDetect(image, options);
    System.out.println(res.toString(2));

    // 参数为本地图片二进制数组
    byte[] file = readImageFile(image);
    res = client.plantDetect(file, options);
    System.out.println(res.toString(2));
}
```

植物识别 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

植物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于 问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例： 吉娃娃
+score	number	是	置信度，示例： 0.5321

植物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

4.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。


```
public void sample(AipImageClassify client) {  
    // 传入可选参数调用接口  
    HashMap<String, String> options = new HashMap<String, String>();  
    options.put("with_face", "0");  
  
    // 参数为本地图片路径  
    String image = "test.jpg";  
    JSONObject res = client.objectDetect(image, options);  
    System.out.println(res.toString(2));  
  
    // 参数为本地图片二进制数组  
    byte[] file = readImageFile(image);  
    res = client.objectDetect(file, options);  
    System.out.println(res.toString(2));  
}
```

图像主体检测 [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
with_face	否	String	0 - 不带人脸区域1 - 带人脸区域	1	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id, 用于问题定位	507499361
result	object	是	裁剪结果	-
+left	number	是	表示定位位置的长方形左上顶点的水平坐标	50
+top	number	是	表示定位位置的长方形左上顶点的垂直坐标	60
+width	number	是	表示定位位置的长方形的宽度	200
+height	number	是	表示定位位置的长方形的高度	200

[图像主体检测](#) [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

4.4 错误信息

4.4.1 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 and 解决发生的错误。

4.4.2 错误码

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
SDK101	image length error	图片边长不符合要求
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
SDK109	unsupported image format	不支持的图片格式

服务端返回的错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
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（649285136）或工单联系技术支持团队。
216100	invalid param	请求中包含非法参数，请检查后重新尝试

错误码	错误信息	描述
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
282003	missing parameters: {参数名}	请求参数缺失

错误码	错误信息	描述
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
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

第5章 Nodejs SDK文档

5.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对Nodejs开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

5.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

5.1.2 版本更新记录

上线日期	版本号	更新内容
2017.12.21	2.0.0	实现代码重构, 接口返回标准promise对象

上线日期	版本号	更新内容
2017.9.27	1.4.0	图像识别服务新增动物识别，植物识别接口
2017.9.7	1.3.0	新增图像识别服务

5.2 快速入门

5.2.1 安装图像识别 Node SDK

图像识别 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
├─ AipImageClassify.js     //图像识别交互类
├─ 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图像识别 Node SDK。

直接使用npm安装依赖：

```
{\color{emcolor}\textbf{npm install baidu-aip-sdk}}
```

5.2.2 新建AipImageClassifyClient

AipImageClassifyClient是图像识别的node客户端，为使用图像识别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AipImageClassifyClient：

```
var AipImageClassifyClient = require("baidu-aip-sdk").imageClassify;

// 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 AipImageClassifyClient(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”。

5.3 接口说明

5.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用菜品识别
client.dishDetect(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["top_num"] = "3";

// 带参数调用菜品识别
client.dishDetect(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

[菜品识别 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
top_num	否	string	返回预测得分top结果数，默认为5

菜品识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	是	number	唯一的log id，用于问题定位
result_num	否	number	返回结果数目，及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	number	卡路里，每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值，0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤（串类）",
      "calorie": 83.12,
      "probability": 0.35874313116074
    },
    {
      "name": "鱿鱼",
```

```
        "calorie": 402.12,
        "probability": 0.20610593259335
    },
    {
        "name": "板筋",
        "calorie": 333.55,
        "probability": 0.15860831737518
    },
    {
        "name": "鸡脆骨",
        "calorie": 661.33,
        "probability": 0.077698558568954
    },
    {
        "name": "麻辣烫",
        "calorie": 273.33,
        "probability": 0.041968926787376
    }
]
}
```

5.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用车辆识别
client.carDetect(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["top_num"] = "3";

// 带参数调用车辆识别
```

```
client.carDetect(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

车辆识别 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
top_num	否	string	返回预测得分top结果数，默认为5

车辆识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	number	置信度，示例：0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
  ],
}
```

```
{
  "name": "宝马||X5",
  "score": 0.3136293888092
},
{
  "name": "宝马||X4",
  "score": 0.13881112635136
}
]
```

