印刷文字识别

文字识别

文字识别

产品简介

最近更新

- 自定义模板识别（解决个性化OCR需求）：
  https://market.aliyun.com/products/57124001/cmapi029975.html
- 模板管理入口：https://ocr.data.aliyun.com/

服务简介

- 产品地址：https://data.aliyun.com/product/ocr
- 产品列表：
  https://market.aliyun.com/products/56956004?spm=a2c0j.8222507.973835.btn1.3cd6e53284tk2A&tag=%E9%98%BF%E9%87%8C%E4%BA%91%E5%AE%98%E6%96%B9

随着智能手机和移动设备的普及，越来越多的图片被产生，也有越来越多的图片文字识别需求。典型的应用场景有证件信息的自动识别和提取，自然场景中的文字识别，文档或者宣传资料中的文字检测识别等。同时，由于深度学习和图像检测技术的发展，使得上述场景中的文字的检测和识别效果越来越好，使得机器自动识别成为可能，在业务审核中给公司节省了大量的人力。本服务包含多种场景下的文字识别

定制化能力包括：

1. 印刷文字识别-身份证识别
2. 印刷文字识别-行驶证识别
3. 印刷文字识别-驾驶证识别
4. 印刷文字识别-护照识别
5. 印刷文字识别-营业执照识别
6. 印刷文字识别-银行卡识别
7. 印刷文字识别-名片识别
8. 印刷文字识别-车牌识别
9. 印刷文字识别-vin码识别
10. 印刷文字识别-火车票识别

11. 印刷文字识别-公章识别
12. 印刷文字识别-出租车机打发票识别

通用识别能力包括：

1. 印刷文字识别-通用文字识别
2. 印刷文字识别-表格识别
3. 自定义模板识别

产品示例

身份证证件信息识别

身份证识别服务可以自动地从图片中定位身份证图片区域，识别出其中包含的身份信息，包括:

- 姓名
- 身份证号

- 性别
- 出生日期
- 有效期
- 地址

与传统的方法相比，自动识别可以极大的提高用户体验，节约审核业务的人力，提高审核的时效性和准确性。

驾驶证识别

驾驶证识别服务用于自动从图片中定位身份证图片，以及从包含驾驶证的图片中提取驾驶人员的身份信息，提取的信息包括:

- 姓名
- 身份证号
- 准驾车型
- 有效期限

营业执照识别

营业执照识别服务用于自动从图片中定位营业执照图片，识别出其中包含的企业信息，包括：

- 注册号/统一社会信用代码
- 公司名称
- 法定代表人
- 公司地址
- 营业期限
- 注册资本

目前营业执照识别服务只适用于识别新版“三证合一”营业执照，模板如样图所示。

快速入门

前言

欢迎使用OCR服务，这里主要为您介绍如何使用OCR的各种服务，如何快速找到需要的帮助信息。下文主要通过身份证识别服务的例子来介绍各个流程。

- 云市场
- 数加接口

参考示例 — 身份证识别

云市场调用

购买服务

开通API网关https://www.aliyun.com/product/apigateway

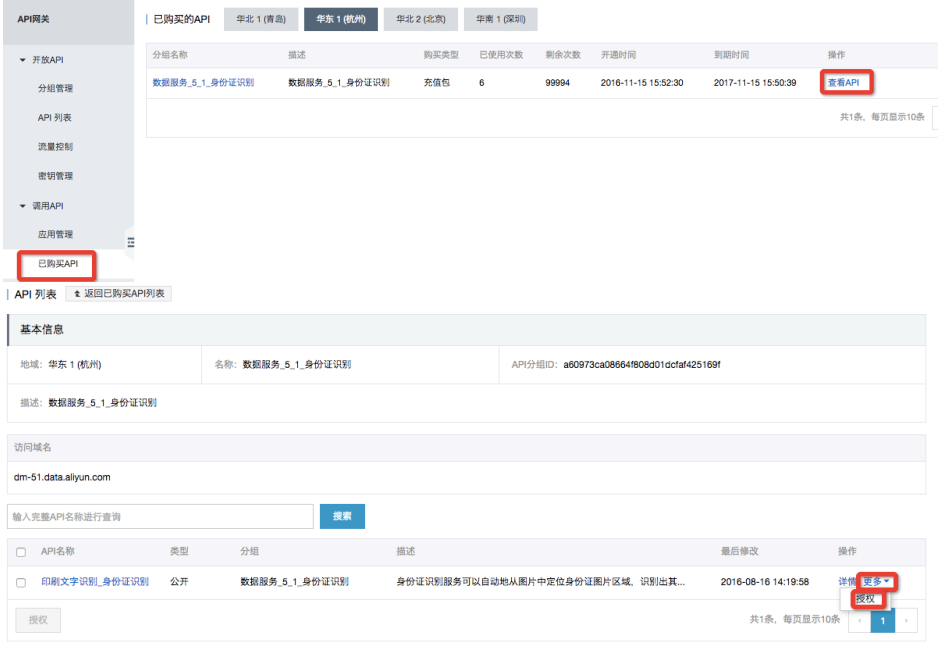
在身份证服务https://market.aliyun.com/products/57124001/cmapi010401.html 页面购买服务

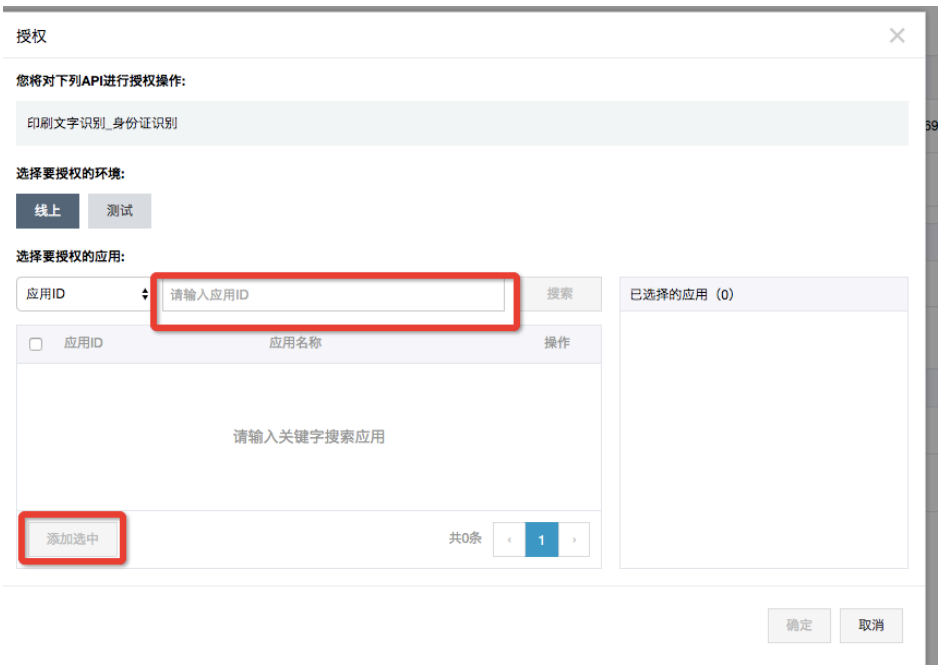
授权API

进入API网关 管理控制台，点击左侧调用API—> 应用管理，创建新应用。

应用创建后，点击应用名称，查看应用ID

点击左侧已购买API，在对应的API一行中选择查看API，点击更多,授权，输入步骤2获取的应用ID，进行授权即可。





API调用

API的具体调用方式见身份证服务产品页面

具体的示例代码见产品页面的请求示例代码，通过此页面查看APPCODE， 请阅读数据格式说明了解印刷文字识别服务的输入输出格式，阅读API介绍-身份证了解身份证服务具体的输入输出格式。

示例如下：

```
{
"image": "图片二进制数据的base64编码",
"configure": "{\"side\":\"face\"}" #身份证正反面类型:face/back
}
```

上面列出的是识别身份证正面图像的输入格式，主要是传输了图像数据和配置字符串，其中图像是经过base64编码后的数据，配置字符串主要传递了一个参数，表示当前图像为身份证正面图像，进行正面识别。

