

# Android端opensdk唇动集成说明文档

---

## 一、Android studio配置工程

## 二、SDK调用步骤，功能实现

### 唇动API使用

#### 1 创建算法实例

算法实例化接口：

参数说明：

返回值：

具体代码示例如下：

#### 2 单个能力license鉴权

接口描述：

license证书验签接口：

参数说明(获取路径方式可参考demo)：

返回值：

具体代码示例如下：

#### 3 传入对应唇动能力的model模型文件

接口描述：

传入文件接口

参数说明(获取路径方式可参考demo)：

返回值：

具体代码示例如下：

#### 4 唇动及结果返回（重要）

接口描述：

接口示例：

参数说明：

返回值：

调用样例代码如下：

接口描述（用于处理本地图片传入）：

接口示例：

参数说明:

返回值:

调用样例代码如下:

5 销毁算法。

接口描述:

接口示例:

参数说明:

返回值:

具体代码示例如下:

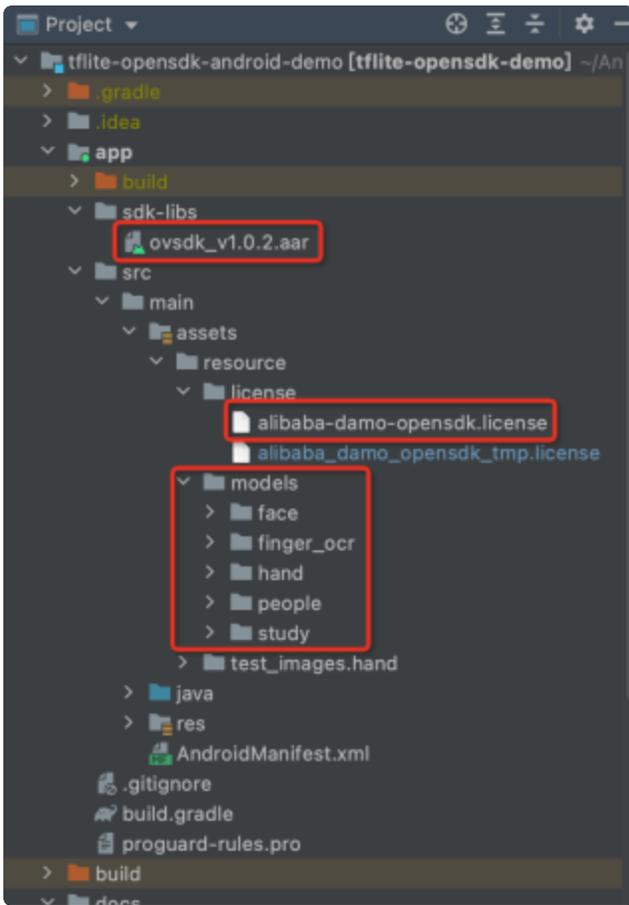
三、支持的系统和硬件版本

四、注意事项

附 离线鉴权错误码定义

## 一、Android studio配置工程

1、获取相关资源压缩包（由官网线上购买申请或阿里云相关人员提供下载链接）后，可看到如下资源文件，demo示例工程、支持相关能力的aar及支持相关能力的license文件。如下图：



注意：alibaba-damo-opensdk.license为正式证书（官网下载获取的都是正式证书），\_tmp结尾的为临时证书一般线下提供，临时证书不能改名，正式license可以改名字，但是不能与tmp license重名。两个证书只需要调用其中一个进行鉴权就可以。

## 二、SDK调用步骤，功能实现

### 唇动API使用

#### 1 创建算法实例

OVLipDist：在需要用到唇动算法的地方，创建 OVLipDist 唇动实例，OVLipDist 是唇动API接口对象，通过此对象可以完成 camera 实时视频流、Video视频流、图片等唇动的能力的使用。

