

香港身份证识别 **SDK-Android** 使用文档

2019年8月

版本 1.1.9



© 2019 TransUnion LLC
All Rights Reserved

No part of this publication may be reproduced or distributed in any form or by any means, electronic or otherwise, now known or hereafter developed, including, but not limited to, the Internet, without the explicit prior written consent from TransUnion LLC.

Requests for permission to reproduce or distribute any part of, or all of, this publication should be mailed to:

Law Department
TransUnion
555 West Adams
Chicago, Illinois 60661

The "T" logo, TransUnion, and other trademarks, service marks, and logos (the "Trademarks") used in this publication are registered or unregistered Trademarks of TransUnion LLC or their respective owners. Trademarks may not be used for any purpose whatsoever without the express written permission of the Trademark owner.

transunion.com

历史版本

版本	描述	准备者	更新时间
1.0	起始版本	TU	2019 年 6 月 26 日
1.1	文档新增对系统和 CPU 架构支持的描述 对代码结构进行描述 对详细代码逻辑进行说明	TU	2019 年 7 月 23 日
1.1.9	更新状态处理	TU	2019 年 8 月 6 日

目录

1. 引言	5
1.1. 目的	5
1.2. 文档范围	5
2. SDK 接入.....	5
2.1. 操作步骤	5
3. SDK 调用.....	8
3.1. 调用步骤	8
3.2. 返回结果参数说明	8
4. SDK 源代码描述	9
4.1. 代码 ZaHKIDActivity 描述说明	9
4.2. 源代码调用初始化说明	9
4.3. 扫描检测证件说明	12
4.4. 消息类型描述说明	14

1. 引言

1.1. 目的

方便第三方调用香港身份证验证SDK。

1.2. 文档范围

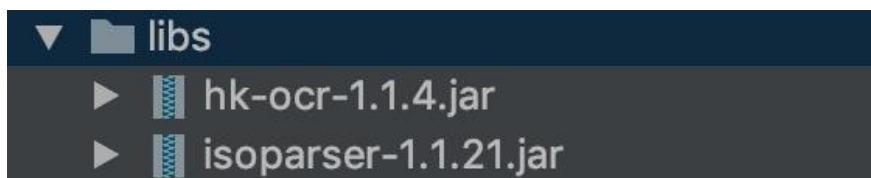
支持 Android 5.0+以上系统，支持 32 位和 64 位 CPU 架构

2. SDK 接入

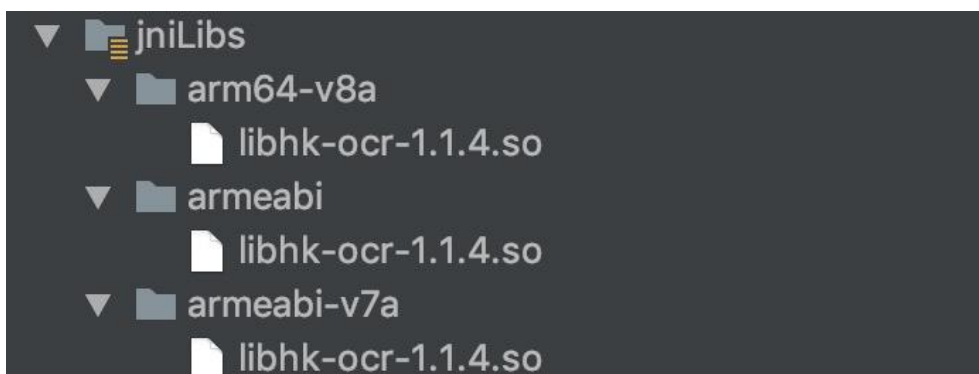
注意:文档中的版本号只是示例，真实使用需要用我们提供的最新的版本号对应

2.1. 操作步骤

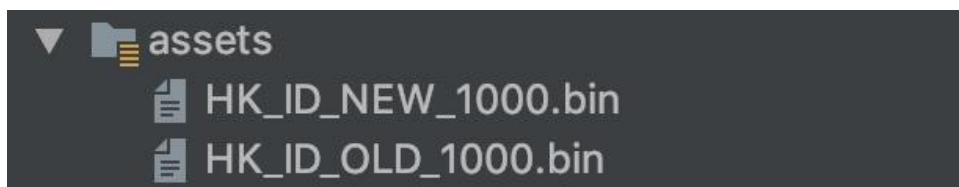
- 在App的 libs中加入:



- 在App的jniLibs中加入:



- 在App的assets中添加bin文件:



- 在App的build.gradle中配置aar：

```
dependencies {
    implementation fileTree(include: ['*.jar'], dir: 'libs')
    implementation 'com.android.support:appcompat-v7:27.1.1'
    implementation 'com.android.support:cardview-v7:22.1.0'
    implementation 'com.android.support.constraint:constraint-layout:1.1.3'
    implementation 'com.android.support:support-annotations:27.1.1'
    implementation(name: 'SDKAuthCheckForAnLink-1.2.1-20190127-release', ext: 'aar') {
        transitive = true
        changing = true
    }
    api project(':livenessLib_sdk')
    implementation files('libs/hk-ocr-x.x.x.jar')
}
```

在 AndroidManifest.xml 中配置相应的权限和识别页面: (注意: keepScreenOn="true"设置当前界面不熄屏)

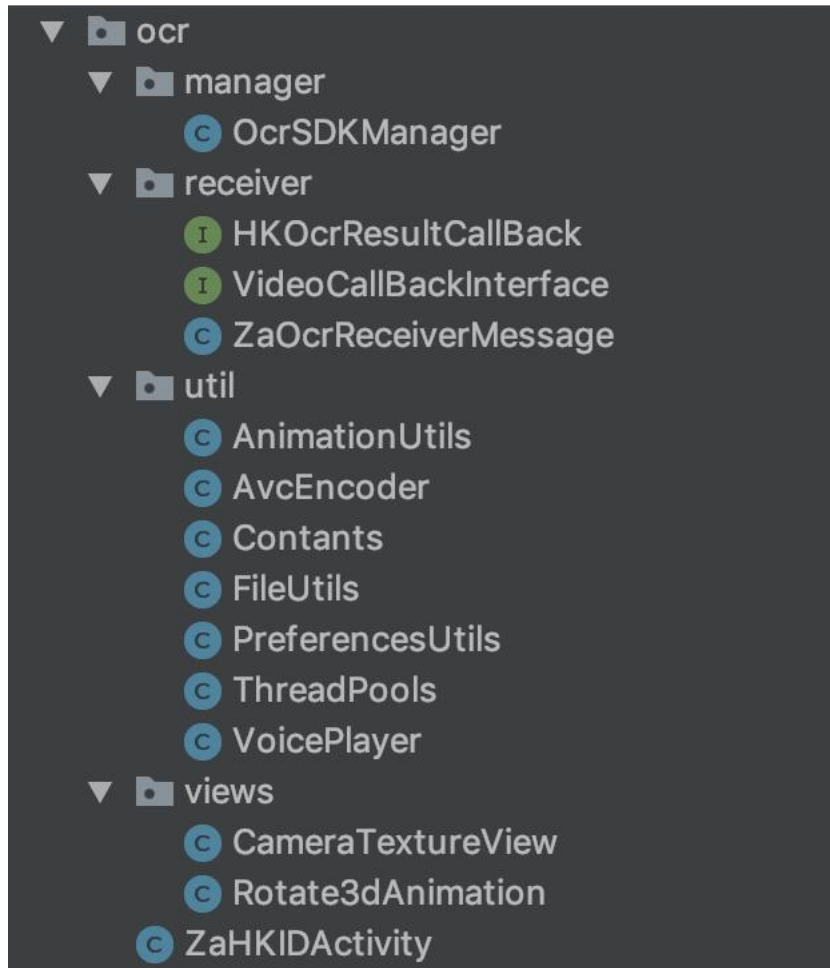
```
<uses-permission android:name="android.permission.CAMERA" />
<uses-feature android:name="android.hardware.camera" />
<uses-feature android:name="android.hardware.camera.autofocus" />
<uses-permission android:name="android.permission.FLASHLIGHT" />
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE" />
<uses-permission android:name="android.permission.WRITE_EXTERNAL_STORAGE" />
```

```
<activity android:name=".ocr.ZaHKIDActivity"
    android:screenOrientation="portrait"
    android:keepScreenOn="true"
    android:theme="@android:style/Theme.Holo.Light.NoActionBar" />
```