返回结果示例如下：

```
{

"address" : "浙江省杭州市余杭区文一西路969号", #地址信息
"config_str" : "{\\\"side\\\":\\\"face\\\"}", #配置信息，同输入configure
"face_rect":{
"angle": -90,
"center":{
"x" : 952,
"y" : 325.5
},
"size":{
"height":181.99,
"width":164.99
}
}, #人脸位置，center表示人脸矩形中心坐标，size表示人脸矩形长宽，angle表示矩形顺时针旋转的度数。
"name" : "张三", #姓名
"nationality": "汉" ， #民族
"num" : "1234567890", #身份证号
"sex" : "男", #性别
"birth" : "20000101", #出生日期
"nationality" : "汉", #民族
"success" : true #识别结果，true表示成功，false表示失败
}
```

```
反面返回结果:
{
"config_str" : "{\\\"side\\\":\\\"back\\\"}",#配置信息，同输入configure
"start_date" : "19700101", #有效期起始时间
"end_date" : "19800101", #有效期结束时间
"issue" : "杭州市公安局", #签发机关
"success" : true #识别结果，true表示成功，false表示失败
}
```

对应的字段含义具体可以参见API介绍-身份证。

功能示例

demo

演示demo详见 AI体验馆

API 参考

数据格式

数据格式说明

图片格式说明

目前图片支持如下格式：

- Windows bitmaps - *.bmp
- JPEG 文件 - *.jpeg,* .jpg, *.jpe
- JPEG 2000 文件 - *.jp2
- Portable Network Graphics - *.png
- Portable image format - *.pbm,* .pgm, *.ppm

- TIFF 文件 - *.tiff,* .tif

请求字符串格式说明

各识别服务API使用方法大相径庭，参数均通过http请求的body传输，格式以json字符串封装，json字段因服务而异。各服务的返回结果同样以json字符串形式保存，输入输出的具体信息请参考各项服务API说明文档。

上传参数

服务所需的参数通过http POST请求的body上传，以json字符串格式封装，而每个参数又有两个json字段，dataType和dataValue，形式如下：

```
{
param :{
"dataType": xxx # int类型
"dataValue": xxx #根据dataType定类型
}
}
```

dataType的定义如下，用户可以根据具体的数字，把dataValue转换成对应的格式存储。

```
Bool = 1;
Int32 = 10
Int64 = 20;
Float = 30;
Double = 40;
String = 50;
DateTime = 60;
```

上传请求的body示例如下：

```
"inputs": [
{
"image": {
"dataType": 50,
"dataValue": "base64_image_string" #图片以base64编码的string
},
"configure": {
"dataType": 50,
"dataValue": "{
\"arg1\" : \"arg1_value\",
\"arg2\" : \"arg2_value\"
}"
} #[可选参数]
}
```

```
]
```

输入参数是json字符串信息，关键词inputs对应一个json数组，里面可以存放多个json object，每个json object包含单个识别请求所需的参数，服务支持多机并发访问，因此可以一次上传多组参数进行识别。每一组参数则又有两个字段构成：

参数名称	参数类型	是否可选	描述	默认值
image	string	否	dataType为50(字符串)，dataValue是base64编码后的图像数据	空字符串
configure	string	是	dataType为50(字符串)，dataValue是json格式字符串，其中的具体参数字段根据不同的识别服务而定，参见API介绍	空字符串

结果解析

返回结果以json字符串格式封装，存储在以outputs为关键词对应的的数组中。 如果inputs为关键词的数组中包含多个请求参数，那么对应的outputs对应的数组中也包含同样个数的json object。每个json object有3个字段：

- outputLabel: 所用识别服务的名称
- outputMulti: 暂时未使用，为{}
- outputValue: json字符串，其中dataType 50表示数据类型为字符串，dataValue为对应的输出结果字符串，该字符串为json字符串格式，需要重新使用json解析获取每个字段的结果，其中具体的json字 段由不同的识别服务决定，可以参考API介绍页面。

```
{
"outputs": [
{
"outputLabel": "service_name",
"outputMulti": {},
"outputValue": {
"dataType": 50,
"dataValue": "{
\"outputinfo1\" : \"output1\",
\"outputinfo2\" : \"output2\"
}"
}
}
]
}
```

身份证识别

身份证证件信息识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了身份证服务调用的程序示例，以供参考。

请求接口

- 云市场接口
    - 请求方法: POST
    - 请求url: 商品售卖页https://market.aliyun.com/products/57124001/cmapi010401.html
    - 在商品售卖页的API接口中找到调用地址

输入格式

```
{
"image": "图片二进制数据的base64编码",
"configure": "{\"side\":\"face\"}" #正面/反面:face/back
}
```

输出格式

```
 正面返回结果：
{
"address" : "浙江省杭州市余杭区文一西路969号", #地址信息
"config_str" : "{\"side\":\"face\"}", #配置信息，同输入configure
"face_rect":{ #人脸位置
"angle": -90, #angle表示矩形顺时针旋转的度数
"center":{ #center表示人脸矩形中心坐标
"x" : 952,
"y" : 325.5
},
"size":{ #size表示人脸矩形长宽
"height":181.99,
"width":164.99
}
},
"card_region":[ #身份证区域位置，四个顶点表示，顺序是逆时针(左上、左下、右下、右上)
{"x":165,"y":657},
{"x":534,"y":658},
{"x":535,"y":31},
```

```
{"x":165,"y":30}
],
"face_rect_vertices":[ #人脸位置，四个顶点表示
{ "x":1024.66, "y":336.62 },
{ "x":906.66, "y":336.14},
{ "x":907.15, "y":214.14},
{"x":1025.15, "y":214.63}
],
"name" : "张三", #姓名
"nationality": "汉" ， #民族
"num" : "1234567890", #身份证号
"sex" : "男", #性别
"birth" : "20000101", #出生日期
"nationality" : "汉", #民族
"success" : true #识别结果，true表示成功，false表示失败
}


反面返回结果:
{
"config_str" : "{\"side\\\":\"back\"}", #配置信息，同输入configure
"card_region":[ #身份证区域位置，四个顶点表示，顺序是逆时针(左上、左下、右下、右上)
{ "x":212, "y":371},
{ "x":2188, "y":350},
{ "x":2201, "y":1607},
{ "x":225, "y":1627}
],
"start_date" : "19700101", #有效期起始时间
"end_date" : "19800101", #有效期结束时间
"issue" : "杭州市公安局", #签发机关
"success" : true #识别结果，true表示成功，false表示失败
}
```

驾驶证识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解**请求数据格式介绍**，了解输入输出的通用数据格式。 此外，在本页最后，附上了驾驶证识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
    - 请求方法: POST
    - 请求url: 在云市场搜索印刷文字识别-驾驶证识别,在API接口中找到调用地址

输入格式:

```
{
"image": "图片二进制数据的base64编码",
"configure": "{\"side\":\"face\"}" #首页/副页:face/back
}
```

输出格式

```
首页识别返回格式：
{
"config_str": "{\\\"side\\\": \\\"face\\\" }" #配置字符串信息
"name": "张三三", #姓名字符串，识别不出来时，可能为"NoResult"/"InvalidInput"
"num": "360502xxxx03071357", #驾驶证号，识别错误时，为"NoResult"/"InvalidInput"
"vehicle_type":"C1", #驾驶证准驾车型
"start_date": "2010xxxx", #驾驶证有效期开始时间
"end_date": "6", #驾驶证有效期时长
"addr":"北京市海淀区清华园6号楼", #地址
"sex":"男", #性别
"success": true #识别成功与否 true/false
}

副页识别返回格式：
{
"config_str": "{\\\"side\\\": \\\"back\\\" }", #配置字符串信息
"archive_no":"370211375349", #档案编号
"success": true #识别成功与否 true/false
}
```

行驶证识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解**请求数据格式介绍**，了解输入输出的通用数据格式。 此外，在本页最后，附上了驾驶证识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
    - 请求方法: POST
    - 请求url: 在云市场搜索印刷文字识别-行驶证识别,在API接口中找到调用地址

