

行驶证预览识别 Android SDK 说明文档

V1.0.0.20170707

目录

1、行驶证预览识别功能介绍	2
2、SDK 开发准备	3
3、行驶证预览识别 Android 版本 SDK 开发说明	4
3.1 调用 SDK 中的相机模块进行行驶证识别;	4
3.2 直接调用 SDK 中的方法进行行驶证识别;	4
4、SDK 接口使用说明	4
4.1 调用 SDK 中的相机模块识别接口;	5
4.2 直接调用 SDK 中的方法识别接口;	7
4.3 行驶证预览识别 API 返回错误码说明	9
4.4 行驶证预览识别图片要求	10
5、维护与故障说明	10
6 调用示例代码	11
6.1 调用 SDK 相机模块的 API 识别的示例	11
6.2 调用 SDK 方法 API 识别的示例	11
7 相机预览拍照的使用文档	11
8 修订历史记录	12

1、行驶证预览识别功能介绍

通过上海合合信息全球领先的 OCR 技术，对相机预览返回视频帧数据中的行驶证图片进行自动识别，返回行驶证图片上的号牌号码、车辆类型、所有人、住址、使用性质、品牌型号、车辆识别代码、发动机号码、注册日期、发证日期等信息，可以省去用户手动录入的过程，给用户带来极大的便利。为了给用户提供行驶证识别的体验，合合信息支持提供 Android 与 iOS 系统行驶证识别 SDK。用户只需在 APP 中集成上海合合信息提供的行驶证识别 SDK，就可以给用户提本地行驶证识别功能。

行驶证预览识别的功能列表如下：

功能列表	基础版
1. 号牌号码	支持
2. 车辆类型	支持
3. 所有人	支持
4. 地址	支持
5. 使用性质	支持
6. 品牌型号	支持
7. 车辆识别代号	支持
8. 发动机号码	支持
9. 注册日期	支持
10. 发证日期	支持

2、SDK 开发准备

SDK 开发准备步骤如下：

1. 使用行驶证预览识别 SDK 前，请先将 sdk\libs\下的 vlcardscansdk.jar 和 armeabi-v7a 文件夹（文件夹下包含了本地识别库）放置到 Android 项目的 libs 文件夹下，如图 1

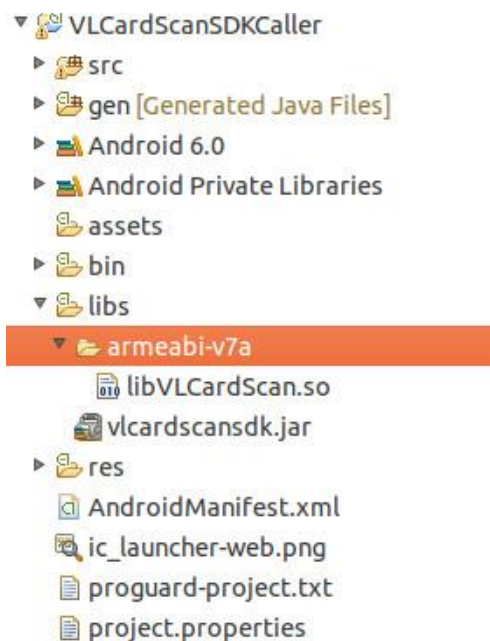


图 1

备注：

- 1) armeabi-v7a 下的 libVLCARDSCAN.so 是支持 ARM 和 ARM-V7 CPU 架构的 .so 库文件；如果只需要支持 ARM 架构 CPU，则可以直接把 armeabi-v7a 下的 libVLCARDSCAN.so 拷贝到 armeabi 文件夹下，在 libs 目录下，只需保留一个 armeabi 文件夹即可；
- 2) libVLCARDSCAN.so 文件名不可以修改，否则会出现 SDK 加载失败的问题；

