

Android 在线活体SDK 集成帮助文档

南京壹证通信息科技有限公司

版本历史

时间	版本	描述
2023/10/13	2.0.0	优化了底层算法，提高了安全性和识别速度

目录

1 引言

1.1 概述

1.2 开发平台及开发语言

1.3 SDK兼容性

1.4 功能介绍

2 SDK集成

2.1 SDK内容说明

2.2 SDK导入

3 SDK参数配置

4 活体检测sdk调用

5 活体检测sdk中关于人脸采集页面的UI自定义

5.1 通过配置的形式修改人脸采集页面的颜色

5.2 通过配置修改人脸采集页面上的icon

6 注意事项

7 混淆说明

8 版权说明

1 引言

1.1 概述

壹证通活体检测sdk Android 版是一种面向 Android 移动设备人脸技术开发包，此版SDK包含人脸检测、活体识别等功能，以aar包（aar包中包含动态链接库）的形式发布。基于该方案，开发者可以轻松的构建包含人脸检测、采集和活体识别的应用

活体检测 SDK 的集成和使用请仔细阅读本手册，对需要特别注意的地方，手册中将加以提醒。

1.2 开发平台及开发语言

开发平台：**Android Studio** (建议使用3.0.1以上的release版本)

开发语言：**java**

1.3 SDK兼容性

系统：支持 Android 5.0(API Level 21)及以上系统。需要开发者通过 minSdkVersion来保证支持系统的检测。

机型：手机和平板皆可

构架：支持 CPU架构平台【arm-v7a, arm64-v8a】

网络：支持 WIFI 及移动网络,移动网络支持使用NET 网关及 WAP 网关(CMWAP、CTWAP、UNIWAP、3GWAP)。

1.4 功能介绍

- **本地版活体检测：**通过让用户做出指定人脸配合式的交互动作，识别当前操作者是否为活体，此功能为离线使用，可设定指定动作是否使用及应用顺序。动作包含：眨眼、张嘴、左摇头、右摇头、向上抬头，向下低头六个。可有效抵御高清图片、3D建模、视频等攻击。
- **本地版人脸质量检测：**判断视频流中的图片帧中，哪些图片质量较佳，即人脸图像特征清晰（满足角度、姿态、光照、模糊度等校验）。
- **本地版人脸图像采集：**通过本地SDK能力，采集人脸图像，同时经过人脸质量检测，确保采集到的人脸图像符合各条件校验（角度、姿态、光照、模糊度等），为设备前端获取有效可分析人脸的主要功能。
- **离线授权：**SDK的授权判断，授权介质也称为license，在SDK使用中，需要通过license向授权服务器发起请求，判断SDK的使用合法性及使用有效期。

注意：license的授权文件需要您工程的applicationId（即应用包名）和应用的签名文件的MD5值。授权文件存放在assets目录下具体流程可联系壹证通商务

2 SDK集成

2.1 SDK内容说明

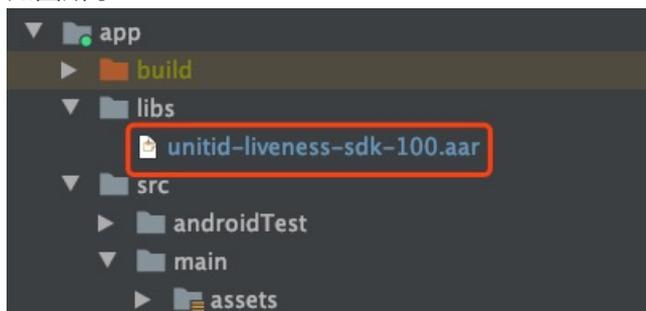
1、活体检测sdk只有一个unitid-liveness-sdk-xxx.aar的aar包，该包中包含了动态链接库。开发者只需要导入该aar包即可。

2、SDK在使用之前需要向大识蛛平台提供您工程的applicationId（即应用包名）和应用的签名文件的MD5值注册您的应用。具体的注册方式，可联系商务解决

2.2 SDK导入

将unitid-liveness-sdk-xxx.aar放入libs包下面

如图所示：



gradle配置如图所示：

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

3 SDK参数配置

1、授权文件idl-license.face-android放到assets目录下

2、在UnitidFaceConfig类中配置授权文件和授权id

如下：

```
public class UnitidFaceConfig {  
  
    /**  
     * 活体检测sdk授权id  
     */  
    public static String licenseID = "nanjingunitidcn-face-android";  
    /**  
     * 活体检测sdk的授权文件，存放在assets目录下名称一定要一致  
     */  
    public static String licenseFileName = "idl-license.face-android";  
  
}
```