输入格式

```
{
"image": "Base64编码的字符",
"configure": "{\"side\":\"face\"}" #正反面类型face/back
```

```
}
```

输出格式

```
正面
{
"config_str": "null\n", #配置字符串信息
"plate_num": "沪A0M084", #车牌号码
"vehicle_type":"小型轿车", #车辆类型
"owner": "张三", #所有人名称
"use_character":"出租转非", #使用性质
"addr":"浙江省宁波市江东区丁街88弄", #地址
"model":"桑塔纳牌SVW7180LE1", #品牌型号
"vin" : "LSVFF66R8C2116280", #车辆识别代号
"engine_num" : "416098", #发动机号码
"register_date":"20121127", #注册日期
"issue_date":"2013-07-08", #发证日期
"request_id": "84701974fb983158_20160526100112", #请求对应的唯一表示
"success": true #识别成功与否 true/false
}

反面
{
"config_str":"{\"side\": \"back\" }", #配置字符串信息
"appproved_passenger_capacity":"5人", #核定载人数
"approved_load":"", #核定载质量
"file_no":"530100001466", #档案编号
"gross_mass":"2000kg", #总质量
"inspection_record":"检验有效期至2014年09月云A(01)", #检验记录
"overall_dimension":"4945x1845x1480mm", #外廓尺寸
"traction_mass":"", #准牵引总质量
"unladen_mass":"1505kg" #整备质量
"plate_num":"云AD8V02", #号牌号码
"success":true, #识别成功与否 true/false
"request_id":"20180131144149_c440540b20a4dc079a10680ff60b2d2a" #请求对应的唯一表示
}
```

营业执照识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了营业执照识别服务调用的程序示例，以供参考。(注：目前营业执照识别服务只适用于识别最新的竖版“三证合一”营业执照)

请求接口

- 云市场接口

- 请求方法: POST
- 请求url: 在云市场搜索印刷文字识别-营业执照识别,在API接口中找到调用地址

输入格式

```
{
"image": "对图片内容进行Base64编码"
}
```

输出格式

```
{
"config_str" : "null\n", #配置字符串信息
"angle" : float, #输入图片的角度（顺时针旋转），[ 0， 90， 180，270 ]
"reg_num" : string, #注册号，没有识别出来时返回"FailInRecognition"
"name" : string, #公司名称，没有识别出来时返回"FailInRecognition"
"person" : string, #公司法人，没有识别出来时返回"FailInRecognition"
"establish_date": string, #公司注册日期(例：证件上为"2014年04月16日"，算法返回"20140416")
"valid_period": string, #公司营业期限终止日期(例：证件上为"2014年04月16日至2034年04月15日"，算法返回
"20340415")
#当前算法将日期格式统一为输出为"年月日"(如"20391130"),并将"长期"表示为"29991231"，若证件上没有营业期限，则默认
其为"长期",返回"29991231"。
"address" : string, #公司地址，没有识别出来时返回"FailInRecognition"
"captial" : string, #注册资本，没有识别出来时返回"FailInRecognition"
"business": string, #经营范围，没有识别出来时返回"FailInRecognition"
"elbem" : string, #国徽位置［top,left,height,width］，没有识别出来时返回"FailInDetection"
"title" : string, #标题位置［top,left,height,width］，没有识别出来时返回"FailInDetection"
"stamp" : string, #印章位置［top,left,height,width］，没有识别出来时返回"FailInDetection"
"qrcode" : string, #二维码位置［top,left,height,width］，没有识别出来时返回"FailInDetection"
"success" : bool, #识别成功与否 true/false
"request_id": string
}
```

名片识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了名片识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
    - 请求方法: POST

- 请求url: 在云市场搜索印刷文字识别-名片识别,在API接口中找到调用地址

输入格式

```
{
"inputs": [
{
"image": {
"dataType": 50,
"dataValue": "Base64编码的字符"
}
}
]
}
```

输出格式

```
{
"outputs": [
{
"outputLabel": "ocr_businesscard",
"outputMulti": {},
"outputValue": {
"dataType": 50,
"dataValue": "{
\"name\": \"张三\", #姓名
\"company\": [\"阿里巴巴\", \"阿里巴巴有限公司\" ], #公司结果数组，数组可能为空
\"department\": [\"市场部\" ], #部门结果数组，数组可能为空
\"title\" : [ \"经理\" ], #职位结果数组，数组可能为空
\"tel_cell\" : [\"15234563443\"], #手机结果数组，数组可能为空
\"tel_work\": [\"057185212345\"], #座机结果数组，数组可能为空
\"addr\": [\"浙江省杭州市西湖区文一西路969号\"], #地址结果数组，数组可能为空
\"email\": [], #邮箱结果数组 ，数组可能为空
\"request_id\" : 20160822_32423dfsa23432f #请求对应的唯一表示
\"success\": true #识别成功与否 true/false
}"
}
}]
}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

银行卡识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了银行卡识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-银行卡识别,在API接口中找到调用地址

输入格式

```
{
"inputs": [
{
"image": {
"dataType": 50, #50表示image的数据类型为字符串
"dataValue": "base64_image_string" #图片以base64编码的string
}
}]
}
```

输出格式

```
{
"outputs": [
{
"outputLabel": "ocr_bankcard",
"outputMulti": {},
"outputValue": {
"dataType": 50,
"dataValue": "{
\"card_num\" : \"1234567890123456\", #银行卡卡号，可能为空
\"request_id\" : \"20170301160849_918cfcae128fc919ad6d6e3b865d2973\", #请求唯一标识，用于错误追踪
\"success\" : true #识别成功与否 true/false
}"
}
}]
}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两

种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

通用识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了通用识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-通用识别,在API接口中找到调用地址

输入格式

```
{
"image": "图片以base64编码的string"，
"configure":
"{\"min_size\" : 16, #图片中文字的最小高度，单位像素
\"output_prob\" : true}" #是否输出文字框的概率
}
```

输出格式

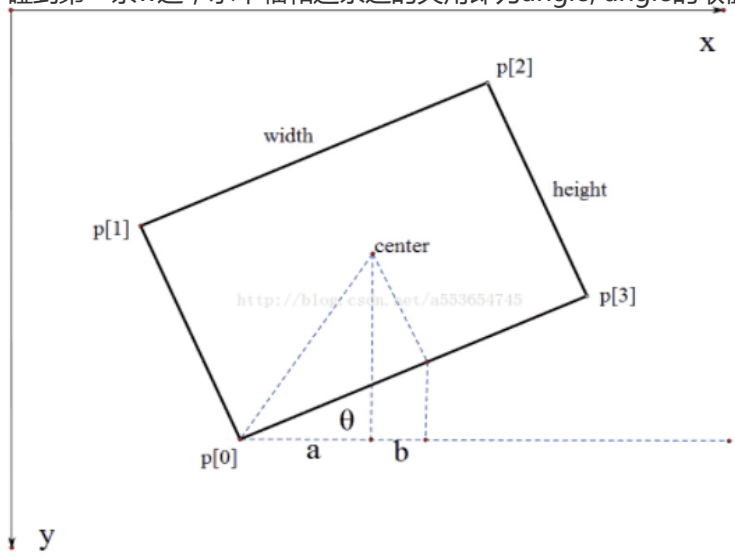
```
{
"request_id" : "20170301160849_918cfcae128fc919ad6d6e3b865d2973", #请求唯一标识，用于错误追踪

"ret":[
{
\"prob\" : 0.95983, #文字区域概率
"rect":{ #文字区域
"angle" : 0, #文字区域角度
"left" : 50, #文字区域左上角x坐标
"top" : 50, #文字区域左上角y坐标
"width" : 100, #文字区域宽度
"height" : 40 #文字区域高度
},
"word":"文字内容" #文字内容
},
{
\"prob\" : 0.95983, #文字区域概率
"rect":{ #文字区域
```

```
"angle" : 10, #文字区域角度
"left" : 50, #文字区域左上角x坐标
"top" : 50, #文字区域左上角y坐标
"width" : 100, #文字区域宽度
"height" : 40 #文字区域高度
},
"word":"文字内容" #文字内容
}
],
"success" : true
}
```