5.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用logo商标识别
client.logoSearch(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["custom_lib"] = "false";

// 带参数调用logo商标识别
client.logoSearch(image, options).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});;
```

[logo商标识别 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
custom_lib	否	string	true - 返回自定义库 false - 返回自定义库+默认库的识别结果	false	是否只使用自定义 logo 库的结果，默认 false：返回自定义库+默认库的识别结果

logo商标识别 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result 数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的 logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100

参数	类型	是否必须	说明	示例
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东
+probability	number	是	分类结果置信度 (0-1.0)	0.8
+type	number	是	type=0为1千种高优商标识别结果;type=1 为 2 万 类 logo 库 的结果；其它 type 为 自定义 logo库结果	1 （ 2万类的结果 ）

logo商标识别 返回示例

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
      "type": 0,
      "name": "科颜氏",
      "probability": 0.99998807907104,
      "location": {
        "width": 296,
        "top": 20,
        "height": 128,
        "left": 23
      }
    }
  ]
}
```

5.3.4 logo商标识别—添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");
var brief = "{\"name\\\": \"\\\"宝马\\\",\\\"code\\\":\\\"666\\\"}\"";

// 调用logo商标识别-添加
client.logoAdd(image, brief).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});
```

logo商标识别-添加 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
brief	是	string	brief，检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B

logo商标识别-添加 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

logo商标识别-添加 返回示例


```
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

5.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用删除logo商标，传入参数为图片
client.logoDeleteByImage(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});

var contSign = "8cnn32frvrr2cd901";

// 调用删除logo商标，传入参数为图片签名
client.logoDeleteBySign(contSign).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});
```

logo商标识别-删除 [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

参数名称	是否必选	类型	说明
cont_sign	是	string	图片签名（和image二选一，image优先级更高）

logo商标识别-删除 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

logo商标识别-删除 返回示例

```
{
  "log_id": 50488417
}
```

5.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用动物识别
client.animalDetect(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["top_num"] = "3";

// 带参数调用动物识别
client.animalDetect(image, options).then(function(result) {
```

```
        console.log(JSON.stringify(result));
    }).catch(function(err) {
        // 如果发生网络错误
        console.log(err);
    });
```

动物识别 请求参数详情

参数名称	是否必选	类型	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
top_num	否	string	6	返回预测得分 top结果数，默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	number	是	置信度，示例：0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
```

```
      "name": "蒙古马",
      "score": 0.41471540927887
    },
    {
      "name": "伊犁马",
      "score": 0.39988899230957
    },
    {
      "name": "哈萨克马",
      "score": 0.39988899230957
    }
  ]
}
```

5.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");

// 调用植物识别
client.plantDetect(image).then(function(result) {
  console.log(JSON.stringify(result));
}).catch(function(err) {
  // 如果发生网络错误
  console.log(err);
});
```

植物识别 [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

植物识别 [返回数据参数详情](#)

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃
+score	number	是	置信度，示例：0.5321

植物识别 [返回示例](#)

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

5.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。

```
var fs = require('fs');

var image = fs.readFileSync("assets/example.jpg").toString("base64");
```

```
// 调用图像主体检测
client.objectDetect(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["with_face"] = "0";

// 带参数调用图像主体检测
client.objectDetect(image, options).then(function(result) {
    console.log(JSON.stringify(result));
}).catch(function(err) {
    // 如果发生网络错误
    console.log(err);
});;
```

[图像主体检测](#) [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

参数名称	是否必选	类型	可选值范围	默认值	说明
with_face	否	string	0 - 不带人脸区域 1 - 带人脸区域	1	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取1，带人脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id，用于问题定位	507499361
result	object	是	裁剪结果	-
+left	number	是	表示定位位置的长方形左上顶点的水平坐标	50
+top	number	是	表示定位位置的长方形左上顶点的垂直坐标	60
+width	number	是	表示定位位置的长方形的宽度	200
+height	number	是	表示定位位置的长方形的高度	200

图像主体检测 [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

5.4 错误信息

5.4.1 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 and 解决发生的错误。

5.4.2 错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期

错误码	错误信息	描述
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（649285136）或工单联系技术支持团队。
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整

错误码	错误信息	描述
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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
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

第6章 PHP SDK文档

6.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对PHP开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