2. 在 Android 项目的 AndroidManifest.xml 里声明以下权限

```
<uses-permission  
android:name="android.permission.INTERNET"/>  
<uses-permission  
android:name="android.permission.CAMERA"/>  
<uses-permission
```

```
android:name="android.permission.READ_PHONE_STATE"/>
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_STORAGE"/>
```

3. 行驶证预览识别 SDK 支持 Android 2.3 及以上版本;
4. 使用 Proguard 进行混淆时, 不能混淆 SDK 中任何类, 需要在 Proguard 文件中添加下面这一行:

```
-keep class com.intsig.** {*;} 
```

至此行驶证预览识别 SDK 开发的准备工作就已完成, 接下来介绍使用 SDK 接口的使用说明。

3、行驶证预览识别 Android 版本 SDK 开发说明

行驶证预览识别 Android 版本 SDK, 对外提供行驶证本地识别. so 库文件与 Java JNI 接口, 方便用户直接集成到 APP 开发项目中。为了方便客户集成, 行驶证预览识别 SDK 支持以下 2 种调用方式:

3.1 调用 SDK 中的相机模块进行行驶证识别;

通过 Intent 调用启动 SDK 中的相机拍摄模块, 对行驶证进行自动识别, 并把识别结果返回给第三方 APP, 整个行驶证拍摄与识别的过程都封装在 SDK 的相机模块中。对于相机拍摄模块没有自定义需求的客户, 可以直接采用这种调用方式, 更加便捷。

3.2 直接调用 SDK 中的方法进行行驶证识别;

第三方 APP 根据自己的需求开发实现相机拍摄模块, 并在相机拍摄模块中按照行驶证预览识别 SDK 对外提供的行驶证识别方法进行行驶证识别。对于需要自定义相机模块的客户, 可以采用这种调用方式, 更方便自定义调整。

4、SDK 接口使用说明

行驶证预览识别 SDK 的代码集成示例, 可以参考“VLCardScanSDKCaller”demo 项目, 包含

调用 SDK 中的相机模块识别和直接调用 SDK 方法进行识别 2 种，调用说明如下。

4.1 调用 SDK 中的相机模块识别接口；

为了方便用户直接调用，在行驶证预览识别 SDK 中封装了相机拍摄模块，可以直接开发给第三方 APP 通过 Intent 进行调用。相关说明如下：

a) 启动相机拍摄识别模块：

```
//通过Intent调用SDK中的相机拍摄模块ISCardScanActivity进行识别
Intent intent = new Intent(this, ISCardScanActivity.class);
intent.putExtra(ISCardScanActivity.EXTRA_KEY_IMAGE_FOLDER,
"/sdcard/vlcardscan/");
intent.putExtra(ISCardScanActivity.EXTRA_KEY_COLOR_MATCH,
0xffff0000);
intent.putExtra(ISCardScanActivity.EXTRA_KEY_COLOR_NORMAL,
0xff00ff00);
intent.putExtra(ISCardScanActivity.EXTRA_KEY_APP_KEY, APP_KEY);
intent.putExtra(ISCardScanActivity.EXTRA_KEY_TIPS, "请将行驶证放在框内识别");
startActivityForResult(intent, REQ_CODE_CAPTURE);
```

Intent参数说明如下：

参数	说明
ISCardScanActivity.class	SDK中的相机拍摄模块 ISCardScanActivity类
ISCardScanActivity.EXTRA_KEY_IMAGE_FOLDER	指定要临时保存的行驶证图片路径
ISCardScanActivity.EXTRA_KEY_COLOR_MATCH	指定SDK相机模块 ISCardScanActivity四边 框角线条,检测到行驶证图 片后的颜色,可以自定义;
ISCardScanActivity.EXTRA_KEY_COLOR_NORMAL	指定SDK相机模块 ISCardScanActivity四边 框角线条,检测到行驶证图 片后的颜色,可以自定义;
ISCardScanActivity.EXTRA_KEY_APP_KEY	合合信息授权提供的 APP_KEY;
ISCardScanActivity.EXTRA_KEY_TIPS	指定SDK相机模块 ISCardScanActivity界面 提示字符串,可以自定义;
REQ_CODE_CAPTURE	自定义onActivityResult 中的requestCode

b) 获取行驶证识别结果：

在 `onActivityResult()` 中通过返回的 `Intent data` 对象获取行驶证识别结果，说明如下：

```
ResultData result = (ResultData)
data.getSerializableExtra(ISCardScanActivity.EXTRA_KEY_RESULT_DATA);
```