倾斜框的输出格式：

opencv中旋转矩形框格式为[x_center, y_center, w, h, angle] 注意夹角 angle 定义为，把水平轴逆时针旋转，碰到第一条w边，水平轴和这条边的夹角即为angle, angle的取值范围为(-90,0]。



通用识别输出的格式为[left, top, height, width, angle], 其中left=x_center - w/2, top=y_center - h/2,height=h, width=w, angle=angle. 可以通过left,top,height, width计算出(x_center, y_center)，然后根据上图图示画出对应的文本框区域，转换代码参考

```
def makeRectangle(width, height, theta, offset=(0,0)):
c, s = math.cos(theta), math.sin(theta)
rectCoords = [(width/2.0, height/2.0), (width/2.0, -height/2.0), (-width/2.0, -height/2.0), (-width/2.0, height/2.0)]
return [(c*x-s*y+offset[0], s*x+c*y+offset[1]) for (x,y) in rectCoords]

origin_center_x = left + width/2.0
origin_center_y = top + height/2.0
vertices = makeRectangle(width, height, angle*math.pi/180, (origin_center_x, origin_center_y))
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

护照识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了护照识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-护照识别,在API接口中找到调用地址

输入格式

```
{
"image": "base64_image_string" #图片以base64编码的string
}
```

输出格式

```
{
"authority": "公安部出入境管理局*", #签发机关
"birth_date": "19861030", #生日
"birth_day": "861030", #生日(即将弃用)
"birth_place": "广西", #出生地
"country": "CHN", #国籍
"expiry_date": "20230501", #到期日期
"expiry_day": "230501", #到期日期(即将弃用)
"issue_date": "20130502", #发证日期
"issue_place": "广西", #发证地址
"line0": "P0CHNWANG<<JING<<<<<<<<<<<<<<<<<<<<<<<<",
"line1": "E203545580CHN8610304M2305019MNPELOLIOKLPA938",
"name": "WANG.JING", #姓名英文
"name_cn": "汪婧", #姓名中文
"passport_no": "E20354xxxx", #护照号码
"person_id": "MNPELOLIOKLPA9", #持照人身份ID
"request_id": "20171120113612_813974f02a16b81ab911292d181b0b42", #请求唯一标识，用于错误追踪
1
"sex": "M", #性别
"src_country": "CHN", #国籍
"success": true,
"type": "P0" #护照类型
}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

户口页识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-户口页识别,在API接口中找到调用地址

输入格式

```
{
"image": "对图片内容进行Base64编码"
}
```

输出格式

```
{
"angle": 90, # 图片顺时针旋转角度，[ 0， 90， 180，270 ]
"name": "张三", # 姓名，没有识别出来时返回"FailInRecognition"
"gender": "男", # 性别，没有识别出来时返回"FailInRecognition"
"relation": "户主", # 户主或与户主关系，没有识别出来时返回"FailInRecognition"
"birth_place": "四川省梓潼县", # 出生地，没有识别出来时返回"FailInRecognition"
"nation": "汉", # 民族，没有识别出来时返回"FailInRecognition"
"native_place": "四川省梓潼县", # 籍贯，没有识别出来时返回"FailInRecognition"
"birth_date": "1988年8月8日", # 出生日期，没有识别出来时返回"FailInRecognition"
"idcard": "510725198808081234", # 公民身份证件编号，没有识别出来时返回"FailInRecognition"
"title": { # 标题区域位置，没有识别出来时返回"FailInDetection"
"left": 652, # 区域左上角x坐标
"top": 345, # 区域左上角y坐标
"height": 345, # 区域高度
"width": 57 # 区域宽度
},
```

```
"invalid_stamp":[ # 注销/作废章位置列表
{
"left": 315, # 区域左上角x坐标
"top": 698, # 区域左上角y坐标
"height": 63, # 区域高度
"width": 107 # 区域宽度
},
...
],
"undertake_stamp":[ # 承办人签章位置列表
{
"left": 221, # 区域左上角x坐标
"top": 941, # 区域左上角y坐标
"height": 55, # 区域高度
"width": 108 # 区域宽度
},
...
],
"register_stamp":[ # 大圆形登记注册章位置列表
{
"left": 473, # 区域左上角x坐标
"top": 808, # 区域左上角y坐标
"height": 170, # 区域高度
"width": 168 # 区域宽度
},
...
],
"other_stamp":[ # 其他印章位置列表
{
"left": 157, # 区域左上角x坐标
"top": 654, # 区域左上角y坐标
"height": 150, # 区域高度
"width": 95 # 区域宽度
},
...
],
"success": true # 识别成功与否 true/false
}
```

自定义模板识别

预备知识

- 请查看**自定义模板使用介绍**，了解如何创建和管理模板
- 请查看**请求数据格式介绍**，了解输入输出的通用数据格式。

自定义模板

购买页面：https://market.aliyun.com/products/57124001/cmapi029975.html
调用地址：在购买页面-API接口查看
模板设计页面: https://ocr.data.aliyun.com

输入格式(单模板)：

```
{
"image": "图片二进制数据的base64编码",
"configure": "{\"template_id\":\"95d551ee-8b2b-4ad0-89a9-ed13417f4a781536146904\"}"
}
```

输入格式(多模板自动匹配):

基于关键字

- 可以输入一组template_id，每组template_id有对应的条件(cond),满足条件即使用该template_id
- 基本条件包含include，exclude，分别表示包含/不包含某个关键字
- 组合条件使用and, or表示，详见下面的示例

```
{
"image": "图片二进制数据的base64编码",
"configure":{"template_list": [
{"template_id": "fc7b375c-8e98-48ef-81fb-ec3843b2fb371534902155", #星巴克的模板id
"cond": {"and": [{"or": [{"include": "星巴克"},
{"include":"starbuck"}]
},
{"exclude":"Costa"}]}
},
{"template_id": "7bc26bd7-41ad-4145-9f5a-4af21b26de9a", #costa的模板id
"cond": {"include":"costa"}
}]
}
}
```

- 如果关键字匹配失败，将执行下述的**自动匹配**

自动匹配

- 也可以直接通过如下方式进行多模板调用，系统会自动匹配出与调用图片最相近的模板

```
{
"image": "图片二进制数据的base64编码",
"configure":{"template_list": [
"fc7b375c-8e98-48ef-81fb-ec3843b2fb371534902155", #星巴克的模板id
"7bc26bd7-41ad-4145-9f5a-4af21b26de9a", #costa的模板id
```

```
]
}
}
```

输出格式(不带表格的):

```
{
"config_str":"{\"template_id\":\"95d551ee-8b2b-4ad0-89a9-ed13417f4a781536146904\"}",
"items":{"cash_id":":11535001", #key-value组合，key是用户在界面上填写的
"cash_name":":李伟",
"change":"￥0.00",
"date":":2018.03.16 16:46",
"drop":"￥-0.10",
"num":":2",
"pay":"￥65.50",
"sell_type":"外带",
"seller":"24662",
"shop_name":"TEL:0571-5697229",
"total":"￥65.60"
},
"request_id":"20180723151536_fa0edbe7408fc589b93119f7a53a1260",
"success":true
}
```

输出格式(带表格的输出)：

```
{
"config_str":"{\"template_id\":\"dd3ada93-c6ed-4427-8e0b-fbdebad614061532347130\"}",
"items":{
"item_no" : "1313131313",
"mail" : "daigou@gmail.com",
"name" : "代购小哥",
"nation" : "美国",
"phone" : "987654321",
"table0" : { #table0是用户在界面上填写的表格名称
#labels是用户在界面上填写的， 表格每一列的名称, values是每一行的值
#一行的数组和labels是一一对应的关系
  "labels":["brand", "model", "name", "quantity", "standard", "total"]
"values":[["Iphone","8p","手机","3","128G","15000rmb"],
["戴森","V8","吸尘器","1","Absolute","3500元"],
["","","","","","",""],
["","","","","","",""],
["","","","","","",""]
]
}
}
}
```