6.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

6.1.2 版本更新记录

上线日期	版本号	更新内容
2017.12.22	2.0.0	SDK代码重构

上线日期	版本号	更新内容
2017.9.27	1.6.6	图像识别服务新增动物识别，植物识别接口
2017.9.7	1.6.5	新增图像识别服务

6.2 快速入门

6.2.1 安装图像识别 PHP SDK

图像识别 PHP SDK目录结构

```
├─ AipImageClassify.php          //图像识别
└─ lib
    ├─ AipHttpClient.php         //内部http请求类
    ├─ 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AipImageClassify.php以及lib/*到工程文件夹中。
- 3.引入AipImageClassify.php

6.2.2 新建AipImageClassify

AipImageClassify是图像识别的PHP SDK客户端，为使用图像识别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AipImageClassify:

```
require_once 'AipImageClassify.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 AipImageClassify(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”。

6.2.3 配置AipFace

如果用户需要配置AipFace的网络请求参数(一般不需要配置)，可以在构造AipFace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）

6.3 接口说明

6.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

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

// 调用菜品识别
\$client->dishDetect(\$image);

// 如果有可选参数
\$options = array();
\$options["top_num"] = 3;

// 带参数调用菜品识别
\$client->dishDetect(\$image, \$options);
```

菜品识别 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
top_num	否	string	返回预测得分top结果数，默认为5

菜品识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	是	number	唯一的log id，用于问题定位
result_num	否	number	返回结果数目，及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	number	卡路里，每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值，0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤（串类）",
      "calorie": 83.12,
      "probability": 0.35874313116074
    },
    {
```

```
        "name": "鱿鱼",
        "calorie": 402.12,
        "probability": 0.20610593259335
    },
    {
        "name": "板筋",
        "calorie": 333.55,
        "probability": 0.15860831737518
    },
    {
        "name": "鸡脆骨",
        "calorie": 661.33,
        "probability": 0.077698558568954
    },
    {
        "name": "麻辣烫",
        "calorie": 273.33,
        "probability": 0.041968926787376
    }
]
}
```

6.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

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

// 调用车辆识别
\$client->carDetect(\$image);

// 如果有可选参数
\$options = array();
\$options["top_num"] = 3;

// 带参数调用车辆识别
\$client->carDetect(\$image, \$options);
```

[车辆识别 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
top_num	否	string	返回预测得分top结果数，默认为5

车辆识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	number	置信度，示例：0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
    {
      "name": "宝马||X5",
      "score": 0.3136293888092
    },
    {
      "name": "宝马||X4",
      "score": 0.13881112635136
    }
  ]
}
```


6.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

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

// 调用logo商标识别
\$client->logoSearch(\$image);

// 如果有可选参数
\$options = array();
\$options["custom_lib"] = "true";

// 带参数调用logo商标识别
\$client->logoSearch(\$image, \$options);
```

logo商标识别 [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
custom_lib	否	string	true - 返回自定义库 false - 返回自定义库+默认库的识别结果	false	是否只使用自定义 logo 库的结果，默认 false：返回自定义库+默认库的识别结果

[logo商标识别](#) [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东
+probability	number	是	分类结果置信度（0-1.0）	0.8
+type	number	是	type=0为1千种高优商标识别结果;type=1为2万类logo库的结果；其它type为自定义logo库结果	1（2万类的结果）

[logo商标识别](#) [返回示例](#)

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
```

```
{
  "type": 0,
  "name": "科颜氏",
  "probability": 0.99998807907104,
  "location": {
    "width": 296,
    "top": 20,
    "height": 128,
    "left": 23
  }
}
```

6.3.4 logo商标识别-添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
\$image = file_get_contents('example.jpg');
\$brief = "{\\"name\\": \\"宝马\\",\\"code\\":\\"666\\"}";

// 调用logo商标识别-添加
\$client->logoAdd(\$image, \$brief);
```

logo商标识别-添加 [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
brief	是	string	brief，检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B

[logo商标识别-添加](#) [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

[logo商标识别-添加](#) [返回示例](#)

```
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

6.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

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

// 调用删除logo商标，传入参数为图片
\$client->logoDeleteByImage(\$image);
\$contSign = "8cnn32frvrr2cd901";

