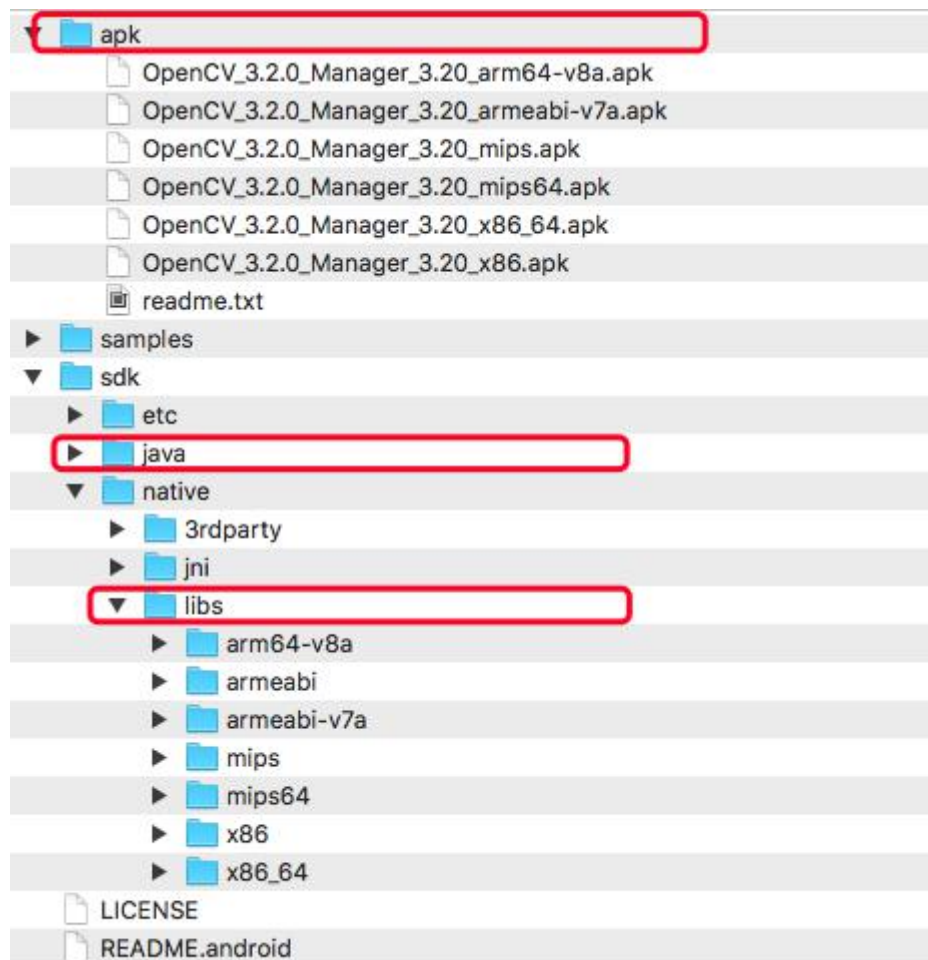


# Android For OpenCV 的环境搭建

OpenCV 是一个基于 BSD 许可（开源）发行的跨平台计算机视觉库，可以运行在 Linux、Windows、Android 和 Mac OS 操作系统上。它轻量级而且高效——由一系列 C 函数和少量 C++ 类，同时提供了 Python、Ruby、MATLAB 等语言的接口，实现了和计算机视觉方面的很多通用算法[百度百科 词条:opencv](#)

用到东西如下：

- Android Studio 2.3
- OpenCV for Android
- 将压缩包解压后可以看到有以下文件，其中使用红线包裹的文件夹是非常重要的文件夹，后面要用到



- apk 保存的是一些基础包，要根据你的手机芯片的架构安装合适的版本,否则将会提醒你安装，以至于各种错误，所以在之前，还是老老实实安装吧。
- samples 保存的一些 demo，可以根据自己的需要进行学习
- 

