

# 车牌 OCR SDK 开发说明

当前版本： V1.1

当前状态： 已发布

## 版本记录

版本	状态	日期	记录
V1.0	修订中	2021-09-23	第一个版本
V1.1		2021-10-21	修改格式
V1.2		2022-04-19	接口统一

## 目录

车牌 OCR SDK 开发说明.....	1
1、 开发流程说明.....	2
2、 SDK 接口说明.....	2
2.1、 获取 SDK 实例.....	2
2.2、 服务初始化.....	2
2.3、 获取服务状态.....	2
2.4、 服务关闭接口.....	3
2.5、 车牌识别接口-byte[].....	3
2.6、 车牌识别接-Bitmap.....	3
2.7、 Sdk 版本信息.....	3
2.8、 获取 SDK 名称.....	4
2.9、 Debug 开关.....	4
3、 返回结果说明.....	4
3.1、 获取文字坐标点.....	4
3.2、 获取文字信息.....	4
3.3、 置信度.....	5
4、 DEMO Code.....	5
5、 附录.....	6
4.1、 错误代码说明.....	6

## 1、 开发流程说明

### ➤ 开发流程



## 2、 SDK 接口说明

### 2.1、 获取 SDK 实例

```
Predictor predictor = Predictor.getInstance();
```

```
Predictor getInstance()
```

	参数	类型	说明
输入参数			
输出参数	predictor	Predictor	null 失败，其他成功

### 2.2、 服务初始化

调用该接口初始化 OCR 识别服务，完成激活和初始化。

```
predictor.initModel(appCtx);
```

```
boolean initModel(Context appCtx))
```

	参数	类型	说明
输入参数	appCtx	Context	
输出参数	Ret	boolean	true 成功， false 失败

### 2.3、 获取服务状态

```
int getModelState()
```

	参数	类型	说明
输入参数			
输出参数	ret	int	0 初始化成功，-1 授权失败，-2 模型加载失败，-3 字典加载失败

## 2.4、服务关闭接口

调用该接口释放 OCR 识别模型资源。

```
predictor.releaseModel();  
void releaseModel()
```

	参数	类型	说明
输入参数			
输出参数			

## 2.5、车牌识别接口-byte[]

初始化成功后开始识别功能，接口定义如下：

```
ArrayList<OcrResult> runOcrPredictor(byte[] nv21, int w, int h, int rotate)
```

	参数	类型	说明
输入参数	nv21	byte[]	Camera 预览回调上来的图像数据 nv21
	w	int	预览图像宽度
	h	int	预览图像高度
	rotate	int	预览图像需要旋转的角度
输出参数	ocrResults	ArrayList<OcrResult>	识别结果

## 2.6、车牌识别接-Bitmap

```
ArrayList<OcrResult> runOcrPredictor(Bitmap image, int rotate)
```

	参数	类型	说明
输入参数	image	Bitmap	识别的图像
	rotate	int	预览图像需要旋转的角度
输出参数	ocrResults	ArrayList<OcrResult>	识别结果

## 2.7、Sdk 版本信息

调用获取 OCR SDK 版本信息

```
String getVersionInfo()
```

	参数	类型	说明
输入参数			
输出参数	Version	String	

## 2.8、 获取 SDK 名称

调用获取 OCR SDK 名称

**String getSDKName()**

	参数	类型	说明
输入参数			
输出参数	SDKname	String	

## 2.9、 Debug 开关

sdk 调试开启、关闭

**void enableDebug(boolean enable)**

	参数	类型	说明
输入参数			
输出参数	enable	boolean	true 开启，false 关闭

## 3、 返回结果说明

OcrResult 接口说明

### 3.1、 获取文字坐标点

**List<Point> getPoints()**

	参数	类型	说明
输入参数			
输出参数	points	List<Point>	返回文字矩形框坐标点

### 3.2、 获取文字信息

**String getLabel()**

	参数	类型	说明
输入参数			
输出参数	Label	String	返回识别的文字

3.3、置信度

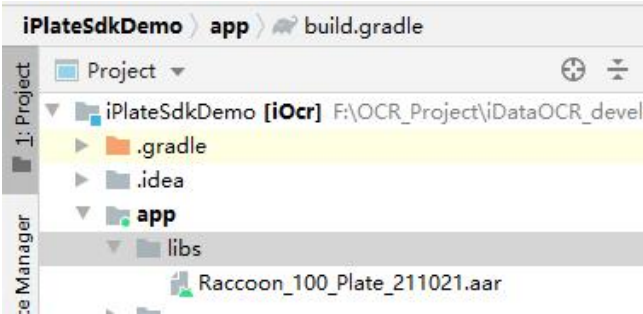
float getConfidence()

	参数	类型	说明
输入参数			
输出参数	confidence	float	

4、 DEMO Code

➤ 注意：请先将厂商提供的 lib 库加项目依赖路径，如下图：

Raccoon\_100\_Plate\_211021.aar



Build.gradle 中加增加如下配置：

```
dependencies {  
    implementation fileTree(include: ['*.jar','*.aar'], dir: 'libs')  
}
```

➤ AndroidManifest.xml 配置权限

```
<uses-permission android:name="android.permission.CAMERA"/>  
<uses-permission android:name="android.permission.INTERNET"/>  
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>  
<uses-permission android:name="android.permission.ACCESS_NETWORK_STATE" />  
<uses-permission android:name="android.permission.READ_PHONE_STATE"/>  
<uses-permission android:name="android.permission.READ_PRIVILEGED_PHONE_STATE" />
```

需要在 AndroidManifest.xml 中增加如下配置方可访问网络；

```
android:networkSecurityConfig="@xml/network_security_config"
```

```
AndroidManifest.xml | network_security_config.xml
12 <uses-permission android:name="android.permission.READ_PHONE_STATE"/>
13 <uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
14 <uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
15 <uses-permission android:name="android.permission.ACCESS_WIFI_STATE"/>
16 <uses-permission android:name="android.permission.INTERNET"/>
17
18 <application
19     android:allowBackup="true"
20     android:icon="@mipmap/ic_launcher"
21     android:label="@string/app_name"
22     android:roundIcon="@mipmap/ic_launcher_round"
23     android:supportRtl="true"
24     android:theme="@style/Theme.IOCr"
25     android:networkSecurityConfig="@xml/network_security_config"
26     android:requestLegacyExternalStorage="true">
```

network\_security\_config.xml 中内容如下:

```
network_security_config.xml
1 <?xml version="1.0" encoding="utf-8"?>
2 <network-security-config>
3     <base-config cleartextTrafficPermitted="true"/>
4 </network-security-config>
```

➤ Demo Code

```
Predictor predictor = Predictor.getInstance();
boolean ret = predictor.initModel(context);
if(ret) {
    ArrayList<OcrResult> arrayList = predictor.runOcrPredictor(mImage,0);

    drawResults(arrayList);
}
```

详见 demo app

5、 附录

4.1、 错误代码说明

Code	说明	建议
0	成功	
-1	SN 授权失败	
-2	模型加载失败	
-3	字典加载失败	