// 调用删除logo商标，传入参数为图片签名
\$client->logoDeleteBySign(\$contSign);
```

[logo商标识别-删除](#) [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
cont_sign	是	string	图片签名（和image二选一，image优先级更高）

[logo商标识别-删除](#) [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

[logo商标识别-删除](#) [返回示例](#)

```
{
  "log_id": 50488417
}
```

6.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

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

// 调用动物识别
\$client->animalDetect(\$image);

// 如果有可选参数
\$options = array();
\$options["top_num"] = 3;

// 带参数调用动物识别
```

```
\$client->animalDetect(\$image, \$options);
```

动物识别 请求参数详情

参数名称	是否必选	类型	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
top_num	否	string	6	返回预测得分 top结果数，默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	arry(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	number	是	置 信 度，示 例：0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "蒙古马",
      "score": 0.41471540927887
    },
    {
```

```
      "name": "伊犁马",
      "score": 0.39988899230957
    },
    {
      "name": "哈萨克马",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

6.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

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

// 调用植物识别
\$client->plantDetect(\$image);
```

植物识别 [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

植物识别 [返回数据参数详情](#)

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃

参数	类型	是否必须	说明
+score	number	是	置信度，示例： 0.5321

植物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

6.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。

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

// 调用图像主体检测
\$client->objectDetect(\$image);

// 如果有可选参数
\$options = array();
\$options["with_face"] = 0;

// 带参数调用图像主体检测
\$client->objectDetect(\$image, \$options);
```


图像主体检测 [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
with_face	否	string	0 - 不带人脸区域1 - 带人脸区域	1	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取 1，带人脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id，用于问题定位	507499361
result	object	是	裁剪结果	-
+left	number	是	表示定位位置的长方形左上顶点的水平坐标	50

参数	类型	是否必须	说明	示例
+top	number	是	表示定位位置的长方形左上顶点的垂直坐标	60
+width	number	是	表示定位位置的长方形的宽度	200
+height	number	是	表示定位位置的长方形的高度	200

[图像主体检测](#) [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

6.4 错误信息

6.4.1 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 and 解决发生的错误。

6.4.2 错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额

错误码	错误信息	描述
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
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（649285136）或工单联系技术支持团队。
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片

错误码	错误信息	描述
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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
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

第7章 Python SDK文档

7.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对Python开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

7.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

7.1.2 版本更新记录

上线日期	版本号	更新内容
2017.12.22	2.0.0	SDK代码重构

上线日期	版本号	更新内容
2017.9.27	1.6.6	图像识别服务新增动物识别，植物识别接口
2017.9.7	1.6.5	新增图像识别服务

7.2 快速入门

7.2.1 安装图像识别 Python SDK

图像识别 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
│   └─ imageclassify.py //图像识别
└─ 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。

7.2.2 新建AipImageClassify

AipImageClassify是图像识别的Python SDK客户端, 为使用图像识别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AipImageClassify:

```
from aip import AipImageClassify

""" 你的 APPID AK SK """
APP_ID = '你的 App ID'
API_KEY = '你的 Api Key'
SECRET_KEY = '你的 Secret Key'
```

```
client = AipImageClassify(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对应白云的“Access Key ID”，SECRET_KEY对应白云的“Access Key Secret”。

7.2.3 配置AipFace

如果用户需要配置AipFace的网络请求参数(一般不需要配置)，可以在构造AipFace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）

7.3 接口说明

7.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用菜品识别 """
client.dishDetect(image);

""" 如果有可选参数 """
options = {}
options["top_num"] = 3
```

```
""" 带参数调用菜品识别 """
client.dishDetect(image, options)
```

菜品识别 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
top_num	否	string	返回预测得分top结果数，默认为5

菜品识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	是	number	唯一的log id，用于问题定位
result_num	否	number	返回结果数目，及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	number	卡路里，每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值，0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
```



```

    {
        "name": "烧烤(串类)",
        "calorie": 83.12,
        "probability": 0.35874313116074
    },
    {
        "name": "鱿鱼",
        "calorie": 402.12,
        "probability": 0.20610593259335
    },
    {
        "name": "板筋",
        "calorie": 333.55,
        "probability": 0.15860831737518
    },
    {
        "name": "鸡脆骨",
        "calorie": 661.33,
        "probability": 0.077698558568954
    },
    {
        "name": "麻辣烫",
        "calorie": 273.33,
        "probability": 0.041968926787376
    }
]
}

```

7.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