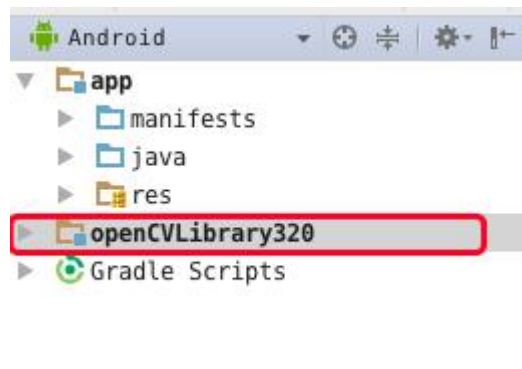
sdk 我们开发所用的一些工具

- 注意:我在学习的过程,直接在手机安装会报解析安装包错误,以至于无法安装,那么这里需要连接开发者模式,使用 adb 进行安装,命令: `adb install ****.apk` 就可以安装成功了。

## 创建工程

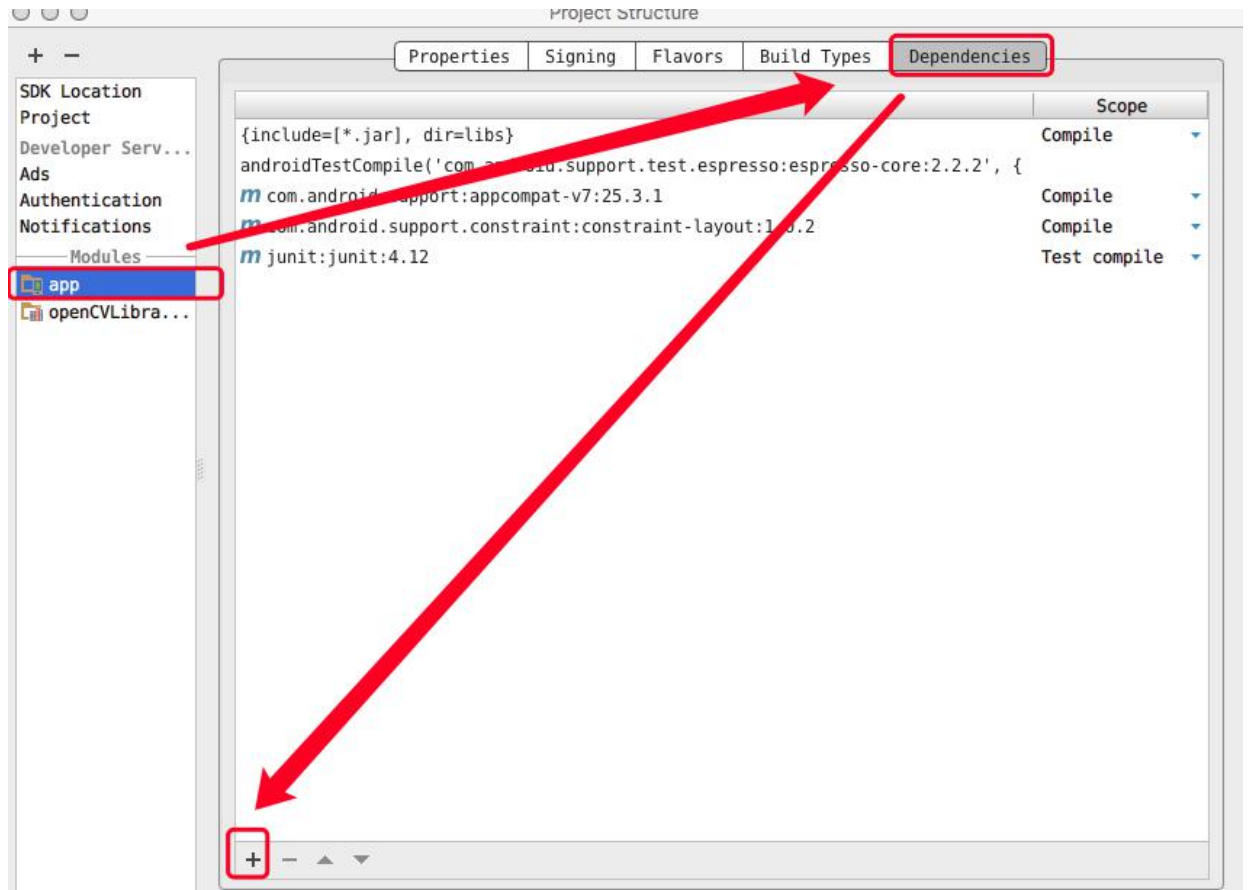
