

# Android端opensdk图像/本地视频增强集成说明文档

---

## 一、Android studio配置工程

## 二、SDK调用步骤，功能实现

### 2.1 SDK初始化

接口描述：

初始化接口：

参数说明：

返回值：

具体代码示例如下：

### 2.2 图像/本地视频增强API使用

#### 2.2.1 创建算法实例

图像/本地视频增强接口描述：

算法实例化接口：

#### 2.2.2 单个能力license鉴权

接口描述：

license证书验签接口：

获取证书路径(具体实现可以参考demo)：

参数说明：

返回值：

具体代码示例如下：

#### 2.2.3 创建算法加载模型

接口描述：

接口示例：

参数说明：

返回值：

具体代码示例如下：

#### 2.2.4 获取增强后的图片/视频帧的尺寸信息

接口描述：

接口示例:

参数说明:

返回值:

具体代码示例如下:

#### 2.2.5 图片/视频帧数据增强处理

接口描述:

接口示例:

参数说明:

返回值:

具体代码示例如下:

#### 2.2.6 算法销毁Destroy

接口描述:

接口示例:

参数说明:

返回值:

具体代码示例如下:

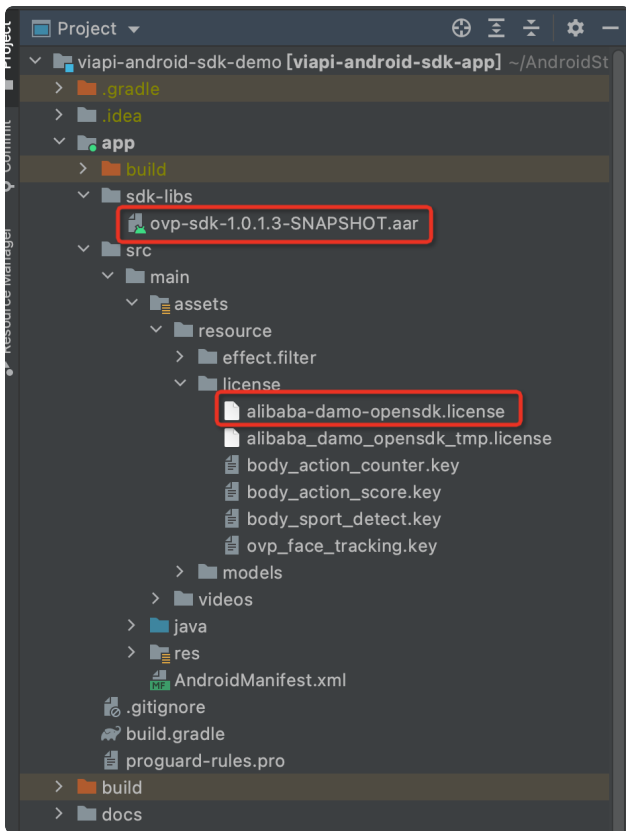
### 三、支持的系统和硬件版本

### 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，解压压缩包，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

### 2.1 SDK初始化

#### 接口描述：

算法API使用前先调用SDK初始化接口，初始化之后，各功能才可以正常使用，否则会引起鉴权等异常，初始化建议放在app进程启动时Application onCreate中进行。

#### 初始化接口：

```
1 VIAPICreateApi.getInstance().getVIAPISdkCore().init(Context context,boolean isDebug);
```

### 参数说明：

Context context 应用上下文。

boolean isDebug SDK调试开关。

### 返回值：

int类型，返回0为初始化成功，其它返回为初始化失败。

### 具体代码示例如下：

```
▼ Plain Text | 复制代码
1 private void initSDK() {
2     int status = VIAPICreateApi.getInstance().getVIAPISdkCore().init(this, false);
3     if (status != 0) {
4         Toast.makeText(this, VIAPISdkCore.getErrorMsg(status), Toast.LENGTH_LONG).show();
5     } else {
6         Toast.makeText(this, "初始化成功!", Toast.LENGTH_LONG).show();
7     }
8 }
```

## 2.2 图像/本地视频增强API使用

### 2.2.1 创建算法实例

#### 图像/本地视频增强接口描述：

OVIImageEnhance：在需要用到图像/本地视频增强算法的地方，创建OVIImageEnhance实例，OVIImageEnhance是图像/本地视频增强API接口对象，通过此对象可以完成图像/本地视频增强能力的使用。