- 在启动页面动态申请权限:

```
ActivityCompat.requestPermissions( activity: this, new String[]{Manifest.permission.INTERNET,
    Manifest.permission.WRITE_EXTERNAL_STORAGE,
    Manifest.permission.READ_EXTERNAL_STORAGE,
    Manifest.permission.CAMERA,
    Manifest.permission.RECORD_AUDIO
}, (requestCode: 1);
```

- 将以下源代码加 App 代码中:



3. SDK 调用

3.1. 调用步骤

- 在 App 的启动香港身份证识别模块的页面中调用:

```
//初始化视频存储路径
String videoBasePath=ZaOcrApplications.context().getExternalCacheDir()+"/.ocrFile/";
OcrSDKManager.getInstance().initVideoPath(videoBasePath);

findViewById(R.id.newOcrButton).setOnClickListener((v) -> {
    //启动新版证件SDK
    OcrSDKManager.getInstance().startNewIdentifyIDCard( context: DemoActivity.this, isDebug: true);
});

findViewById(R.id.oldOcrButton).setOnClickListener((v) -> {
    //启动旧版证件SDK
    OcrSDKManager.getInstance().startOldIdentifyIDCard( context: DemoActivity.this, isDebug: true);
});
```

- 在 App 类继承 HKOcrResultCallBack 接口并实现 result 方法:

```
OcrSDKManager.getInstance().addHKOcrResultCallBack(this);
```

3.2. 返回结果参数说明

```
@Override
public void result(int resultCode,String result) {
    switch (resultCode){
        case Contants.CALLBACKDATA_CANCEL:
            Toast.makeText( context: this, "取消检测", Toast.LENGTH_SHORT).show();
            break;
        case Contants.CALLBACKDATA_SUCCESS:
            Log.e( tag: "返回结果",result);
            break;
        case Contants.CALLBACKDATA_VIDEO_FAIL:
            Toast.makeText( context: this, "视频录制失败", Toast.LENGTH_SHORT).show();
            break;
        case Contants.CALLBACKDATA_EXCEED:
            Toast.makeText( context: this, "检测超时", Toast.LENGTH_SHORT).show();
            break;
    }
}
```

```
public static final int CALLBACKDATA_SUCCESS;    //检测成功
public static final int CALLBACKDATA_VIDEO_FAIL; //视频录制失败
public static final int CALLBACKDATA_CANCEL;     //用户取消操作
public static final int CALLBACKDATA_EXCEED;    //检测超时
```


4. SDK 源代码描述

4.1. 代码 ZaHKIDActivity 描述说明



4.2. 源代码调用初始化说明

- 进入 ZaHKIDActivity, 首先进行算法模型初始化

```
//初始化算法模型
OcrNative.getInstance().OcrJniOcrInit(OcrNative.getInstance().getBinPath(),Contants.isOldOrNew);
OcrNative.getInstance().OcrJniOcrStart(Contants.isOldOrNew);
```

- 设置是否检测反光和检测速度过快

```
//设置是否检测反光 不设置默认为检测
ZaOcrSDKManager.getInstance().setExposure(false); //true 检测 false不检测
//设置是否检测速度过快 不设置默认为检测
ZaOcrSDKManager.getInstance().setOverspeed(false); //true 检测 false 不检测
```

- 设置证件检测超时，并且调用 `message` 方法统一管理消息类型

```
//设置超时时间 单位为秒
ZaTimerTask.getInstance().setZaTimer(60);
//设置超时返回
ZaTimerTask.getInstance().addZaTimerTaskCallBack(new ZaTimerTaskCallBack() {
    @Override
    public void exceedResult() {
        DemoZaHKIDActivity.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                pushResult( msg: "EXCEED");
            }
        });
    }
});
//执行超时
ZaTimerTask.getInstance().excute();
```

- 注册 OCR 识别结果返回监听，添加 OCR 识别状态消息实时返回监听，添加视频录制保存结束返回监听

```
//注册ocr识别结果返回监听
OcrNative.getInstance().setOcrNativeBackListener(ZaNewHKIDActivity.this);
//添加ocr识别消息实时返回监听
OcrNative.getInstance().addOcrReceiveMessageListener(ZaNewHKIDActivity.this);
//添加视频录制保存结束返回监听
ZaOcrReceiverMessage.getInstance().addVideoCallBackInterface(videoCallBackInterface);
```