3、在自定义application中初始化活体检测sdk

如下：

```
/**  
 * 初始化活体检测SDK  
 */  
private void initLib() {  
    // SDK初始化  
    IdentityVerificationSdk.initializeSDK(this, UnitidFaceConfig.licenseID,  
        UnitidFaceConfig.licenseFileName, new IInitCallback() {  
        @Override  
        public void initSuccess() {  
            LogUtils.i(TAG, "初始化成功");  
        }  
    });  
}
```

```

    }

    @Override
    public void initFailure(int paramInt, String paramString) {
        LogUtils.e(TAG, "初始化失败 = "+paramInt+" "+paramString);
    }
});
}

```

4、权限声明

名称	用途
android.permission.INTERNET	允许应用联网,SDK联网授权。
android.permission.READ_PHONE_STATE	获取用户手机的IMEI,用来唯一的标识用户
android.permission.CAMERA	允许调用相机进行拍照
android.hardware.camera.autofocus	允许相机对焦
android.permission.WRITE_EXTERNAL_STORAGE	图片裁剪临时存储

4 活体检测sdk调用

sdk的调用如下:

```

Intent intent = new Intent(this, FaceSecurityLivenessActivity.class);
intent.putExtra(FaceConstants.TOKEN, authToken);
startActivityForResult(intent, REQUEST_CODE);

```

参数描述

- `authToken`: 从服务端获取到授权token, 必填项。

具体的调用事例, 请参考demo

流程结束后SDK会将核验的结果回调给前端:

```

@Override
protected void onActivityResult(int requestCode, int resultCode, @Nullable Intent data) {
    if (RESULT_OK == resultCode && REQUEST_CODE == requestCode) {
        if (data != null) {
            int result = data.getIntExtra(FaceConstants.RESULT_CODE, -1);
            String message = data.getStringExtra(FaceConstants.MESSAGE);
            switch (result) {
                case FaceResultCode.SUCCESS:
                    // 成功
                    ToastUtil.showBottomToast(this, "核验成功");
                    break;
            }
        }
    }
}

```

```

        case FaceResultCode.INVALID_TOKEN:
            // 无效的token, 需要重新获取token
            ToastUtil.showBottomToast(this, "无效的token");
            break;
        case FaceResultCode.DETECT_TIME_OUT:
            // 活体检测超时, 可以拉起人脸采集页面进行重试, 不需要在token有效期内不需要重

新获取token

            ToastUtil.showBottomToast(this, "活体检测超时");
            break;
        case FaceResultCode.CANCEL:
            // 用户取消操作, 可以拉起人脸采集页面进行重试, 不需要在token有效期内不需要重

新获取token

            ToastUtil.showBottomToast(this, "用户取消操作");
            break;
        case FaceResultCode.FACE_VERIFY_FAILURE:
            // 人脸验证失败, 如果需要重试, 需要重新获取token拉起人脸采集页面进行重试
            ToastUtil.showBottomToast(this, message);
            break;
        case FaceResultCode.FAILURE:
            // 其他失败, 可以拉起人脸采集页面进行重试, 不需要在token有效期内不需要重新获

取token

            ToastUtil.showBottomToast(this, "其他失败");
            break;
        default:
            break;
    }
}
}
super.onActivityResult(requestCode, resultCode, data);
}

```

错误码描述:

- FaceResultCode.SUCCESS : 核验成功
- FaceResultCode.CANCEL : 用户取消操作
- FaceResultCode.INVALID_TOKEN : 无效的token
- FaceResultCode.DETECT_TIME_OUT : 活体检测超时
- FaceResultCode.FACE_VERIFY_FAILURE : 人像数据验证失败
- FaceResultCode.FAILURE : 其他失败

5 活体检测sdk中关于人脸采集页面的UI自定义

5.1 通过配置的形式修改人脸采集页面的颜色

```
<!--人脸采集页面背景色-->
<color name="spider_detect_page_background">#FFFFFF</color>
<!--导航栏颜色-->
<color name="spider_navigation_bar_color">#FFFFFF</color>
<!--状态栏颜色-->
<color name="spider_status_bar_color">#FFFFFF</color>
```

以上是SDK中人脸采集页面的默认配色，如果您需要更改配色，直接在您的color配置文件中以同名的方式重新定义颜色即可。

5.2 通过配置修改人脸采集页面上的icon

人脸采集页面关闭icon, ic_face_detect_close_ext.png  宽高为96*96

人脸采集页面静音icon, ic_face_detect_disable_sound_ext.png  宽高为96*96

人脸采集页面开启声音icon, ic_face_detect_enable_sound_ext.png  宽高为96*96

以上是SDK中人脸采集页面中用到的默认的icon，如果您需要更改，您可以在mipmap-xxhdpi文件夹下，以同名的方式替换成您的自定义icon

6 注意事项

程序运行

debug模式下的运行也需要携带签名，也就是说在任何模式下的运行都需要携带签名文件，SDK需要对授权文件，签名文件，包名等进行合法性校验。

7 混淆说明

```
-keep class cn.unitid.liveness.** { *; }
-keep class cn.unitid.security.data.** { *; }
-keep class com.baidu.** { *; }
```

8 版权说明

版权归[南京壹证通信息科技有限公司](#)所有，未经许可禁止翻印。