`ISCardScanActivity.EXTRA_KEY_RESULT_DATA` 是获取行驶证识别结果 `ResultData` 对象的 `Intent KEY` 值。

`ResultData` 行驶证识别结果返回值说明如下：

ResultData 识别结果	行驶证识别结果说明
<code>getPlateNo()</code>	获取行驶证号牌号码；
<code>getType()</code>	获取行驶证车辆类型；
<code>getOwner()</code>	获取行驶证所有人；
<code>getAddress()</code>	获取行驶证住址；
<code>getUseCharacter()</code>	获取行驶证试用性质；
<code>getModel()</code>	获取行驶证品牌型号；
<code>getVin()</code>	获取行驶证车辆识别代号；
<code>getEngineNo()</code>	获取行驶证发动机号码；
<code>getRegisterDate()</code>	获取行驶证注册日期；
<code>getIssueDate()</code>	获取行驶证发证日期；
<code>getPlateNo()</code>	获取行驶证号牌号码；
<code>getOriImagePath()</code>	获取行驶证图片路径；
<code>getAngel()</code>	获取行驶证图片角度识别结果，例如 0 度或 180 度；
<code>getPlateNoPos()</code>	获取行驶证号牌号码条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getTypePos()</code>	获取行驶证车辆类型条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getOwnerPos()</code>	获取行驶证所有人条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getAddressPos()</code>	获取行驶证住址条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getUseCharacterPos()</code>	获取行驶证试用性质条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getModelPos()</code>	获取行驶证品牌型号条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getVinPos()</code>	获取行驶证车辆识别代号条目坐标，从左上角开始，按顺时针方向的 4 个点坐标 <code>[x1, y1, x2, y2, x3, y3, x4, y4]</code> ；
<code>getEngineNoPos()</code>	获取行驶证发动机号码条目坐标，从左上角开始，按顺

	时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getRegisterDatePos()</code>	获取行驶证注册日期条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getIssueDatePos()</code>	获取行驶证发证日期条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getPlateNoPos()</code>	获取行驶证号牌号码条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getTrimImagePath()</code>	行驶证的切边后的图片地址

c) 在 Android 项目的 `AndroidManifest.xml` 里声明以下 Activity:

```
<activity
    android:name="com.intsig.vlcardscansdk.ISCardScanActivity">
</activity>
```

4.2 直接调用 SDK 中的方法识别接口;

SDK 接口使用说明如下:

1. `int VCardScanSDK.initCardScan(Context context, String appKey)`

这个函数的作用是“验证授权信息, 初始化行驶证识别引擎”, 需要在线程中调用; 因此在调用行驶证识别函数之前必须执行, 在应用初始化时调用。

参数说明如下:

初始化返回码请见【[4.3 行驶证预览识别 API 返回错误码说明](#)】

参数	说明
<code>context</code>	第三方应用程序的 application context;
<code>appKey</code>	根据第三方 APP 包名和签名信息生成的 APP_KEY;

2. `int[] VCardScanSDK.detectBorder(byte[] preview, int width, int height, int left, int top, int right, int bottom)`

这是检测相机预览视频帧图片边缘函数。

参数说明如下:

参数	说明
<code>preview</code>	相机预览帧图片数据
<code>width</code>	相机预览帧图片宽度
<code>height</code>	相机预览帧图片高度

left	预设预览区域的四个角坐标左上角 x 坐标
top	预设预览区域的四个角坐标左上角 y 坐标
right	预设预览区域的四个角坐标右下角 x 坐标
bottom	预设预览区域的四个角坐标右下角 y 坐标