- OCR 识别结果返回监听说明

```
/**
 * 算法返回反光，速度过快消息，并且识别证件返回证件正面图片和人脸图片
 * @param result
 */
@Override
public void onIdentifyFinish(final OCRResult result) {
    ThreadPools.execute(new Runnable() {
        @Override
        public void run() {
            if(!TextUtils.isEmpty(result.desc)){
                message(result.desc);
            }
            if(result.data!=null && result.size>0){
                if("first".equals(result.desc)) {
                    //算法返回来第一帧为香港身份证正面图，文件生成base64
                    templateImage= Base64.encodeToString(result.data.get(0), Base64.DEFAULT);
                    FileUtils.createFileWithByte(result.data.get(0), AvcEncoder.za_hk_idcard);
                    //算法返回来香港身份证图像的照片
                    FileUtils.createFileWithByte(result.data.get(result.data.size() - 1), AvcEncoder.za_hk_face);
                }
            }
        }
    });
}
```

`OcrNative.getInstance().setOcrNativeBackListener(ZaNewHKIDActivity.this);`

- OCR 识别消息实时返回监听

OcrNative.getInstance().addOcrReceiveMessageListener(ZaNewHKIDActivity.this)

```
private void pushResult(final String msg){
    ZALog.e( tag: "message=====>",msg);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            switch (msg.toUpperCase()) {
                case "LOST"://目标(证件)丢失,在识别证件过程中,发现某些证件特征持续捕获不到导致识别失败
                    break;
                case "EXPOSURE"://检测到证件有反光
                    break;
                case "OVERSPEED"://检测到证件翻转速度过快
                    break;
                case "EXCEED"://超时 (SDK自定义超时状态)
                    break;
            }
        }
    });
}
```

```
private void pushMessage(final String msg){
    ZALog.e( tag: "message=====>",msg);
    runOnUiThread(new Runnable() {
        @Override
        public void run() {
            switch (msg.toUpperCase()) {
                case "ORTH"://检测到正面
                    break;
                case "RESET"://复位
                    break;
                case "UP"://检测到第一个翻转动作
                    break;
                case "DOWN"://检测到最后一翻转动作
                    break;
                case "FAR"://距离过远
                    break;
                case "CLOSE"://距离过近
                    break;
                case "VALID"://恢复正常 恢复到上一个提示语
                    break;
                case "COMPLETE":
                    break;
            }
        }
    });
}
```

- OCR 添加视频录制保存结束返回监听

SDK 必须等到视频录制完成才可进行下一步操作,如果视频录制失败,则返回到 App 界面, 让用户重新进行扫描识别

```

VideoCallBackInterface videoCallBackInterface=new VideoCallBackInterface() {
    @Override
    public void videoFinished() {
        ZALog.d( tag: "视频录制", msg: "videoFinished"+"=====");
        ZAHKIDActivity.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                if(FileUtils.getFileSize(AvcEncoder.mp4)>0) {
                    Intent data = new Intent();
                    JSONObject json = new JSONObject();
                    JSONObject params = new JSONObject();
                    try {
                        params.put( name: "dRecStartTimeStamp", value: Contents.dRecStartTimeStamp + "|" + Contents.dRecStartTimeStamp + "|" + Contents.d
                        json.put( name: "videoTimeStamp", params.toString());
                        json.put( name: "file", AvcEncoder.mp4);
                        json.put( name: "idcard", AvcEncoder.za_hk_idcard);
                        json.put( name: "templateImage", templateImage);
                        json.put( name: "idcardFace", AvcEncoder.za_hk_face);
                        json.put( name: "sdkVersion", ZAHKIDActivity.this.runOnUiThread(new Runnable() {
                            @Override
                            public void run() {
                                OcrSDKManager.getInstance().getHK0crResultCallBack().result(Contents.CALLBACKDATA_SUCCESS, json.toString());
                                ZAHKIDActivity.this.finish();
                            }
                        });
                    } catch (JSONException e) {
                        e.printStackTrace();
                    }
                } else {
                    OcrSDKManager.getInstance().getHK0crResultCallBack().result(Contents.CALLBACKDATA_VIDEO_FAIL, result: null);
                    ZAHKIDActivity.this.finish();
                }
            }
        });
    }

    @Override
    public void videoError() {
        ZAHKIDActivity.this.runOnUiThread(new Runnable() {
            @Override
            public void run() {
                OcrSDKManager.getInstance().getHK0crResultCallBack().result(Contents.CALLBACKDATA_VIDEO_FAIL, result: null);
                ZAHKIDActivity.this.finish();
            }
        });
    }
};

