工业识别V1.0.0-算法接口

请求URL

demo:

POST /devicesrecognize

V1.0.0:

POST /iot/device/recognize/image

参数列表

返回参数名称	类型	是否必须	说明
tags	string	否	标签信息
imgName	string	若传入url参数，此参数可不传	图片名，以后端传入的为主
image	string	和url二选一	图像数据，base64编码，要求base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，支持jpg/png/bmp格式 。**注意：图片需要base64编码、去掉编码头后再进行urlencode。**
url	string	和image二选一	图片完整URL，URL长度不超过1024字节，URL对应的图片base64编码后大小不超过4M，最短边至少50px，最长边最大4096px，支持jpg/png/bmp格式，当image字段存在时url字段失效。

请求响应

返回参数名称	类型	是否必须	说明
serviceVersion	string	是	识别服务版本号
timestamp	string	是	调用服务时的时间戳，使用python的int(time.time()产生

imgName			string	是	图片名，以后端传入的为主
status			string	是	状态码
message			string	是	状态码信息说明
result			dict	是	返回结果相关
	imgScore		float32	是	图像质量评估得分
	recognizeInfo		list	是	识别结果详细内容
		deviceCrop	base64	是	检测到的设备/零件目标截取图
		box	list	是	检测到的设备/零件目标相对于原图的坐标
			left	int	是	目标框左上角的 x 坐标
			top	int	是	目标框左上角的 y 坐标
			right	int	是	目标框右下角的 x 坐标
			bottom	int	是	目标框右下角的 y 坐标
		deviceList	list	是	识别设备/零件目标ID的top2结果或者阈值大于
			deviceType	int	是	设备目标类型，设备整体deviceType=0(如deviceId=0000A)；设备零件deviceType=1(如deviceId=0000A_0、0000A_1，0000A_3.....等等)
			deviceId	string	是	识别到的设备/零件目标的所属ID
			score	float32	是	识别到的设备/零件目标的置信度得分

请求示例

```
1 # -*- coding:utf-8 -*-
2
```

```
import requests
import json
import argparse

detail_template = {
    "tags": "1.0.0",
    "imgName": "2024-05-24_14:38:50.jpg",
    "image":
"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQE......",
    "url": "https://iot-bucket-2024.obs.cn-south-
1.myhuaweicloud.com:443/iot/436f1fd9-ea3d-43f8-be8f-af08961a3021.jpeg",
}


class RecognizeTest(object):
    def __init__(self, base_url):
        self.base_url = base_url

    def device_recognize(self, data):
        url = self.base_url + '/devicesrecognize'
        header = {"Content-Type": "application/json"}
        params = data
        response = requests.post(data=json.dumps(params), url=url,
headers=header)
        try:
            print("final result: ", response.json())
        except Exception as e:
            print("Error", e)
        print(response.json())
        return response.json()


if __name__ == '__main__':
    parser = argparse.ArgumentParser()
    url = "http://192.168.101.120:8080"
    parser.add_argument('--url', default=url, required=False, help='')
    parser.add_argument('--detail_template', default=detail_template,
required=False, help='')
    opt = parser.parse_args()
    recognize_test = RecognizeTest(opt.url)
    recognize_test.device_recognize(detail_template)
```

```
curl --location '192.168.101.120:8080/iot/d
--header 'Content-Type: application/json' \
--data '{
    "tags": "1.0.0",
    "imgName": "XXXXXXXXXXXX.jpg",
    "url":
"https://iot-bucket-2024.obs.cn-south-1.m
65b61f1c9.jpeg",
    "image":""
}'
```

