

多人脸 SDK 接口使用说明

1. 版本说明

版本号	更新内容
V1.0.0	接口描述

2. 什么是 AAR 移植包

AAR 相对于 JAR 包，多包含 So、assets、res 等文件。

3. 屏蔽对 AAR 类的混淆(如果有开启混淆，非常重要)：

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

4. 什么授权和非授权版本

目前我司根据商务，提供两种版本供客户使用，授权版本可以永久使用，非授权版本有固定的使用期限，到期 SDK 将无法使用。

5. 基本 API 介绍

5.1 初始化 API

初始化结果通过初始化的返回值进行通知，具体如下：

```
Status status = FaceDetectManager.init(Context);
```

返回参数的含义：

AlreadyInitialized	已经初始化
Ok	初始化成功
AuthorizationVerifyFail	初始化失败（授权失败）
FaceManagerInitError	初始化失败（算法初始化失败）
FeatureInitError	初始化失败（特征库初始化失败）
DbInitError	初始化失败（数据库初始化失败）
SDKInitError	初始化失败（算法库初始化失败）

5.2 SDK 释放资源

在程序退出时，调用该 API 释放资源，具体如下：

```
FaceDetectManager.release();
```

5.3 设置旋转角度 API

在通过相机预览进行识别的时候，需要传入相机的旋转角度，保证相机的数据正常。

相关代码如下：

```
FaceDetectManager.setDegree(Degree);
```

传入参数的含义：

Degree	相机旋转角度
--------	--------

5.4 设置探测接口缩放比例 API

在对相机数据进行人脸探测的时候，适当压缩图片比例会加快探测的速度，但带来的反作用是，缩放比例越大，识别距离越近，所以可以根据对距离相应的需求，适当改变缩放比例。相关的代码如下：

```
FaceDetectManager.setScaleRate(Rate);
```

传入参数的含义：

Rate	缩放比例
------	------

参数区间为 0-1，默认为 1 不压缩

5.5 NV21 数据转 BitmapAPI

该接口主要提供一个将相机数据转换为 bitmap 的工具接口，方便使用。相关代码如下：

```
FaceDetectManager.yuvToBitmap(byte[],width,height );
```

传入参数的含义：

byte[]	相机 onPreviewFrame 的原始 byte 数组
width	预览宽
height	预览高

返回参数的含义：

返回转换成功的 bitmap。

5.6 NV21 数据转 Bitmap（含缩放） API

同上述 api，设置不同的缩放比例，图片大小随之缩放。相关代码如下：

```
FaceDetectManager.yuvToBitmapByScale(byte[],width,height , scale);
```

传入参数的含义：

byte[]	相机 onPreviewFrame 的原始 byte 数组
width	预览宽
height	预览高
Scale	缩放比例

返回参数的含义：

返回转换成功的 bitmap。

5.7 请求授权 API

授权版本需要先进行授权，才能使用 SDK，相关代码如下：

```
FaceDetectManager.networkAuthorization(new NetListener() {  
    @Override  
    public void onComplete() {  
  
    }  
    @Override  
    public void onError(Status status, final String s) {  
  
    }  
});
```

返回参数的含义：

AuthorizationDone	已经授权过
-------------------	-------

AuthorizationVerifyFail	授权失败
AuthorizationApplyError	请求授权异常

5.8 图片探测人脸 API

使用该 API 需要传入一张图片，人脸识别完成返回图片中包含的人脸信息。具体代码如下：

```
FaceDetectManager.detectFace(Bitmap, new DetectListener() {
    @Override
    public void onSuccess(List<DetectBean> list) {

    }

    @Override
    public void onError(Status status, String s) {

    }
});
```

传入参数的含义：

Bitmap	需要探测的图片
DetectListener	回调接口

回调成功的含义：

List<DetectBean>	人脸队列(队列长度代表人脸在图片中的个数，长度为0，为没有人脸)
------------------	----------------------------------

回调失败 Status 的含义：

NoInitialization	没有初始化
ParameterError	参数错误
DetectFaceError	探测人脸失败

5.9 相机数据探测人脸 API

使用该 API 需要传入相机的原始数据，人脸识别完成返回图片中包含的人脸信息。具体代码如下：

```
FaceDetectManager.detectFace(byte[],width,height, new DetectListener() {
    @Override
    public void onSuccess(List<DetectBean> list) {

    }

    @Override
    public void onError(Status status, String s) {

    }
});
```

传入参数的含义：

byte[]	相机 onPreviewFrame 的原始 byte 数组
width	预览宽
height	预览高
DetectListener	回调接口

回调成功的含义：

List<DetectBean>	人脸队列(队列长度代表人脸在相机帧中的个数，长度为 0，为没有人脸)
------------------	------------------------------------