算法实例化接口：

```
Plain Text | 复制代码  
1 OVLipDist mOVLipDist = new OVLipDist();
```

### 参数说明:

无

### 返回值:

无

### 具体代码示例如下:

```
▼ Plain Text | 复制代码  
1 private OVLipDist mOVLipDist;  
2 OVLipDist = new OVLipDist();
```

## 2 单个能力license鉴权

### 接口描述:

license证书验签接口，验签通过后才能成功调用算法。

### license证书验签接口:

```
▼ Plain Text | 复制代码  
1 mOVLipDist.nativeCheckLicense(String licensePath);
```

### 参数说明(获取路径方式可参考demo):

String licensePath 传入license路径。

注：需要传入单独的证书文件的绝对路径

### 返回值:

int类型，返回0为验签成功，其它返回为验签失败。

### 具体代码示例如下:

```
1 int status = m0VLipDist.nativeCheckLicense(licensePath);
```

### 3 传入对应唇动能力的model模型文件

#### 接口描述:

传入算法识别所需的model模型文件。

#### 传入文件接口

```
1 m0VLipDist.nativeLipDistCreateHandle(det, landmark, valid);
```

#### 参数说明(获取路径方式可参考demo):

String det 传入算法的模型文件路径, 需具体到文件名, 传绝对路径。

String landmark 传入算法的模型文件路径, 需具体到文件名, 传绝对路径。

String valid 传入算法的模型文件路径, 需具体到文件名, 传绝对路径。

#### 返回值:

int类型, 返回0为验签成功, 其它返回为验签失败。

#### 具体代码示例如下:

```
1 int status = m0VLipDist.nativeLipDistCreateHandle(det, landmark, valid);
```

### 4 唇动及结果返回 (重要)

#### 接口描述:

该方法为处理目标唇动及结果返回的接口。

## 接口示例:

```
Plain Text | 复制代码  
1 mmOVLipDist.nativeStartLipDistFromBuffer(ByteBuffer outBuffer, int width,  
2         int height, int step,int angle,int formatType, OVLipDistRes  
    ult detectResult);
```

## 参数说明:

- outBuffer: 传入算法的outBuffer数组数据。
- width: 传入数据的宽。
- height: 传入数据的高。
- step: 算法的步长, 例RGBA为4通道, 步长\*4。
- angle: 传给算法的角度, 该角度为帧图转正需要的角度, 如图是正向传入的则传值为0。
- formatType: 传入outBuffer的数据格式, 如buffer传入的是Videobuffer数据, 则传入OVImageFormat.OV\_IMG\_FMT\_RGBA.formatType。
- detectResult: 唇动结果返回。需唇动结果实体 OVLipDistResult detectResult = new OVLipDistResult(),作为参数传进去。返回结果 detectResult.getObjects()[i].getLipdist() 为唇动结果, 表示上下唇距离, 具体示例可参考demo获取方法。

## 返回值:

int类型, 返回0为删除算法相关成功, 其它返回为重置失败。

## 调用样例代码如下:

```
int status = mOVLipDist.nativeStartLipDistFromBuffer(outBuffer, size.getWidth(), size.getHeight(),  
size.getWidth() * 4, 0, OVImageFormat.OV_IMG_FMT_RGBA.formatType, detectResult);
```

## 接口描述 (用于处理本地图片传入) :

该方法为唇动的接口, 传入图片的路径, 获得唇动结果的返回数据。

## 接口示例:

```
1 mOVPeopleDetection.nativeStartLipDist(String imagePath, OVLipDistResult detectResult);
```

### 参数说明:

- imagePath: 传入算法的图片路径地址。
- detectResult: 唇动结果返回。需坐姿测结果实体 OVLipDistResult detectResult = new OVLipDistResult(), 作为参数传进去。返回结果 detectResult.getObjects()[i].getLipdist() 为唇动结果, 表示上下唇距离, 具体示例可参考demo获取方法。

### 返回值:

int类型, 返回0为打分处理成功, 其它返回为处理失败。

### 调用样例代码如下:

```
int status = mOVPeopleDetection.nativeStartLipDist(imagePath, detectResult);
```

注意: 算法内部没有对内存进行处理, 输出buffer需提前申请内存空间, 初始化格式为:

```
1 outBuffer = ByteBuffer.allocateDirect(textureWidth * textureHeight * 4);
```

## 5 销毁算法。

### 接口描述:

在销毁算法时, 调用该接口。

### 接口示例:

```
1 mOVLipDist.nativeLipDistDestroy();
```

### 参数说明:

无

### 返回值：

int类型，返回0为删除算法相关成功，其它返回为销毁算法失败。

### 具体代码示例如下：

```
Plain Text | 复制代码  
1 int status = m0VLipDist.nativeLipDistDestroy();
```

## 三、支持的系统和硬件版本

- 1、硬件要求：要求设备上有相机模块,陀螺仪模块
- 2、CPU架构：armeabi-v7a、arm64-v8a
- 3、系统：最低支持 Android 4.0（API Level 14）需要开发者通过minSdkVersion来保证支持系统的检测

## 四、注意事项

- 1、viapi-android-sdk的 minSdkVersion为 14。
- 2、demo工程Android Studio 3.4 及以上，Open GLES 2.0 及以上。
- 3、证书分为临时证书以\_tmp结尾和正式证书。临时证书一般作为调试用，正式证书一般作为发布用。根据证书的不同代码需要做相应调整，全局搜索mIsTmpLicense字段，如果是\_tmp证书，需把该字段改为true，正式证书的话把该字段改为false。

## 附 离线鉴权错误码定义

- 20011 license没有初始化直接调用API接口。
- 20012 当前的license与调用app不是绑定关系，license用在其他app中使用。
- 20013 license无效。
- 20014 license授权时间过期。
- 20015 此license中不包含调用的算法能力（未购买此能力）。
- 20016 bundle id获取失败。

-20017 临时license时间校验失败。