正常返回：http_status = 200
错误代码：图片输入错误或者算法错误, http_status = 400
没有设置template_json, http_status = 452

没有设置锚点, http_status = 453
没有设置识别内容, http_status = 454
没有对应的template_id, http_status = 455

表格识别

表格识别

调用地址：https://form.market.alicloudapi.com/api/predict/ocr_table_parse
云市场文档页面：https://market.aliyun.com/products/57124001/cmapi024968.html

适用场景：线条为黑色的，横线和竖线都齐全的表格识别，如：财务报表（请用finance模型）;房产证等；

输入格式

```
输入：
{
"image": "图片二进制数据的base64编码",
"configure": "{\"format\":\"html\", \"finance\":false, \"dir_assure\":false, "\line_less\":false}"
}

参数说明：
1. format 输出格式：html/json/xlsx;
2. finance 是否使用财务报表模型: true/false;
3. dir_assure 图片方向是否确定是正向的: true(确定)/false(不确定)
4. line_less:是否无线条: true(无线条,或者只有横线没有竖线)/false(有线条)
```

输出格式

```
支持三种格式输出：html/xlsx(excel)/json
#html 格式输出
{
"success":true,
"tables":"<html>\n<meta http-equiv=\"Content-Type\" content=\"text/html;charset=UTF-8\">\n<style
type=\"text/css\">\n table tr td { border: 1px solid blue }\n table { border: 1px solid blue }\n span.note { font-size:
```

9px; color: red }\n</style>\n<table \"id\"=0>\n<tr><td colspan=1 rowspan=1>项 目 </td><td colspan=1 rowspan=1>期末余额 </td><td colspan=1 rowspan=1>年初余额 </td></tr><tr><td colspan=1 rowspan=1>合计 </td><td colspan=1 rowspan=1>5,423,591,988.10 </td><td colspan=1 rowspan=1>4,281,407,583.62 </td></tr>...</table></html>\n"
}

#xlsx 格式：
{"success":true,
"tables":"UEsDBBQAAAAIAAAAIQAR0e9YNAoAAIpUAAAYAAAA..." #base64 encoded excel file(base64编码的excel文件)
}
直接拷贝tables后面的字符串到文件保存，需要将里面的\n替换掉，在linux环境下，可以执行如下操作：
sed -i -e 's/\\n/\n/g' tmp_base64
base64 -d tmp_base64 > 9_100.xlsx

#json 格式输出
{
"success":true,
"tables":[
[#table 0
[# table0 row 0
{ # table 0 row 0 col 0
"sx":0, #start from column(单元格的起始列id)
"sy":0, #start row(单元格的起始行id)
"ex":1, #one past end column index(单元格所占的列数(colspan)为ex - sx)
"ey":1, #one past end row index(单元格所占的行数(rowspan)为ey - sy)
"height":96, #cell height,图片上单元格的高度
"width":573 #cell width，图片上单元格的宽度
"text":[
"项", #text block 0 (第一个文字块)
"目" #text block 1 (第二个文字块)
],
},
...
],
[#table 0 row 1
{
"ex":1,
"ey":2,
"height":94,
"sx":0,
"sy":1,
"text":[
"合计"
],
"width":572
},
...
],
... #more rows
]
]
}

公章识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-公章识别,在API接口中找到调用地址

输入格式

```
{
"image": "对图片内容进行Base64编码"
}
```

输出格式

```
{
"result" : [
{ # 印章#1
"roi": { # 文字区域
"left": 325, # 文字区域左上角x坐标
"top": 119, # 文字区域左上角y坐标
"width": 122, # 文字区域宽度
"height": 161, # 文字区域高度
},
"text": { # 印章文字
"context": "北京开单科技有限公司", # 印章文字内容
"prob": 0.9441, # 印章文字概率
},
"general_text": [ # 其他印章区域的文字
{ # 其他文字#1
"content": "2018年07月11日", # 其他文字内容
"prob": 0.6544, # 其他文字概率
},
{ # 其他文字#2
"content": "电话:010-86468909", # 其他文字内容
"prob": 0.9612, # 其他文字概率
},
... # 其他文字#N
],
},
```

```
{ # 印章#2
"roi": { # 文字区域
"left": 140, # 文字区域左上角x坐标
"top": 139, # 文字区域左上角y坐标
"width": 118, # 文字区域宽度
"height": 122, # 文字区域高度
},
"text": { # 印章文字
"context": "北京开单科技有限公司", # 印章文字内容
"prob": 0.8877, # 印章文字概率
},
}
... # 印章#N
],
"success": true, # 识别成功与否 true/false
}
```

出租车机打发票识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-出租车机打发票识别,在API接口中找到调用地址

输入格式

```
{
"image": "base64_image_string"
}
```

输出格式

```
{
"recipts": [ #发票列表
{ # 发票1
"items": [ #每张发票的字段列表
{
```

"roi": { #对应opencv RotatedRect
"angle": -2.5791473388671875,
"center": {
"x": 332.9342041015625,
"y": 117.53900146484375
},
"size": {
"h": 23.030702590942383,
"w": 180.89764404296875
}
},
"txt": "142011671003"
},
{
"roi": {
"angle": -90,
"center": {
"x": 361,
"y": 289.5
},
"size": {
"h": 106,
"w": 17
}
},
"txt": "A-X9F99"
}
],
"roi": { #发票1 位置, 对应opencv的Rect(x, y, w, h)
"h": 763, #height
"w": 379, #width
"x": 96, #upper left corner point x
"y": 0 #upper left corner point y
},
"rotate_type": 0 #0，不需要旋转;1.顺时针转90;2.顺时针转180;3.顺时针转270
}
{ #发票2
"items": [
{ ...
},
{ ...
}
],
"roi": {
...
},
"rotate_type": 0
}
],
"success": true
}

火车票识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-火车票识别,在API接口中找到调用地址

输入格式

```
{
"image": "图片文件内容的base64编码"
}
```

输出格式

```
{
"date": "2013年10月07日10:43",
"destination": "潍坊",
"level": "新空调硬座",
"number": "K970",
"origin": "高密",
"place": "16车无座",
"price": 14.5,
"request_id": "20170720134032_416f8b6b6a13b69647e4dc9fdc696ecd",
"success": true
}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

车牌识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了车牌识别服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-车牌识别,在API接口中找到调用地址

输入格式

```
{
"image": "base64_image_string"，
"configure": "{\"multi_crop\":false}" #optional, 当设成true时,会做多crop预测，只有当多crop返回的结果一致，并且置信度>0.9时，才返回结果
}
```

输出格式

```
{
"config_str":"{\\\"multi_crop\\\":true}",
"plates":[
{
"detail":"冀AA617A,0.99753#冀AA617A,0.997782#冀AA617A,0.999783#冀AA617A,0.999999",
"prob":0.99752956628799438,
"roi":{"h":35,"w":90,"x":17,"y":21},
"txt":"冀AA617A"
}
],
"success":true
}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

vin码识别

本页面主要介绍服务对应的接口和返回结果中的关键字段的含义，请在阅读本页面之前，了解请求数据格式介绍，了解输入输出的通用数据格式。 此外，在本页最后，附上了服务调用的程序示例，以供参考。

请求接口

- 云市场接口
  - 请求方法: POST
  - 请求url: 在云市场搜索印刷文字识别-vin码识别,在API接口中找到调用地址

输入格式

```
{
"image": "图片文件内容的base64编码"
}
```

输出格式

```
{
"success" : true,
"vin" : "LSGPB54R4DD331665",
"request_id" : 20171031122455

}
```

程序示例

调用方法分为使用APPCODE调用， 使用app_key, app_secret调用。 在git中我们针对多种语言，均提供了两种调用方式的示例代码，详情请见：https://github.com/ALIBABAOCR/OCR_EXAMPLE