DLCardScanSDK.detectBorder() 函数返回值说明如下：

int[] 返回值	说明
int[8] 数值	SDK 返回的行驶证图片四个角的坐标，请参考“VLCardScanSDKCaller”demo 项目

3. ResultData VLCardScanSDK.recognize(byte[] preview, int width, int height, String folder)

行驶证图片识别函数。

VLCardScanSDK.recognize() 参数说明如下：

参数	说明
preview	相机预览帧图片数据；
width	相机预览帧图片宽度；
height	相机预览帧图片高度；
folder	图片保存路径；

VLCardScanSDK.recognize() 函数返回值 ResultData 说明如下：

ResultData 识别结果	行驶证识别结果说明
getPlateNo()	获取行驶证号牌号码；
getType()	获取行驶证车辆类型；
getOwner()	获取行驶证所有人；
getAddress()	获取行驶证住址；
getUseCharacter()	获取行驶证试用性质；
getModel()	获取行驶证品牌型号；
getVin()	获取行驶证车辆识别代号；
getEngineNo()	获取行驶证发动机号码；
getRegisterDate()	获取行驶证注册日期；
getIssueDate()	获取行驶证发证日期；
getPlateNo()	获取行驶证号牌号码；
getOriImagePath()	获取行驶证图片路径；
getAngel()	获取行驶证图片角度识别结果，例如 0 度或 180 度；
getPlateNoPos()	获取行驶证号牌号码条目坐标，从左上角开始，按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4]；
getTypePos()	获取行驶证车辆类型条目坐标，从左上角开始，按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4]；
getOwnerPos()	获取行驶证所有人条目坐标，从左上角开始，按顺时针

	方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getAddressPos()</code>	获取行驶证住址条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getUseCharacterPos()</code>	获取行驶证试用性质条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getModelPos()</code>	获取行驶证品牌型号条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getVinPos()</code>	获取行驶证车辆识别代号条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getEngineNoPos()</code>	获取行驶证发动机号码条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getRegisterDatePos()</code>	获取行驶证注册日期条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getIssueDatePos()</code>	获取行驶证发证日期条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];
<code>getPlateNoPos()</code>	获取行驶证号牌号码条目坐标, 从左上角开始, 按顺时针方向的 4 个点坐标[x1, y1, x2, y2, x3, y3, x4, y4];

4. VCardScanSDK.release()

释放行驶证识别引擎资源, 在应用推出时调用即可, 必须和

VCardScanSDK.initCardScan() 函数匹配调用, 而且**必须是 initCardScan() 初始化成功才能调用 release()**。

详细使用行驶证预览识别 SDK 代码请参考 VCardScanSDKCaller 项目

4.3 行驶证预览识别 API 返回错误码说明

VCardScanSDK.initCardScan() 函数返回 int 值说明如下:

int 返回值	说明
0	初始化成功
101	原因: 包名错误, 授权 APP_KEY 与绑定的 APP 包名不匹配; 解决方法: 请检查工程中的包名是否与提供给合合信息授权绑定的包名一致。
102	原因: appKey 错误, 传递的 APP_KEY 填写错误; 解决方法: 请检查工程传入的 appKey 是否与合合信息授权的 APP_KEY 一致。