回调失败 Status 的含义：

NoInitialization	没有初始化
ParameterError	参数错误
DetectFaceError	探测人脸失败

5.10 探测返回人脸介绍

下面是介绍探测返回的人脸队列，没个人脸的含义。

X0	人脸左上角 X 坐标
Y0	人脸左上角 Y 坐标
X1	人脸右下角 X 坐标
Y1	人脸右下角 Y 坐标
faceState	人脸状态
	OK(正常)
	UNNORMAL(异常)
	LEFT_DEVIATION(左偏)
	RIGHT_DEVIATION(右偏)
	BOW(低头)
	RISE(抬头)
Light	人脸亮度
Blur	人脸模糊度
landmarks	五官坐标

5.11 提取特征值 API

识别到了人脸，通过该 api 提取人脸的特征，用于人脸对比。具体代码如下：

```
float[] feature = FaceDetectManager.extractFeature(Bitmap,DetectBean);
```

传入参数的含义：

Bitmap	上述探测接口，探测的数据源
DetectBean	上述探测接口，探测到的人脸

返回值：

Float[]	提取到的特征值
Null	提取特征值失败

5.12 提取特征值(异步) API

该 api 同上述 api5.11 含义相同，只不过传入参数不同，上面的 api 需要传入检测到的人脸，该 api 会自己探测人脸，然后取最大人脸进行提取特征值，该 api 是异步的，所以说非实时的。具体代码如下：

```
FaceDetectManager.extractFeatureASync(Bitmap, new ExtractFeatureListener() {  
    @Override  
    public void onSuccess(DetectBean detectBean ,ArrayList<Float> floats) {  
  
    }  
    @Override  
    public void onError(Status code) {  
  
    }  
});
```

传入参数的含义：

Bitmap	上述探测接口，探测的数据源
ExtractFeatureListener	回调接口

成功回调值：

floats	提取到的特征值
detectBean	图片探测的人脸信息

失败回调值：

NoInitialization	没有初始化
ParameterError	参数错误
DetectNoFace	没有人脸
FeatureExtractFail	提取特征值失败
FeatureIsNull	提取特征值为空
DetectFaceError	探测人脸失败

5.13 对比特征值 API

将提取到的两个人的特征值进行对比，得到一个 0-1 的数据，就是相似度，相似度越高，越像同一个人。

当返回 score 值为 0 时，有可能是对比异常

具体代码如下：

```
float score = FaceDetectManager.compareFeature(float[],float[]);
```

5.14 图片注册 API

将通过探测接口探测到的人，添加名称进行注册，保存到数据库，方便下次识别。具体代码如下：

```
FaceDetectManager.registerUser(Bitmap, DetectBean, RegisterBean, new RegisterListener() {  
    @Override
```

```

        public void onSuccess(UserBean userBean) {

        }

        @Override
        public void onError(Status status) {

        }

    });

```

传入参数的含义：

Bitmap	探测的数据源
DetectBean	探测到的人脸
RegisterBean	注册人脸信息
RegisterListener	回调接口

成功回调值：

UserBean	注册成功的人脸信息
Userid	Id
Name	名称
localImagePath	保存头像的本地地址

onSimilarity 回调值：

UserBean	当注册开启过滤已经注册的人，并且失败的回调值为 HaveRegisteredThePerson, 此时回调的 UserBean 值为已经注册的人员
----------	---

失败回调值：

NoInitialization	没有初始化
ParameterError	参数错误
FaceRectIsNotCenterPic	人脸位置不在图片内
UpperLimitOfRegistration	注册人数已经达到数据库最大值
FaceBrightnessDarkOrLight	人脸太亮或者太暗
FaceTooBlur	人脸太模糊
NotRightFace	人脸状态错误，非正面脸
FeatureIsNull	特征值为空
HaveRegisteredThePerson	已经注册该人员
DbInsertFail	数据库插入失败

5.15 相机数据注册 API

相机数据注册，返回值同上述 5.14 一致。

具体代码如下：

FaceDetectManager.registerUser(byte[] ,width ,height, DetectBean, RegisterBean, RegisterListener);

传入参数的含义：

byte[]	相机 onPreviewFrame 的原始 byte 数组
width	预览宽
height	预览高
RegisterListener	回调接口

5.16 图片注册 API

使用图片注册，前提是已经对图片进行了探测和提取了图片的特征值，对应 5.12 接口。
具体代码如下：

```
FaceDetectManager.registerUser(bitmap,detectBean,registerBean,floats,
RegisterListener() {
    @Override
    public void onSuccess(UserBean userBean) {

    }
    @Override
    public void onError(Status status) {

    }
});
```

