

# 数据宝活体检测人像比对产品对接说明

( Android )

2018 年 4 月 23 日  
数据宝技术部

## 目录

1. 活体检测人像比对 SDK 介绍 .....	3
1.1. SDK 详细内容 .....	3
1.2. 活体检测人像比对流程介绍 .....	3
1.3. Sample 示例 .....	4
2. 开发集成指南 .....	5
2.1. 导入 aar 文件包在主 model 下的 libs 文件夹下 .....	5
2.2. 在 gradle 中引入该 aar 文件 .....	5
2.3. 开发指南 .....	6
2.3.1. 初始化和可用产品检查 .....	6
2.3.2. OCR+活体检测+实名认证+人像比对产品 .....	7
2.3.3. 实名认证+活体检测+人像比对产品 .....	8
3. 函数库说明 .....	8
3.1. CDPDataApi .....	8
3.2. 回调结构说明 .....	10
3.2.1. OCR 识别接口返回结果说明 .....	10
3.2.2. 实名认证接口返回结果说明 .....	11
3.2.3. 比对服务接口返回结果说明 .....	11
3.2.4. 活体检测返回结果说明 .....	12

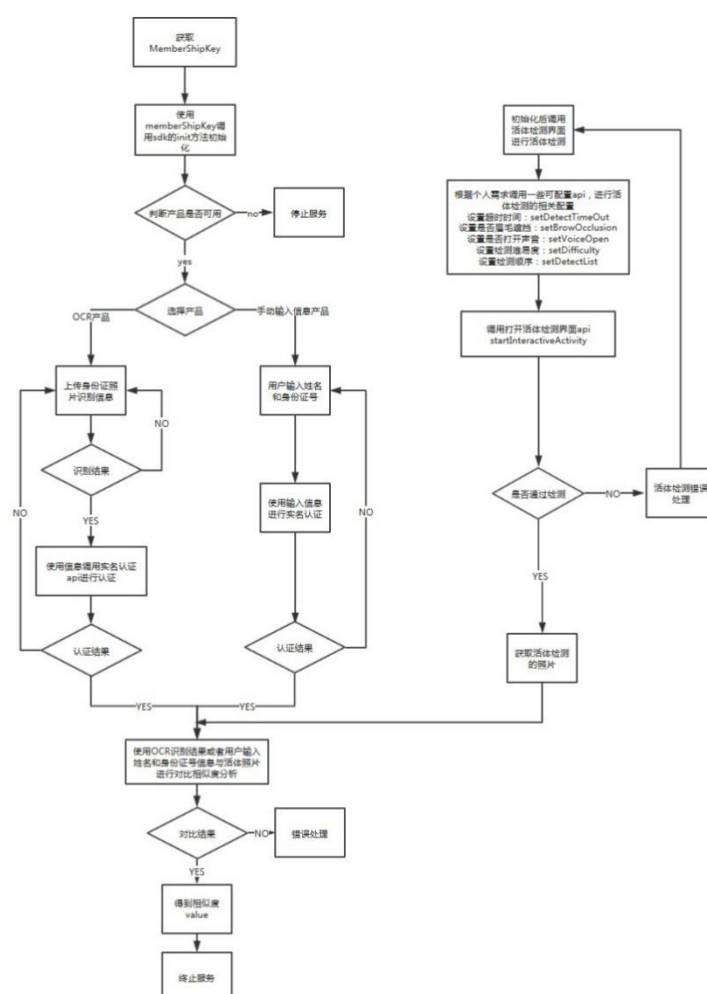
## 1.活体检测人像比对 SDK 介绍

### 1.1.SDK 详细内容

SDK 主要包括四个方面内容，如下图所示