```

""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用车辆识别 """
client.carDetect(image);

```

```
""" 如果有可选参数 """
options = {}
options["top_num"] = 3

""" 带参数调用车辆识别 """
client.carDetect(image, options)
```

车辆识别 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
top_num	否	string	返回预测得分top结果数，默认为5

车辆识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	number	置信度，示例：0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
  ],
}
```

```
{
  "name": "宝马||X5",
  "score": 0.3136293888092
},
{
  "name": "宝马||X4",
  "score": 0.13881112635136
}
]
```

7.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

```
""" 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logo商标识别 """
client.logoSearch(image);

""" 如果有可选参数 """
options = {}
options["custom_lib"] = "true"

""" 带参数调用logo商标识别 """
client.logoSearch(image, options)
```

[logo商标识别 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
custom_lib	否	string	true - 返回自定义库 false - 返回自定义库+默认库的识别结果	false	是否只使用自定义 logo 库的结果，默认 false：返回自定义库+默认库的识别结果

logo商标识别 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result 数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的 logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100

参数	类型	是否必须	说明	示例
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东
+probability	number	是	分类结果置信度 (0-1.0)	0.8
+type	number	是	type=0为1千种高优商标识别结果;type=1 为 2 万 类 logo 库 的结果；其它 type 为自定义 logo库结果	1 （2万类的结果）

logo商标识别 返回示例

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
      "type": 0,
      "name": "科颜氏",
      "probability": 0.99998807907104,
      "location": {
        "width": 296,
        "top": 20,
        "height": 128,
        "left": 23
      }
    }
  ]
}
```

7.3.4 logo商标识别—添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')
brief = "{\\"name\\": \\"宝马\\",\\"code\\":\\"666\\"}"

""" 调用logo商标识别-添加 """
client.logoAdd(image, brief);
```

logo商标识别-添加 [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
brief	是	string	brief，检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B

logo商标识别-添加 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

logo商标识别-添加 [返回示例](#)

```
{
  "log_id": 972934653,
```

```
"cont_sign": "217113248,419422301"
}
```

7.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用删除logo商标，传入参数为图片 """
client.logoDeleteByImage(image);

contSign = "8cnn32frvrr2cd901"

""" 调用删除logo商标，传入参数为图片签名 """
client.logoDeleteBySign(contSign);
```

logo商标识别-删除 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
cont_sign	是	string	图片签名（和image二选一，image优先级更高）

logo商标识别-删除 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

[logo商标识别-删除](#) [返回示例](#)

```
{
  "log_id": 50488417
}
```

7.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用动物识别 """
client.animalDetect(image);

""" 如果有可选参数 """
options = {}
options["top_num"] = 3

""" 带参数调用动物识别 """
client.animalDetect(image, options)
```

[动物识别](#) [请求参数详情](#)

参数名称	是否必选	类型	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
top_num	否	string	6	返回预测得分 top结果数，默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	arry(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	number	是	置信度，示例：0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "蒙古马",
      "score": 0.41471540927887
    },
    {
      "name": "伊犁马",
      "score": 0.39988899230957
    },
    {
      "name": "哈萨克马",
```

```
        "score": 0.39988899230957
    }
    ...
]
}
```

7.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用植物识别 """
client.plantDetect(image);
```

植物识别 [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

植物识别 [返回数据参数详情](#)

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃

参数	类型	是否必须	说明
+score	number	是	置信度，示例： 0.5321

植物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

7.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。

```
""" 读取图片 """
def get_file_content(filePath):
    with open(filePath, 'rb') as fp:
        return fp.read()

image = get_file_content('example.jpg')

""" 调用图像主体检测 """
client.objectDetect(image);

""" 如果有可选参数 """
options = {}
```

```
options["with_face"] = 0

""" 带参数调用图像主体检测 """
client.objectDetect(image, options)
```

图像主体检测 [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
with_face	否	string	0 - 不带人脸区域1 - 带人脸区域	1	如果检测主体是人，主体区域是否带上人脸部分，0-不带人脸区域，其他-带人脸区域，裁剪类需求推荐带人脸，检索/识别类需求推荐不带人脸。默认取 1，带人脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id，用于问题定位	507499361
result	object	是	裁剪结果	-