```

4.3. 扫描检测证件说明

- 在开始扫面之前有用户操作指引页面

先关闭给算法模型传递视频帧，防止在用户知道页面，给算法传递视频帧
`mCameraView.stopPushBuffer();`

当用户操作点击开始扫描，开始检测视频帧，给算法模型传递视频帧
`mCameraView.startPushBuffer();`

(如果不需要用户引导页，直接进来打开摄像头进行扫描检测，去掉这一句，因为默认直接进来开启摄像头，开始检测)

- 当打开检测证件反光和旋转速度过快，如果发生证件反光或旋转速度过快，会有弹框，会让用户自己进行点击重新开始识别，具体操作参考代码，有详细的注释说明

(注意：重新识别需要重新加载算法模型，并且超时计时器也需要重新设置，并且注意一下返回监听描述)

- 当在扫描时，不管任何原因关闭，正常关闭 SDK 识别界面，需要做以下释放

```
@Override
public void onDestroy() {
    //停止录制视频
    ZaOcrReceiverMessage.getInstance().stopVedio();
    //取消超时
    ZaTimerTask.getInstance().cancelTimer();
    //关闭动画
    AnimationUtils.getInstance().animationClosed();
    //停止预览
    mCameraView.stopCamera();
    //关闭摄像头
    mCameraView.closeCamera();
    //注销声音播放器
    VoicePlayer.getInstance().stopSound();
    //注销算法模块
    OcrNative.getInstance().OcrJniDestory(Contants.isOldOrNew);
    //注销检测视频帧监听
    OcrNative.getInstance().removeOcrReceiveMessageListener(this);
    //注销视频录制完成监听
    ZaOcrReceiverMessage.getInstance().removeVideoCallBackInterface();
    //注销线程
    ZaOcrReceiverMessage.getInstance().stopThread();

    if(exposureDialog!=null){
        exposureDialog.dismiss();
        exposureDialog=null;
    }

    super.onDestroy();
}
```

- 当在扫描时，用户可能想取消操作，会点击手机物理返回按键，或自定义关闭按钮， SDK 里面有一下操作通知到 App 用户进行了取消操作

拦截系统物理返回按键

```
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if(keyCode==KeyEvent.KEYCODE_BACK){
        OcrSDKManager.getInstance().getHKOcrResultCallBack().result(Constants.CALLBACKDATA_CANCEL, result: null);
        ZaHKIDActivity.this.finish();
    }
    return super.onKeyDown(keyCode, event);
}
```

自定义返回按钮

```
cancelBtn=(ImageView)findViewById(R.id.cancelBtn);
cancelBtn.setOnClickListener((v) -> {
    OcrSDKManager.getInstance().getHKOcrResultCallBack().result(Constants.CALLBACKDATA_CANCEL, result: null);
    ZaNewHKIDActivity.this.finish();
});
```

4.4. 消息类型描述说明

- 共有状态:

ORTH:	检测到证件正面时的状态
COMPLETE:	检测证件完成时的状态
EXPOSURE:	检测到证件反光时的状态
OVERSPEED:	检测到证件旋转速度过快时的状态
LOST:	目标(证件)丢失，在识别证件过程中，发现某些证件特征持续捕获不到导致识别失败
RESET:	动作复位回到正面位置
EXCEED:	自定义超时消息状态
FAR:	检测到证件距离过远
CLOSE:	检测到证件距离过近

(注意：在 EXPOSURE，OVERSPEED，EXCEED 和 LOST 类型的时候，需要重新调用 OcrJniOcrStart 方法)

- 独有状态

UP:	检测到证件第一个翻转状态
DOWN:	检测到证件最后一个翻转状态