cURL.txt

```python
# -*- coding:utf-8 -*-
from flask import Flask, request, Response
import json

app = Flask(__name__)




def predict_image(img_content):
    """
    识别主函数
    :param img_content:
    :return:
    """
    data = {
        "serviceVersion": "v1.0.0",
        "timestamp": "1242536363",
        "imgName": "2024-05-24_14:38:50.jpg",
        "status": "A0200",
        "message": "runSuccess",
        "result": {
            "imgScore": 0.6,
```

```
            "recognizeInfo": [{'deviceCrop':
"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQE......",
                               'box': [{"left": 172,
                                        "top": 402,
                                        "right": 588,
                                        "bottom": 928}],
                               "deviceList": [{'deviceId': '10001', 'score':
0.87, "deviceType": 0},
                                              {'deviceId': '10002', 'score':
0.76, "deviceType": 0}]},
                              {'deviceCrop':
"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAEBAQEBAQEBAQEBAQEBAQEBAQE......",
                               'box': {"left": 523,
                                        "top": 161,
                                        "right": 994,
                                        "bottom": 919},
                               "deviceList": [{'deviceId': '10003_0', 'score':
0.91, "deviceType": 1},
                                              {'deviceId': '10004_0',
'score': 0.88, "deviceType": 1}]}
                              ]
        }
    }
    return data


@app.route('/devicesrecognize', methods=['POST'])
def devices_recognize():
    """
    识别入口
    :return:
    """
    request_data = request.get_json()
    image_input = request_data["image"]
    predict_info = predict_image(image_input)
    print(predict_info)
    return Response(json.dumps(predict_info))


if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080, processes=True, debug=False)
```

{
"serviceVersion": "1.0.0",
"timestamp": 1718281382,
"imgName": "6ae7bcaf-c928-46d1-928c-5bd65b61f1c9.jpeg",
"status": "A0200",
"message": "runSuccess",
"result": {
"recognizeInfo": [
{
"deviceCrop":
"/9j/4AAQSkZJRgABAQAAAQABAAD/2wBDAAIBAQEBAQIBAQECAgICAgQDAgICAgUEBAMEBg
UGBgYFBgYGBwkIBgcJBwYGCAsICQoKCgoKBggLDAsKDAkKCgr/2wBDAQICAgICAgUDAwUKBw
YHCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgoKCgr/wAARC
ANuBDgDASIAAhEBAxEB/8QAHwAAAQUBAQEBAQEAAAAAAAAAAAECAwQFBgcICQoL/8QAtR
AAAgEDAwIEAwUFBAQAAAF9AQIDAAQRBRIhMUEGE1FhByJxFDKBkaEII0KxwRVS0fAkM2Jygg
kKFhcYGRolJicoKSo0NTY3ODk6Q0RFRkdISUpTVFVWV1hZWmNkZWZnaGlqc3R1dnd4eXqDhIW
Gh4iJipKTlJWWl5iZmqKjpKWmp6ipqrKztLW2t7i5usLDxMXGx8jJytLT1NXW19jZ2uHi4+Tl5ufo6erx
8vP09fb3+Pn6/8QAHwEAAwEBAQEBAQEBAQAAAAAAAAECAwQFBgcICQoL/8QAtREAAgECBA
QDBAcFBAQAAQJ3AAECAxEEBSExBhJBUQdhcRMiMoEIFEKRobHBCSMzUvAVYnLRChYkNOEl8
RcYGRomJygpKjU2Nzg5OkNERUZHSElKU1RVV1dYWVpjZGVmZ2hpanN0dXZ3eHl6goOEhYaHiIm
KkpOUlZaXmJmaoqOkpaanqKmqsrO0tba3uLm6wsPExcbHyMnK0tPU1dbX2Nna4uPk5ebn6Onq8vP09
fb3+Pn6/9oADAMBAAIRAxEAPwD6w/tBv7woGoOOSRXP/wDCS2p6Z/Kmt4itWOAjCv5ls10OVySOi/t
FmxuOPxp0eqsnB5rmTrCv/qzj6mmjXIu+fzpNNE86OsXVmzywx9aedYVgDHwT1Ga5Ma5AcqCfzpBrkY
OQGH403dIjnO1XVnPCuPzofUJlALcj1rkhrijkKR+NPj18lhvZsH/a6VQ1NI66PU2GfmxUseqOmfnHPqa
5RNbjZcl2/M1Kmtwhh+8bHuxp3L50db/asvStTvGsuBg

return.txt