常见问题

输入图片大小有限制么？

由于网络传输延迟较大，在保持清晰度的情况下尽量降低图片大小，尽量保持图片在1.5M以内。如果接口返回以下错误，则说明输入图像太大了

```
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN"><html><head><title>413 Request Entity Too
```

Large</title></head><body bgcolor="white"><h1>413 Request Entity Too Large</h1><P>The requested resource does not allow request data with the requested method or the amount of data provided in the request exceeds the capacity limit. Sorry for the inconvenience.
Please report this message and include the following information to us.
Thank you very much!</p><hr/>Powered by Tengine</body></html>

输入图片清晰度

图像清晰度需要至少保持人眼可辨认，同时尽可能避免反光等干扰。

并发限制 单用户单接口的并发限制在5 QPS，如果您需要更高QPS，请通过工单或者钉钉群与我们联系

错误码

api请求body中有inputs字段的表示旧格式，否则为新格式。如果您检查对应的错误码说明后，仍然无法解决问题，请在钉钉搜索技术支持群11700462进入答疑。

错误码	错误信息	说明
400	Invalid URL	URL错误
403	Forbidden	没有购买，或者购买次数用尽，或者URL错误
408	Request Timeout	超时
413	Payload Too Large	request body太大
450		后端服务队列满，请求被拒绝，重试即可
460	Invalid Input - failed to parse json	上传的body不符合json格式要求，是非法json
461	Invalid Input - json format error - missing key: image	新格式：输入Json中缺少image键 旧格式：输入Json的inputs值中缺少image键
461	Invalid Input - json format error - image format error	新格式：输入Json中的image值不是字符串 旧格式：输入Json中的image值不是包含DataValue和DataType的Json串
461	Invalid Input - json format error - key inputs missing	旧格式：输入Json中缺少inputs键
461	Invalid Input - json format error - value of inputs must	旧格式：输入Json中的inputs值不是一个array

	be a array	
461	Invalid Input - json format error - image DataValue format error	旧格式：输入Json的image键中的DataValue值不是字符串
461	Invalid Input - json format error - configure format error	旧格式：输入Json中的configure值不是包含DataValue和DataType的Json
461	Invalid Input - json format error - failed to parse config str: xxx	输入的configure不是合法的Json
461	Invalid Input - json format error - json parse runtime exception	输入的Json格式不符合要求
462	Invalid Input - image data error - download image from url error	从URL下载图像失败
462	Invalid Input - image data error - input image empty, please check your image binary data	输入的image是空字符串
462	Invalid Input - image data error - image decode failed, please check your image binary data	输入的image解码失败、base64编码的是不合法的图像格式
462	Invalid Input - image data error - base64decode error, please check your image binary data	输入的image不是合法的base64字符串
462	Invalid Input - image data error - please remove additional header: data:image/jpg:base64,	请删除base64字符串多余的头：data:image/jpg:base64
462	Invalid Input - image data error - base64decode error: incorrect padding in base64 string	输入的image不是合法的base64字符串，base64编码的数据长度需要是4的倍数，如果长度不够，需要再末尾加上=补足。
462	Invalid Input - image data error - base64decode error: base64decode error: incorrect base64 format data	输入的image不是合法的base64字符串，存在非法字符
462	Invalid Input - image data error - open gif image failed	GIF图像打开失败
462	Invalid Input - image data error - read gif image failed	GIF图像读取失败
462	Invalid Input - image data error - decode gif image failed	GIF图像解码失败

462	Invalid Input - image data error - gif data error, no color map got	GIF获取颜色映射表错误
462	Invalid Input - image data error - gif image empty	GIF图像是为空
463	Invalid Input - wrong category	输入图像不是对应服务的图像，如行驶证服务请求的不是行驶证
464	Invalid Result - algorithm run failed	OCR识别失败
464	Invalid Result - algorithm runtime exception	OCR识别异常
464	Invalid Result - xxx	OCR识别失败
469	Invalid Service - parse result error	内部异常
469	Invalid Service - error code missing	内部异常
502	Bad Gateway	识别程序超时并断开连接
503	Service Unavailable	API网关等待超时断开连接

如果您遇到的错误码未在上述表格中找到，可在以下链接中进一步查看
https://help.aliyun.com/document_detail/43906.html

自定义模板使用介绍

自定义模板OCR使用手册

基本概念

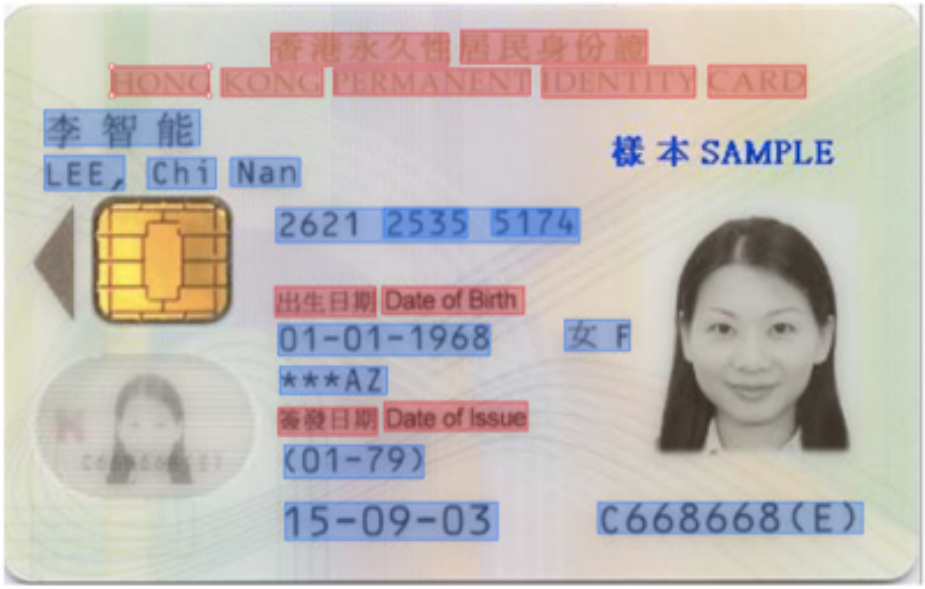
1. OCR：对文本资料的图像文件进行分析识别处理，获取文字及版面信息的过程。我们目前已有一系列定制化的阿里云OCR产品列表，详情见https://ai.aliyun.com/ocr，可以解决身份证信息提取、银行卡信息提取等一系列通用问题，但是各类自定义结构化提取的需求逐一完全满足。
2. 自定义模板OCR识别：https://ai.aliyun.com/ocr/template，支持用户通过简单的标注创建专属自己的模板，生成识别规则。模板创建后，用户可通过API接口批量识别同类图片内容信息，获得定义好的输出结果，满足用户的个性化OCR需求。

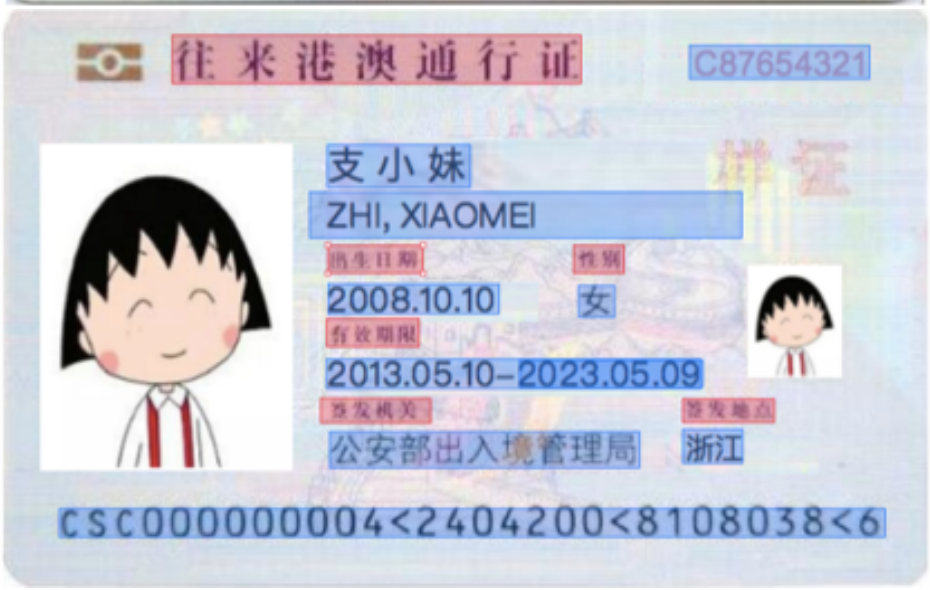
3. 模板：每个经过配置锚点和识别内容框选后的图片都可以作为模板，用来指导算法做到同一类图片的结构化信息提取，不同模板之间通过加密的id作为请求参数来区分。
4. 锚点：在需要识别的所有图片中都存在的固定文字（支持中英文，不支持换行）。作用是根据相对位置用于定位要识别的内容在图片上的位置。
5. 模板保存：暂存修改，此时可以进行模板试用，但是修改还未在线上生效
6. 模板发布：推送至EAS服务，只有发布后的修改才可以通过API请求到