#### 算法实例化接口：

```
▼ Plain Text | 复制代码
1 OVIImageEnhance mImageEnhance = new OVIImageEnhance();
```

## 2.2.2 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 mImageEnhance.nativeCheckLicense(String licensePath);
```

### 获取证书路径(具体实现可以参考demo)：

▼ Plain Text | 复制代码

```
1 String licensePath = AssetsProvider.getReleaseLicenseFilePath();
```

### 参数说明：

- String licensePath： 传入全局证书license路径或自定义的单个能力license路径。

注：如果所有能力使用同一个全局证书默认传入全局证书路径即可，如果接入方有针对此能力的单独证书，则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：

▼ Plain Text | 复制代码

```
1 int errorCode = mImageEnhance.nativeCheckLicense(licensePath);
```

## 2.2.3 创建算法加载模型

### 接口描述：

创建算法内部用于图像/本地视频增强的实例对象。

### 接口示例：

`OVIImageEnhance.nativeCreateHandle(String modelsPath,String modelsRootPath,int width,int height,int type);`

### 参数说明：

- String modelsPath：为算法模型文件系统的绝对路径。
- String modelsRootPath：为存放资源的目录，资源名字不能修改。
- int width:视频帧的宽
- int height：视频帧的高
- int type：2图片超分 3视频超分

### 返回值：

int类型，返回0为算法初始化成功，其它返回为算法初始化失败。

### 具体代码示例如下：

```
1  int errorCode = mImageEnhance.nativeCreateHandle(modelsPath,modelsRootPath,640, 360, 3)
```

## 2.2.4 获取增强后的图片/视频帧的尺寸信息

### 接口描述：

获取增强后的图片/视频帧的尺寸信息，用于5.6.5创建接收算法增强返回数据的ByteBuffer对象。

### 接口示例：

`OVIImageEnhance.nativeGetOutSize(int width, int height, OVSize ovSize);`

### 参数说明：

- int width：传入图片/视频帧的宽度。
- int height：传入图片/视频帧的高度。
- OVSize ovSize：传入OVSize对象，用于接收算法返回增强后的图片/视频帧的尺寸信息。

### 返回值：

int类型，返回0为获取增强后的图片/视频帧的尺寸信息成功，其它返回为获取增强后的图片/视频帧的尺寸信息失败。

### 具体代码示例如下：

▼ Plain Text | 复制代码

```
1  int errorCode = mImageEnhance.nativeGetOutSize(bitmap.getWidth(), bitmap.getHeight(), ovSize);
```

## 2.2.5 图片/视频帧数据增强处理

### 接口描述：

对传入的图片/视频帧数据进行增强处理，并返回处理结果。

### 接口示例：

```
OVLImageEnhance.nativeProcessBuffer(ByteBuffer byteBuffer, int width, int height, int stride, int formatType, ByteBuffer outBuffer);
```

### 参数说明：

- ByteBuffer byteBuffer：传入图片/视频帧的ByteBuffer数据。
- int width：传入图片/视频帧的宽。
- int height：传入图片/视频帧的高。
- int stride：传入图片/视频帧的步进值。
- int formatType：传入图片/视频帧的编码格式。
- ByteBuffer outBuffer：传入ByteBuffer对象（空间为5.6.4获取到的ovSize尺寸所需空间），用于接收算法处理后的RGBA格式的数据信息。

### 返回值：

int类型，返回0为图片/视频帧数据增强处理成功，其它返回为图片/视频帧数据增强处理失败。

### 具体代码示例如下：

```
1 int errorCode = mImageEnhance.nativeProcessBuffer(originalBuffer, bitmap.getWidth(), bitmap.getHeight(), bitmap.getWidth() * 4, OVPImageFormat.OVP_IMG_FMT_RGBA.formatType, outBuffer);
```

## 2.2.6 算法销毁Destroy

### 接口描述：

在不需要用到算法的时候，对算法对象的销毁内存释放操作。

### 接口示例：

```
ImageEnhance.nativeSegmentDestroy();
```

### 参数说明：

无

### 返回值：

int类型，返回0为销毁算法相关成功，其它返回为销毁算法相关处理失败。

### 具体代码示例如下：

```
1 mImageEnhance.nativeDestroyHandle();
```

## 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mlsTmpLicense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

- 2011 license没有初始化直接调用API接口。
- 2012 当前的license与调用app不是绑定关系, license用在其他app中使用。
- 2013 license无效。
- 2014 license授权时间过期。
- 2015 此license中不包含调用的算法能力 (未购买此能力。
- 2016 bundle id获取失败。
- 2017 临时license时间校验失败。