传入参数的含义：

bitmap	注册图片
detectBean	图片的人脸信息
registerBean	注册信息
floats	特征值
RegisterListener	回调接口

成功回调值：

UserBean	注册成功的人脸信息
Userid	Id
Name	名称
localImagePath	保存头像的本地地址

5.17 识别 API

通过对相机数据的连续帧进行处理，对比已经注册的人，完成整个识别流程。
具体代码如下：

```
FaceDetectManager.recognizeFace(byte[], w, h, new RecognizeListener() {
    @Override
    public void onDetectFace(List<FaceBean> faceBeanList) {

    }
    @Override
```

```

    public void onLiveness(LivenessBean livenessBean) {

    }

    @Override
    public void onRecognize(UserBean userBean) {

    }

    @Override
    public void onError(Status status) {

    }

    });

```

传入参数的含义：

byte[]	相机 onPreviewFrame 的原始 byte 数组
width	预览宽
height	预览高
RecognizeListener	回调接口

onDetectFace 回调的含义：

faceBeanList	识别到的人脸
DetectBean	人脸信息
LiveBean	人脸的活体信息(0 未能判断 1 活体 2 非活体)
UserBean	人脸信息

onLiveness 回调的含义：

livenessBean	识别到的人脸
DetectBean	人脸信息
LiveBean	人脸的活体信息(0 未能判断 1 活体 2 非活体)

onRecognize 回调的含义：

userBean	识别到的人的信息
Userid	人脸 id
Name	名称
headImage	人脸头像地址
features	特征值

onError 回调的含义：

NoInitialization	没有初始化
ParameterError	参数错误

5.18 查询数据库数量 API

查询已经注册的人脸数量，返回的 count 就是已经注册的人脸的数量。
具体代码如下：

```
long count = FaceDetectManager.queryCount();
```

5.19 查询所有数据库数据 API

查询已经注册的人脸数量，返回的 list 就是已经注册的所有人。

具体代码如下：

```
IList<UserBean> list = FaceDetectManager.queryAll();
```

5.20 通过 id 查询数据注册 API

通过指定 id 查询，返回的 list 就是查询结果。

具体代码如下：

```
List<UserBean> list = FaceDetectManager.queryById(String userid);
```

5.21 通过 name 查询数据注册 API

通过指定名称查询，返回的 list 就是查询结果。

具体代码如下：

```
List<UserBean> list = FaceDetectManager.queryByName(String name);
```

5.22 通过实体删除数据 API

删除指定注册的人，传入参数为要删除的人，返回的 result 是代表是否删除成功。

具体代码如下：

```
boolean result = FaceDetectManager.deleteByUserInfo(UserBean);
```

5.23 批量删除数据 API

批量删除指定注册的人，传入参数为要删除人的队列，返回的 result 是代表是否删除成功。

具体代码如下：

```
boolean result = FaceDetectManager.deleteByUserInfo(List< UserBean >);
```

5.24 删除所有数据库数据 API

删除所有已经注册的人，返回的 result 是代表是否删除成功。

具体代码如下：

```
boolean result = FaceDetectManager.deleteAll();
```

5.25 通过实体修改数据库数据 API

修改已经注册的人，返回的 result 是代表是否删除成功。

具体代码如下：

```
boolean result = FaceDetectManager.updateUser(UserBean bean);
```

5.26 配置

SDK 提供了一些配置信息，在代码中可以通过 ConfigLib 类，取到 public 的常量值如下表：

名称	含义	默认值
degree	相机旋转角度	0
detectWithLiveness	识别接口是否进行活体探测	False

detectWithRecognition	识别接口是否进行识别	True
dbUserIDUseDefault	不自定义 userid, 因为 userid 必须要保证唯一	True
Register_Check_Brightness	注册接口是否对人脸进行光线判断	True
Register_Check_Blur	注册接口是否对人脸进行模糊度判断	True
Register_Check_FaceState	注册接口是否对人脸进行人脸角度判断	True
similarityFilter	注册接口是否过滤相似的人判断	True
saveHeadPic	注册接口是否保存本地头像判断	True
maxRegisterBrightness	注册人脸最大亮度值	230
minRegisterBrightness	注册人脸最小亮度值	40
blurRegisterThreshold	注册人脸最小模糊度值	0.65(越小越清晰)
maxRecognizeBrightness	识别人脸最大亮度值	230
minRecognizeBrightness	识别人脸最小亮度值	40
blurRecognizeThreshold	识别人脸最小模糊度值	15(越大越清晰)
livenessThreshold	活体阈值	0.78
featureThreshold	人脸对比阈值	0.825
maxShowDetectNum	单帧最大识别人脸个数	10
picScaleRate	识别缩放比例	1