创建一个普通的安卓工程即可,然后我们需要导入 **Module**

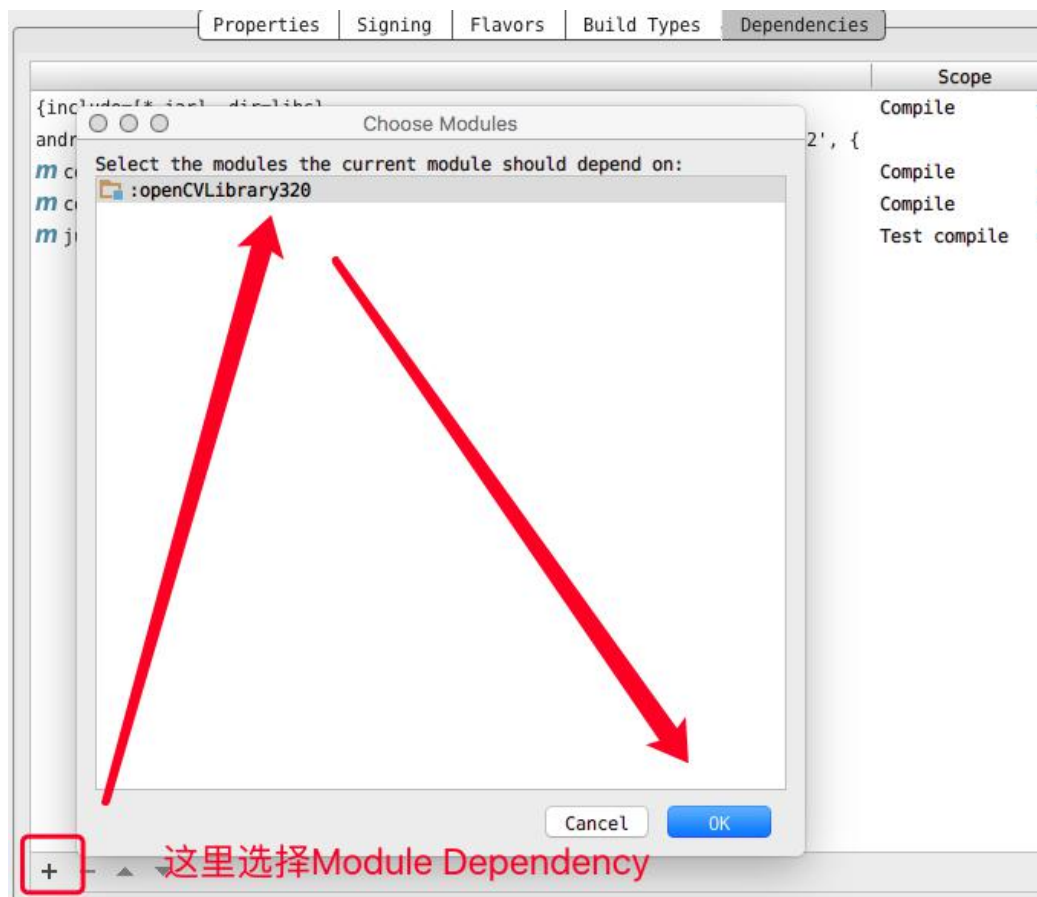
导入 **Module** 这个东西就是上图中的 **java** 文件夹,选择 **File--> new --> import Module** 选择步骤一图中红色框框圈起来的 **JAVA** 文件夹即可,然后在文件栏中就可以看到这个新的 **Module** 了,后面的 **320** 指的是版本号,每个人可能不一样