### 1.2.活体检测人像比对流程介绍



### 1.3.Sample 示例

#### ➤第一步初始化鉴权和产品页面和实名认证输入

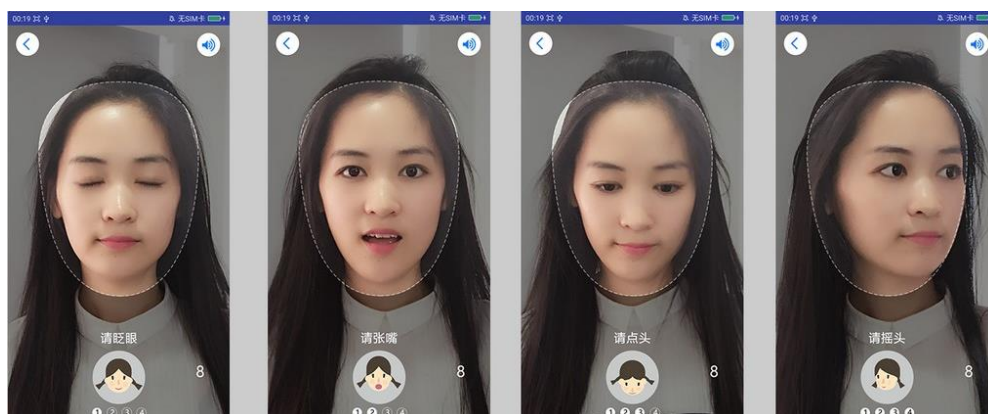


#### ➤第二步活体检测

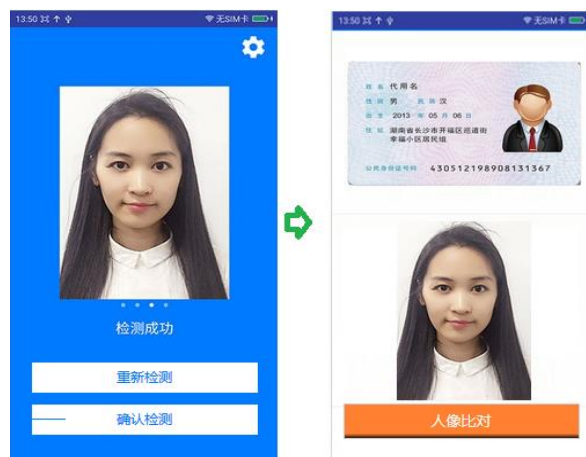
##### A.人脸对正



##### B.根据提示做出相应动作，动作完成后，自动进行活体检测输出结果



##### C.确认检测后进入对比

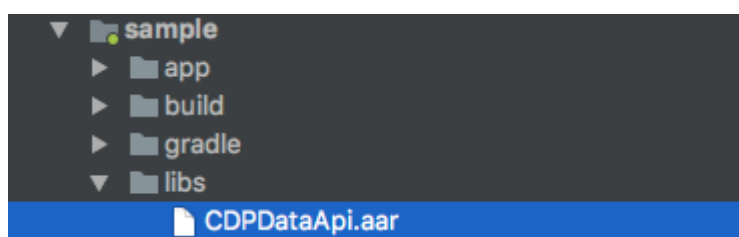


### ➤第三步人像比对



## 2.开发集成指南

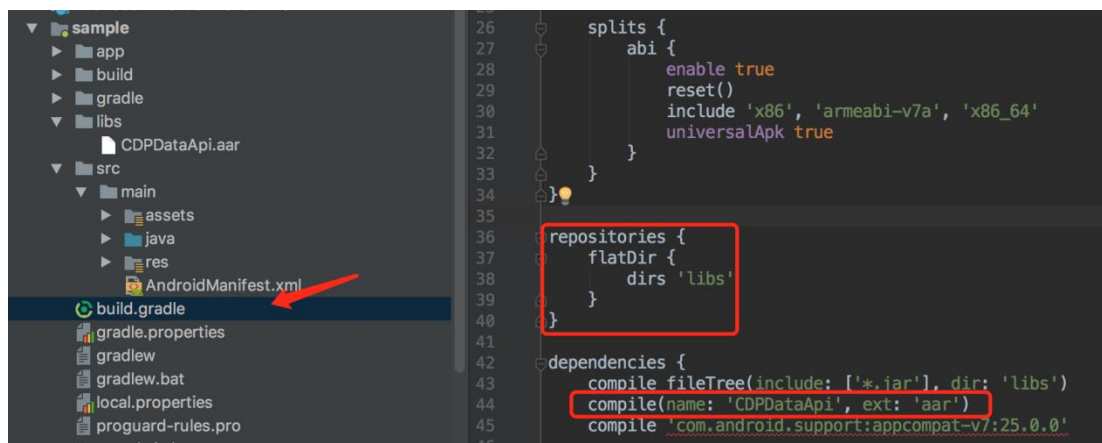
### 2.1.导入 aar 文件包在主 model 下的 libs 文件夹下



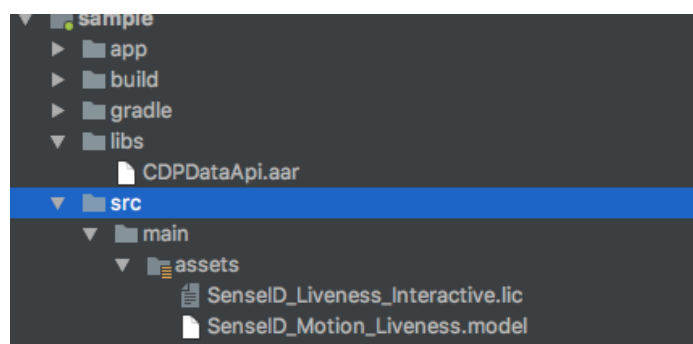
### 2.2.在 gradle 中引入该 aar 文件

```
// 在 gradle 下加上如下代码
repositories {
    flatDir {
        dirs 'libs'
    }
}

dependencies {
    compile(name: 'CDPDataApi', ext: 'aar')
}
```