103	原因：超过时间限制，授权的 APP_KEY 超出使用时间限制； 解决方法：授权到期，如需延长，请与合合信息技术支持联系。
104	原因：达到设备上限，授权的 APP_KEY 使用设备数量达到限制； 解决方法：设备超限，请与合合信息技术支持联系新增设备的授权。
201	原因：签名错误，授权的 APP_KEY 与绑定的 APP 签名不匹配； 解决方法：请检查工程中的 MD5 签名是否与提供给合合信息授权绑定的签名信息一致。
202	原因：其他错误，其他未知错误，比如初始化有问题； 解决方法：请将具体日志发给合合技术支持，方便定位具体问题
203	原因：服务器错误，第一次联网验证时，因服务器问题，没有验证通过； 解决方法：将错误日志及重现流程反馈给合合信息技术支持。
204	原因：网络错误，第一次联网验证时，没有网络连接，导致没有验证通过； 解决方法：检查网络是否畅通，尝试打开 4G 网络是否可以正常运行；如果在联网状态仍然报错，请将错误日志及重现流程反馈给合合信息技术支持。
205	原因：包名/签名错误，授权的 APP_KEY 与绑定的 APP 包名和签名都不匹配； 解决方法：请检查工程中的包名、MD5 签名是否与提供给合合信息授权绑定的包名、签名信息一致。

4.4 行驶证预览识别图片要求

预览相机建议最低的分辨率是 1280*720，经过大量测试这个分辨率是识别率最好，兼容性最强的设置。

5、维护与故障说明

1. SDK 维护说明：

对 SDK 的调用，请按照 SDK 文档说明进行开发与调用。集成 SDK 时，验证 SDK 提供的功能是否都可以正常工作。

2. SDK 故障排除说明：

通过提供的 demo，对清晰的行驶证图片进行识别，测试返回的识别结果是否正确。如果

可以正确返回识别结果，则说明 SDK 运行正常。

如果不能返回识别结果或识别结果有错，请提供测试的图片与 SDK log 给我们分析：
bryant_shen@intsig.net。

6 调用示例代码

6.1 调用 SDK 相机模块的 API 识别的示例

如下（这个 ISCardScanActivity 类是在 SDK 的 jar 包中相机预览类，不可以修改，但是可以传入参数达到不同效果）：

```
//通过Intent调用SDK中的相机拍摄模块ISCardScanActivity进行识别
Intent intent = new Intent(this, ISCardScanActivity.class);
//指定要临时保存图片的路径
intent.putExtra(ISCardScanActivity.EXTRA_KEY_IMAGE_FOLDER, DIR_IMG_RESULT);
//指定SDK相机模块ISCardScanActivity四边框角线条,检测到图片后的颜色
intent.putExtra(ISCardScanActivity.EXTRA_KEY_COLOR_MATCH, 0xffff0000);
//指定SDK相机模块ISCardScanActivity四边框角线条颜色,正常显示颜色
intent.putExtra(ISCardScanActivity.EXTRA_KEY_COLOR_NORMAL, 0xff00ff00);
//合合信息授权提供的APP_KEY
intent.putExtra(ISCardScanActivity.EXTRA_KEY_APP_KEY, APP_KEY);
//指定SDK相机模块ISCardScanActivity提示字符串
intent.putExtra(ISCardScanActivity.EXTRA_KEY_TIPS, "请将行驶证放在框内识别");
startActivityForResult(intent, REQ_CODE_CAPTURE);
```

6.2 调用 SDK 方法 API 识别的示例

如下（这个 PreviewActivity 类是提供给用户自定义调整的相机预览类，可供参考，客户可以按照自身具体需求调整相机预览效果，具体如何使用 PreviewActivity 请参考【[7.相机预览拍照的使用文档](#)】）。

```
//在PreviewActivity中自定义相机拍摄模块,然后直接调用SDK中的预览方法进行识别
//在PreviewActivity中可以直接获得SDK返回的ResultData识别结果
Intent intent = new Intent(this, PreviewActivity.class);
//合合信息授权提供的APP_KEY
intent.putExtra(PreviewActivity.EXTRA_KEY_APP_KEY, APP_KEY);
startActivityForResult(intent, REQ_CODE_CAPTURE);
```

7 相机预览拍照的使用文档

打包文件同目录下有【相机预览识别说明文档_v*****.pdf】文件，请查询，如果没有请联

系本公司技术支持人员提供相应文档。

8 修订历史记录

更新日期	更新内容	更新版本
2017.7.7	重新调整文档布局内容	v1.0.0.2017.7.7