参数	类型	是否必须	说明	示例
+left	number	是	表示定位位置的长方形左上顶点的水平坐标	50
+top	number	是	表示定位位置的长方形左上顶点的垂直坐标	60
+width	number	是	表示定位位置的长方形的宽度	200
+height	number	是	表示定位位置的长方形的高度	200

[图像主体检测](#) [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

7.4 错误信息

7.4.1 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 and 解决发生的错误。

7.4.2 错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超额额
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超额额
18	Open api qps request limit reached	QPS超额额
19	Open api total request limit reached	请求总量超额额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
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（649285136）或工单联系技术支持团队。
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片

错误码	错误信息	描述
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片
216630	recognize error	识别错误，请再次请求，如果持续出现此类错误，请通过QQ群（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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
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

第8章 C++ SDK文档

8.1 简介

Hi, 您好, 欢迎使用百度图像识别服务。

本文档主要针对C++开发者, 描述百度图像识别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;
- 如有疑问, 进入[AI社区交流](http://ai.baidu.com/forum/topic/list/171): <http://ai.baidu.com/forum/topic/list/171>

8.1.1 接口能力

接口名称	接口能力简要描述
菜品识别	检测用户上传的菜品图片, 返回具体的菜名、卡路里、置信度信息。
车型识别	检测用户上传的车辆图片, 识别所属车型, 包括车辆品牌及具体型号。
logo商标识别	识别图片中包含的商品LOGO信息, 返回LOGO品牌名称、在图片中的位置、置信度。
动物识别	检测用户上传的动物图片, 返回动物名称、置信度信息。
植物识别	检测用户上传的植物图片, 返回植物名称、置信度信息。
图像主体检测	识别图像中的主体具体坐标位置。

8.1.2 版本更新记录

上线日期	版本号	更新内容
2017.11.24	0.3.2	修复windows平台VC环境的编译错误

上线日期	版本号	更新内容
2017.11.9	0.3.0	初始化参数修改
2017.10.31	0.1.0	图像识别第一版

8.2 快速入门

8.2.1 安装图像识别 C++ SDK

图像识别 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
└─ imageclassify.h       // 图像识别 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 `imageclassify.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。

8.2.2 新建client

client是图像识别的C++客户端，为使用图像识别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 "imageclassify.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::Imageclassify 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对应百度云的“Access Key ID”，SECRET_KEY对应百度云的“Access Key Secret”。

8.3 接口说明

8.3.1 菜品识别

该请求用于菜品识别。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的菜品名称、卡路里信息、置信度。

```
Json::Value result;

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

// 调用菜品识别
result = client.dish_detect(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["top_num"] = "3";

// 带参数调用菜品识别
result = client.dish_detect(image, options);
```

[菜品识别 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
top_num	否	std::string	返回预测得分top结果数，默认为5

菜品识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	是	number	唯一的log id，用于问题定位
result_num	否	number	返回结果数目，及result数组中的元素个数
result	否	array(object)	菜品识别结果数组
+name	否	string	菜名，示例：鱼香肉丝
+calorie	否	number	卡路里，每100g的卡路里含量
+probability	否	number	识别结果中每一行的置信度值，0-1

菜品识别 返回示例

```
{
  "log_id": 271468280,
  "result_num": 5,
  "result": [
    {
      "name": "烧烤(串类)",
      "calorie": 83.12,
      "probability": 0.35874313116074
    },
    {
      "name": "鱿鱼",
      "calorie": 402.12,
      "probability": 0.20610593259335
    },
  ],
}
```

```
{
  {
    "name": "板筋",
    "calorie": 333.55,
    "probability": 0.15860831737518
  },
  {
    "name": "鸡脆骨",
    "calorie": 661.33,
    "probability": 0.077698558568954
  },
  {
    "name": "麻辣烫",
    "calorie": 273.33,
    "probability": 0.041968926787376
  }
}
```

8.3.2 车辆识别

该请求用于检测一张车辆图片的具体车型。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片的车辆品牌及型号。

```
Json::Value result;

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

// 调用车辆识别
result = client.car_detect(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["top_num"] = "3";

// 带参数调用车辆识别
result = client.car_detect(image, options);
```