## 2.3.在主 model 的 assets 目录下导入获取到的 lic 验证文件和 model 文件



请确保文件名称正确：

SenseID\_Liveness\_Interactive.lic

SenseID\_Motion\_Liveness.model

## 2.4.开发指南

### 2.4.1.初始化和可用产品检查

```

CDPDataApi.getInstance().init(getApplicationContext(), mLivenessListener, memberShip, new CallBackListener() {
    @Override
    public void onSuccess(String s) {
        BaseModel<CheckUserModel> data = GsonUtils.getInstance().fromJson(s, CheckUserModel.class);
        if (data != null && data.data != null) {
            if (data.data.interfaceProductOne != null && data.data.interfaceProductOne.size() > 0) {
                // OCR 产品可用
            }
            if (data.data.interfaceProductTwo != null && data.data.interfaceProductTwo.size() > 0) {
                //实名认证产品可用
            }
        }
    }
    @Override
    public void onFailure(String s) {
        ToastUtils.getInstance(InitActivity.this).showToast(s);
    }
});
    
```

## 2.4.2.OCR+活体检测+实名认证+人像比对产品

### 2.4.2.1.将身份证照片上传到服务器获取身份证号和姓名

```
CDPDataApi.getInstance().callCardOcrApi(imageStr, new CallBackListener() {
    @Override
    public void onSuccess(String s) {
        //解析 json 数据 获取对应信息
    }
    @Override
    public void onFailure(String s) {
        //请求失败
    }
});
//对于摄像头的定制详见 demo 源码，图片的处理在 com.sensetime.sample.motionliveness.util.ImageUtils 类
```

### 2.4.2.2.将姓名和身份证号上传到服务器进行实名认证

```
CDPDataApi.getInstance().callImageCompareApi(data.data.name,
    data.data.idcard, imageStr, line, new CallBackListener() {
    @Override
    public void onSuccess(String s) {
        //解析 json 数据 获取对应信息
    }
    @Override
    public void onFailure(String s) {
        //请求失败
    }
});
```

### 2.4.2.3.使用活体检测功能获取活体照片

```
CDPDataApi.getInstance().startInteractiveActivity(this, getApplicationContext());
@Override
protected void onActivityResult(int requestCode, int resultCode, Intent data) {
    super.onActivityResult(requestCode, resultCode, data);
    if (resultCode == RESULT_OK) {
        //根据保存的路径取得活体检测图片
        Bitmap bitmap = IntentUtils.getBimmapFromPath();
    }else{
        //处理错误情况 视返回码而定
    }
}
在 onActivityResult 中获取检测的状态码，RESULT_OK 代表检测完成，其他状态码详见文档最后说明
IntentUtils 里面封装了获取活体检测后图片的代码，直接调用即可获得 bitmap 对象，进行显示
```

### 2.4.2.4.将姓名，身份证号和活体检测图片上传获取相似度结果

```
CDPDataApi.getInstance().callImageCompareApi(data.data.name,
    data.data.idcard, imageStr, line, new CallBackListener() {
    @Override
    public void onSuccess(String s) {
        //解析 json 数据 获取对应信息
    }
    @Override
    public void onFailure(String s) {
        //请求失败
    }
});
```

## 2.4.3.实名认证+活体检测+人像比对产品

### 2.4.3.1.输入姓名和身份证号进行实名认证

//参数 line:根据产品线路区分, 传 1 或 2, 此处传 2, 其他同 2.3.2.2

### 2.4.3.2.使用活体检测功能获取活体照片

//同 2.3.2.3

### 2.4.3.3.将姓名, 身份证号和活体检测图片上传获取相似度结果

//参数 line 根据产品线路区分, 传 1 或 2, 此处传 2, 其他同 2.3.2.4

## 3.函数库说明

### 3.1.CDPDataApi

#### CallBackListener

通用简单回调

void onSuccess(String var1):

返回的是请求后的json字符串, 根据提供文档, 进行解析

void onFailure(String var1):

返回的是请求失败的原因

#### 初始化

```
public void init(Context context, OnLivenessListener onLivenessListener, String memberShip, CallBackListener callBackListener) {
```

参数	说明
context	上下文
onLivenessListener	活体检测的回调
memberShip	认证序列号
callBackListener	通用回调
返回结果	无



## 实名认证接口

```
public void callRealNameAuthApi(String name, String idCard, int line, final CallBackListener callBackListener){  
    }  
}
```