适用范围

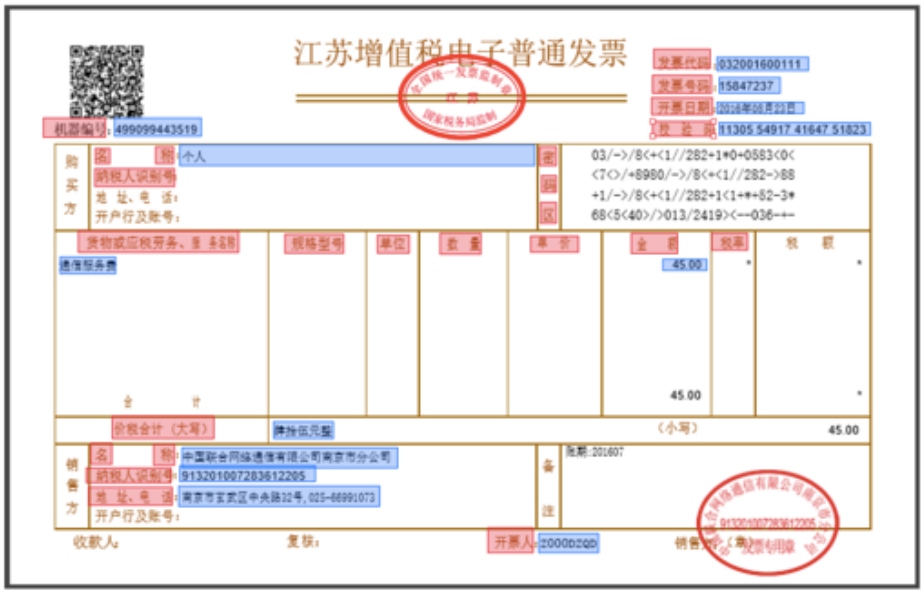
自定义模板可以用来识别并得到结构化输出的图片内容包括但不限于

证件类



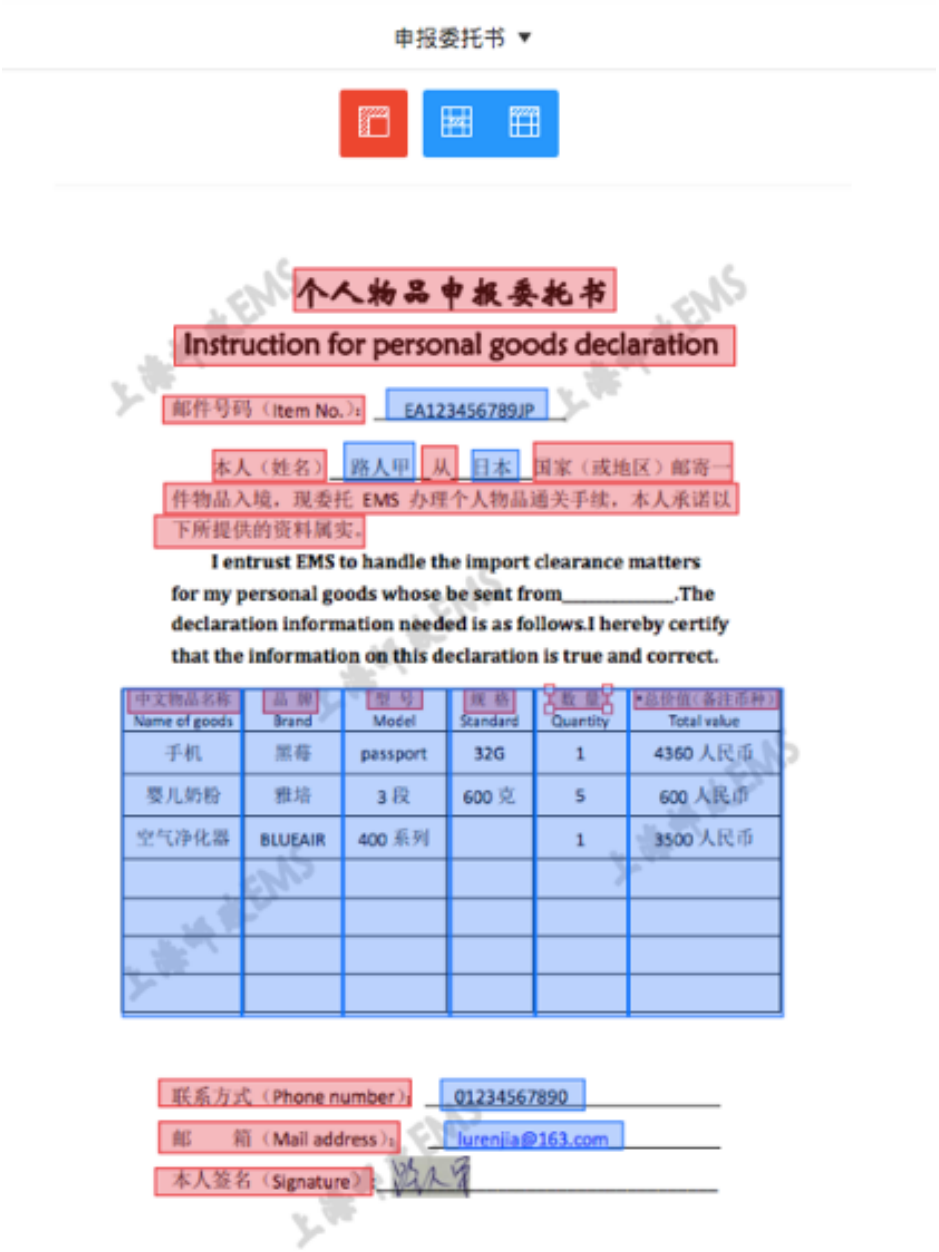


票据类


江苏增值税电子普通发票
机器编号：499099443519
发票代码：032001600111
发票号码：15847237
开票日期：
校验码：11305 54917 41647 51823
购买方
名称：个人
纳税人识别号：
地址、电话：
开户行及账号：
密码区
03/->/8<+<1//282+1*0+0583<0<
<7<>/+8980/->/8<+<1//282->88
+1/->/8<+<1//282+1<1+*+52-3*
68<5<40>/>013/2419>/<--036-+-
货物或应税劳务、服务名称
规格型号
单位
数量
单价
金额
税率
税额
通信服务费
45.00
*
*
45.00
价税合计（大写）
（小写）
45.00
销售方
名称：中国联合网络通信有限公司南京市分公司
纳税人识别号：91320100728361220S
地址、电话：南京市玄武区中央路10号，025-66991073
开户行及账号：
备注
账期:201607
收款人：
复核：
开票人：2000DZQD
销售方（章）：


(IPOS)
BreadTalk
面包新语
深圳万象城店
2 pos02 205 张舒
打印次数: 1 205 张舒
账单90134488
2018-1-14 19:54:51
1 红豆芋包 *外 9.00
1 切达吉士 *外 12.00
1 脆糖皇冠 *外 9.00
1 千层雪顶 *外 13.00
物品售卖总数: 4
小计 43.00
总金额 43.00
付款 43.00
商场礼券 43.00
------205 结束 2018-1-14 19:55:45------
** 多谢惠顾 **
地址:深圳市罗湖区宝安南路1881号B30
电话:0755-82668664