[车辆识别 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
top_num	否	std::string	返回预测得分top结果数，默认为5

车辆识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	否	number	唯一的log id，用于问题定位
result	否	array(object)	车型识别结果数组
+name	否	string	车型名称，示例：宝马x6
+score	否	number	置信度，示例：0.5321

车辆识别 返回示例

```
{
  "log_id": 2399742249,
  "result": [
    {
      "name": "宝马X6",
      "score": 0.40009739995003
    },
    {
      "name": "宝马||X5",
      "score": 0.3136293888092
    },
    {
      "name": "宝马||X4",
      "score": 0.13881112635136
    }
  ]
}
```

8.3.3 logo商标识别

该请求用于检测和识别图片中的品牌LOGO信息。即对于输入的一张图片（可正常解码，且长宽比适宜），输出图片中LOGO的名称、位置和置信度。当效果欠佳时，可以建立子库（请加入QQ群：649285136 联系工作人员申请建库）并自定义logo入库，提高识别效果。

```
Json::Value result;

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

// 调用logo商标识别
result = client.logo_search(image, aip::null);

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

// 带参数调用logo商标识别
result = client.logo_search(image, options);
```

logo商标识别 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
custom_lib	否	std::string	true - 返回 自 定 义 库 false - 返回 自 定 义 库 +默 认 库 的 识别结果	false	是否只使用 自定义 logo 库的结果， 默认 false： 返回自定义 库 +默 认 库 的识别结果

logo商标识别 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
result_num	number	是	返回结果数目，即：result数组中元素个数	2
result	array	是	返回结果数组，每一项为一个识别出的logo	-
+location	object	是	位置信息（左起像素位置、上起像素位置、像素宽、像素高）	{ "left" : 100, "top" : 100, "width" : 10, "height" : 10 }
++left	number	是	左起像素位置	100
++top	number	是	上起像素位置	100
++width	number	是	像素宽	100
++height	number	是	像素高	100
+name	string	是	识别的品牌名称	京东
+probability	number	是	分类结果置信度（0-1.0）	0.8
+type	number	是	type=0为1千种高优商标识别结果;type=1为2万类logo库的结果；其它type为自定义logo库结果	1（2万类的结果）

logo商标识别 返回示例

```
{
  "log_id": 843411868,
  "result_num": 1,
  "result": [
    {
```

```
        "type": 0,
        "name": "科颜氏",
        "probability": 0.99998807907104,
        "location": {
            "width": 296,
            "top": 20,
            "height": 128,
            "left": 23
        }
    }
]
```

8.3.4 logo商标识别-添加

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
Json::Value result;

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

std::string brief = "{\"name\\\": \\\"宝马\\\",\\\"code\\\":\\\"666\\\"}";

// 调用logo商标识别-添加
result = client.logo_add(image, brief, aip::null);
```

logo商标识别-添加 [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
brief	是	std::string	brief，检索时带回。此处要传对应的name与code字段，name长度小于100B，code长度小于150B

logo商标识别-添加 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361
cont_sign	string	是	输入图片签名，可用于删除	-

logo商标识别-添加 返回示例

```
{
  "log_id": 972934653,
  "cont_sign": "217113248,419422301"
}
```

8.3.5 logo商标识别-删除

该接口尚在邀测阶段，使用该接口之前需要线下联系工作人员完成建库方可使用，请加入QQ群：649285136 联系相关人员。

```
Json::Value result;

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

// 调用删除logo商标，传入参数为图片
result = client.logo_delete_by_image(image, aip::null);

Json::Value result;

std::string cont_sign = "8cnn32frvrr2cd901";

// 调用删除logo商标，传入参数为图片签名
result = client.logo_delete_by_sign(cont_sign, aip::null);
```

logo商标识别-删除 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
cont_sign	是	std::string	图片签名（和image二选一，image优先级更高）

logo商标识别-删除 返回数据参数详情

参数	类型	是否必须	说明	示例
log_id	number	是	请求标识码，随机数，唯一	507499361

logo商标识别-删除 返回示例

```
{
  "log_id": 50488417
}
```

8.3.6 动物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出动物识别结果

```
Json::Value result;

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

// 调用动物识别
result = client.animal_detect(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["top_num"] = "3";