参数	说明
name	姓名
idCard	身份证号
line	产品线，1：OCR方案，2：文字方案，不填直接返回失败处理
callBackListener	通用回调
返回结果	无

## OCR识别接口

```
public void callCardOcrApi(String photo, final CallBackListener callBackListener) {  
    }  
}
```

参数	说明
photo	身份证转码字符串
callBackListener	通用回调
返回结果	无

## 比对服务

```
public void callImageCompareApi(String name, String idCard, String image, int line, final CallBackListener callBackListener){  
    }  
}
```

参数	说明
name	姓名
idCard	身份号
image	活体识别转码字符串
line	产品线，1：OCR方案，2：文字方案，不填直接返回失败处理
callBackListener	通用回调
返回结果	无

## 活体识别配置项

```
public void setVoiceOpen(boolean isOpen) {  
    }  
}
```

参数	说明
isOpen	是否打开声音
返回结果	无

```
/**
 * public void setDifficulty(String difficulty) {
 *     SPUtil.put(Constants.COMPLEXITY, difficulty);
 * }
 */
```

参数	说明
difficulty	当前活体检测难度 有四种模式 Easy、Normal、Hard、Hell
返回结果	无

```
public void setDetectList(String detectList) {
    SPUtil.put(Constants.DETECTLIST, detectList);
}
```

参数	说明
detectList	活体检测的顺序 默认顺序 [BLINK,NOD,MOUTH,YAW]
返回结果	无

```
public void setDetectTimeOut(long timeOut) {
}
```

参数	说明
timeOut	设置超时时间（默认为10秒）
返回结果	无

```
public void setBrowOcclusion(boolean flag) {
}
```

参数	说明
flag	是否设置眉毛遮挡
返回结果	无

```
public void startInteractiveActivity(Activity context, Context mApplication) {
}
```

参数	说明
context	当前activity上下文对象
mApplication	applicationContext的上下文
返回结果	无

## 3.2.回调结构说明

回调 CallbackListener 中 onSuccess(String var1)返回的 json 结构详细描述如下。

### 3.2.1.OCR 识别接口返回结果说明

```
{
  "code": "10000",
  "message": "成功",
  "data": {
    "birthday": "1994*20",
    "sex": "男",
    "address": "武汉市***1 楼 3 号",
    "name": "小王",
    "idcard": "420***36",
    "image": "/9j/4AAQSAKKEDP/Z",
    "nation": "汉"
  }
}
```

Code 码如下：

返回状态码	说明
10001	识别错误
10002	系统错误
10000	成功
10005	服务器异常
10006	数据库错误
10003	服务器忙
10004	访问超时
10007	无合格图像
10008	失败
SYSTEM_000-044	系统报错

### 3.2.2.实名认证接口返回结果说明

```
{
  "code": "10000",
  "message": "成功",
  "data": {
    "result": "1"
  },
  "seqNo": "4XU29Z4D1704061618"
}
```

result : 查询结果 ( 1 : 验证一致 , 2 : 验证不一致 , 3 : 异常情况 )

返回状态码	说明
10001	报文错误(包含 MAC 验证错误)
10002	系统错误
10000	成功
10005	系统超时
10006	系统繁忙
10003	系统异常
10004	尚未开通此业务
10007	非法地址
SYSTEM_000-044	系统报错

### 3.2.3.比对服务接口返回结果说明

```
{
  "code": "10000",
  "message": "成功",
  "data": {
    "score": 0.9663585424423218
  },
  "seqNo": "34R952F61803161404"
}
```

score : 得分,值为 0~1 , 值越大表示是同一个人的可能性越大

Code 码如下：

返回状态码	说明
10014	身份证号无效
10001	系统异常
10015	照片不存在

10002	输入参数无效
10016	识别失败
10000	成功
10005	使用频率超过限制
10010	外部服务不可用
10006	图片大小不符合要求
10011	外部服务超时
10012	外部服务出错
10003	系统超时
10013	姓名与身份证号不匹配
10004	接口访问失败
10009	图片损坏
10007	输入内容长度不符合要求
10008	图片类型不符合要求
SYSTEM_000-044	系统报错

### 3.2.4.活体检测返回结果说明

返回状态码	说明
-1	正常检测运行
4	调用接口出错
5	授权文件校验不通过
6	model 文件校验不通过
17	未找到授权文件
18	授权文件绑定的包名出错
7	授权文件过期
8	License 授权文件跟 SDK 版本不匹配
10	platform 不支持
11	模型文件没找到
12	API KEY 不合法
13	模型过期
9	超时错误
14	服务器错误
21	检测为 HACK
15	参数配制出错/人脸状态错误
23	人脸遮挡
16	检测失败
19	网络超时
20	人脸远近参数设置无效
22	参数设置无效