文书类



【重点】如何创建模板

上面的所有自定义模板案例图片中，红色部分都是图片中不会变化的文字，我们称之为「锚点」，蓝色部分就是对您业务来说比较重要的需要结构化提取的「识别内容」，接下来我们以常见的大学四六级证书介绍一下如何创建一个自定义模板。

1. 新建模板

在模板管理页面上点击「新建模板」

2. 上传图片

上传一张平整、无遮挡、文字清晰且方向尽量水平的图片作为模板图片

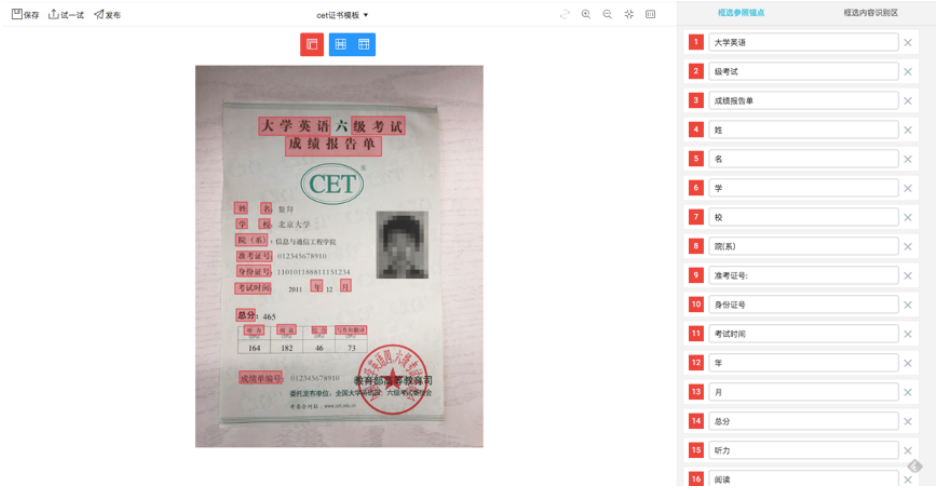
3. 调整图片

设置完模板名称之后，我们进入模板编辑页面，在编辑页面最上方是我们的工具栏，分别对应的功能

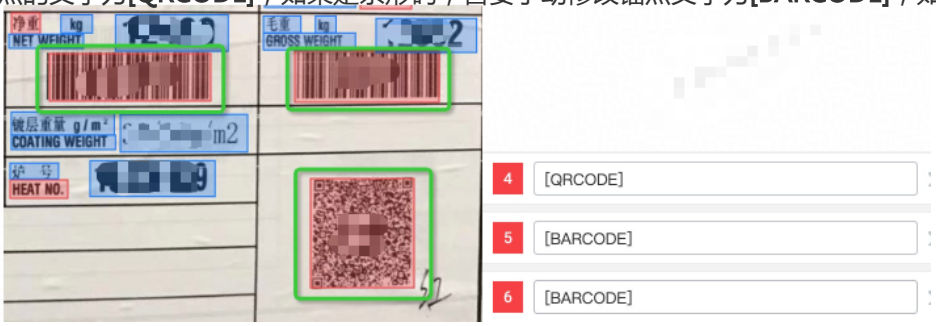
如果发现图片拍的不太正，可以点击右上角的「旋转」按钮，进行角度调整，（调整的标准是使得图片上的文字均与页面水平线平行），然后可以点击确认保存，旋转后的图片就会被保存下来，后续的标注就都是在旋转过的图片上进行了（不要在已经有了锚点或内容圈选之后再进行旋转哦，会导致已标注的内容位置不准确）。

4. 标注锚点区域

我们在主功能操作栏中选择「框选锚点」，然后圈选出图片上尽可能多的固定不变的文字作为锚点，用来定位，这一步是识别成功的关键。每次圈选，屏幕右侧都会自动识别出圈选的文字内容，自动识别的锚点内容如果有错误，可能是当前图片相关区域不太清晰，您可以手动修改正确。



- 【近期更新】二维码/条形码做锚点支持将图片中的二维码、条形码作为锚点。如果是二维码需要手动修改锚点的文字为**[QRCODE]**；如果是条形码，首要手动修改锚点文字为**[BARCODE]**，如下示例



5. 标注识别内容区域

接下来，我们圈选图片上需要识别的内容

a) **识别普通文本**

依次圈选之后，在右侧列表中将系统给的默认字段名改为有意义的名称。注意相同名称的内容，在实际识别的时候会被合并到同一个字段(合并后的先后顺序按照右侧排列的顺序，顺序可以拖拽调整哦)，常用于文字内容有多行的情况，后面的小贴士中会有介绍。

b) **识别表格内容**

本例中的表格比较简单，只有一行内容，因此用普通文本识别也是可以逐一圈选并得到结果的，当然我们也可

以用表格内容识别来进行处理，插入一个4列表格

所有的标注都完成后，点击左上角保存，一个模板就制作完成了！

6. 测试效果

接下来我们再找一张四六级证书，点击左上角的「试一试」检测一下模板配置，结构化输出的结果如图，我们发现在这张图片中对应的字段已经全部识别正确

1. 模板配置可以多次修改并试用，调整至最优后，点击左上角的「发布」，成功后就可以通过模板的唯一标识id去进行接口调用了，具体可以查看API调试方式

【*】小贴士

1. 锚点标注

a. 锚点必须是不变的文字，且中间没有大片空白

b. 锚点需要尽可能靠近需要提取的内容项目，如位于同一行或者相邻行
c. 锚点越多越好，建议在4个以上
d. 请尽量用最小面积的四边形圈选锚点文字
e. 表格中的表头通常也是不变的文字内容，建议也都设置为锚点，如上述例子中的六级成绩单中的分科目成绩表中设置

2. 识别区标注

a.**换行字段配置**

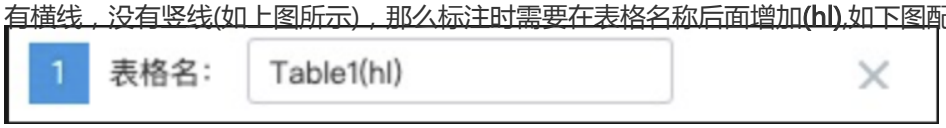
- 如图红色粗实线框所示，图中的收件地址可能会换行
- 需要在send_addr后面**增加(cr)**（图中有两个send_addr,其中一个加就可以），系统就会知道当前字段会有换行的内容，如图

- 如果模板中只有一行，那么：
    - 如果在全部需要识别的图片中存在换行的可能，那么也**需要**加上**(cr)**；
    - 如果不存在换行的可能，**不要**添加**(cr)**。

3. 表格内容标注

- 请将标注的框线和表格本身的边框尽量重合
- 对于复杂的表格类型，比如包含合并单元格，返回结果中的表格内容暂时无法支持，可以将表格识别可以和普通识别配合使用
- 暂时不支持只返回部分表格行内容
- 在表格上下，最好有**贴近的锚点**
- **【新功能】我们增强了对于表格框线不全的表格识别的支持**

- 如果表格有横线，没有竖线(如上图所示)，那么标注时需要在表格名称后面增加**(hl)**,如下图配置



- 如果表格有竖线，没有横线，那么表格名称后面增加**(vl)**
- 如果表格既没有竖线也没有横线，那么表格名称后面增加**(nl)**

4. 关于试一试

试用后如果发现准确率不符合要求，可以重新尝试调整模板，建议的调整方案包括：
(1).增加锚点数量
(2).扩大识别内容圈选区域
(3).锚点中空白区域大可以拆分成多个锚点
(4).降低锚点标注区域重合

总结

只要您需要大量提取的图片文字内容有固定的排版样式，都可以来试一试自定义模板，用DIY的方式解决业务需求！