// 带参数调用动物识别
```

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

动物识别 请求参数详情

参数名称	是否必选	类型	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
top_num	否	std::string	6	返回预测得分top结果数，默认为6

动物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	识别结果数组
+name	string	是	动物名称，示例：蒙古马
+score	number	是	置 信 度，示 例：0.5321

动物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "蒙古马",
      "score": 0.41471540927887
    },
    {
      "name": "伊犁马",
      "score": 0.39988899230957
    },
    {
      "name": "哈萨克马",
```

```
        "score": 0.39988899230957
    }
    ...
]
}
```

8.3.7 植物识别

该请求用于识别一张图片。即对于输入的一张图片（可正常解码，且长宽比适宜），输出植物识别结果。

```
Json::Value result;

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

// 调用植物识别
result = client.plant_detect(image, aip::null);
```

植物识别 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

植物识别 返回数据参数详情

参数	类型	是否必须	说明
log_id	number	是	唯一的log id，用于问题定位
result	array(object)	是	植物识别结果数组
+name	string	是	植物名称，示例：吉娃娃
+score	number	是	置信度，示例：0.5321

植物识别 返回示例

```
{
  "log_id": 304829369,
  "result": [
    {
      "name": "吉娃娃",
      "score": 0.41471540927887
    },
    {
      "name": "番杏科",
      "score": 0.39988899230957
    },
    {
      "name": "花月夜",
      "score": 0.39988899230957
    }
    ...
  ]
}
```

8.3.8 图像主体检测

用户向服务请求检测图像中的主体位置。

```
Json::Value result;

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

// 调用图像主体检测
result = client.object_detect(image, aip::null);

// 如果有可选参数
std::map<std::string, std::string> options;
options["with_face"] = "0";

// 带参数调用图像主体检测
result = client.object_detect(image, options);
```

[图像主体检测 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
with_face	否	std::string	0 - 不带人脸 区域1 - 带人 脸区域	1	如果检测主 体是人，主 体区域是否 带上人脸部 分，0-不 带 人脸区域， 其他-带人脸 区域，裁剪 类需求推荐 带人脸，检 索/识别类需 求推荐不带 人脸。默认 取 1，带 人 脸。

图像主体检测 [返回数据参数详情](#)

参数	类型	是否必须	说明	示例
log_id	number	是	唯一的log id, 用于问题定位	507499361
result	object	是	裁剪结果	-
+left	number	是	表示定位位置 的长方形左上 顶点的水平坐 标	50
+top	number	是	表示定位位置 的长方形左上 顶点的垂直坐 标	60
+width	number	是	表示定位位置 的长方形的宽 度	200

参数	类型	是否必须	说明	示例
+height	number	是	表示定位位置的长方形的高度	200

[图像主体检测](#) [返回示例](#)

```
{
  "log_id": 895582300,
  "result": {
    "width": 486,
    "top": 76,
    "left": 134,
    "height": 394
  }
}
```

8.4 错误信息

8.4.1 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 and 解决发生的错误。

8.4.2 错误码

错误码	错误信息	描述
4	Open api request limit reached	集群超限额
14	IAM Certification failed	IAM鉴权失败，建议用户参照文档自查生成sign的方式是否正确，或换用控制台中ak sk的方式调用
17	Open api daily request limit reached	每天流量超限额

错误码	错误信息	描述
18	Open api qps request limit reached	QPS超限额
19	Open api total request limit reached	请求总量超限额
100	Invalid parameter	无效参数
110	Access token invalid or no longer valid	Access Token失效
111	Access token expired	Access token过期
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（649285136）或工单联系技术支持团队。
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，现阶段我们支持的图片格式为：PNG、JPG、JPEG、BMP，请进行转码或更换图片
216202	image size error	上传的图片大小错误，现阶段我们支持的图片大小为：base64编码后小于4M，分辨率不高于4096*4096，请重新上传图片

错误码	错误信息	描述
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（649285136）或工单联系技术支持团队。
216631	recognize bank card error	识别银行卡错误，出现此问题的原因一般为：您上传的图片非银行卡正面，上传了异形卡的图片或上传的银行卡正品图片不完整
216633	recognize idcard error	识别身份证错误，出现此问题的原因一般为：您上传了非身份证图片或您上传的身份证图片不完整
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（649285136）或工单联系技术支持团队。
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
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