- 
- 

导入 **Module** 之后我们需要配置一下,在 **File--->Project Structure** 菜单中选择该 **Moudle** 作为库依赖即可,如下操作





注意：在某些情况下，再倒入 Module 的情况下，会出现报错的情况，这是修改 Module 的编译版本和目标版本和 APP 的保持一致，然后重新 Sync 即可。

## 导入库文件

- 这里需要导入 so 库等文件，文件就在步骤一图的用红线圈中的 libs 文件夹里面，选中里面的所有文件，注意是选中 lib 里面所有的文件，不是选中 libs 文件夹。

- 

在项目中，选 app 工程，在 src/main 目录下新建 jniLibs 文件夹，将上面选中的文件全部复制到其中。

- 
- 

为了确保工程创建成功，我们可以重新 reBuild 一下工程

- 

## 载入 Opencv

- 我们在 **Activity** 的回调函数中异步初始化 **OpenCV**，代码很简单，一句话加上一个回调。

```
@Override

protected void onResume() {

    super.onResume();

    //载入 OpenCv 初始化引擎，并载入库，可以看到 initAsync，使用的是异步加载

    //因此需要提供回调函数

    //OpenCVLoader.OPENCV_VERSION_3_2_0 指的是版本号，请根据你的版本选择

    OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_3_2_0, getApplicationContext(),
mLoaderCallback);

}

//OpenCV 库加载并初始化成功后的回调函数

private BaseLoaderCallback mLoaderCallback = new BaseLoaderCallback(this) {

    @Override

    public void onManagerConnected(int status) {

        // TODO Auto-generated method stub

        switch (status) {

            case BaseLoaderCallback.SUCCESS:

                Log.e("OpenCv", "成功加载");

                break;

            default:

                super.onManagerConnected(status);
```

```
Log.e("OpenCv","加载失败");
```

```
break;
```

```
}
```

```
}
```

```
};
```

## 使用 OpenCV

- 我们来使用 OpenCv 做一个简单灰度图
- 首先是布局文件,就一个简单的图片

```
<?xml version="1.0" encoding="utf-8"?><RelativeLayout  
xmlns:android="http://schemas.android.com/apk/res/android"
```

```
    android:layout_width="match_parent"
```

```
    android:layout_height="match_parent">
```

```
    <ImageView
```

```
        android:id="@+id/img"
```

```
        android:layout_width="600px"
```

```
        android:layout_height="1000px"
```

```
        android:layout_centerHorizontal="true" />
```

```
    </RelativeLayout>
```

- 具体代码如下

```
Mat rgbMat = new Mat();
```

```
Mat grayMat = new Mat();
```

```
srcBitmap = BitmapFactory.decodeResource(getResources(), R.drawable.psb);
```

`//构建灰度 RGB 图片`

```
grayBitmap = Bitmap.createBitmap(srcBitmap.getWidth(), srcBitmap.getHeight(),  
Bitmap.Config.RGB_565);
```

`//将资源图片转化为 Mat 类资源`

```
Utils.bitmapToMat(srcBitmap, rgbMat); //convert original bitmap to Mat, R G B.
```

`//将图片灰度化`

```
Imgproc.cvtColor(rgbMat, grayMat, Imgproc.COLOR_RGB2GRAY);
```

`//将灰度 Mat 转换为灰度 Bitmap 资源`

```
Utils.matToBitmap(grayMat, grayBitmap);
```

`//设置显示`

```
((ImageView) findViewById(R.id.img)).setImageBitmap(binaryBitmap);
```