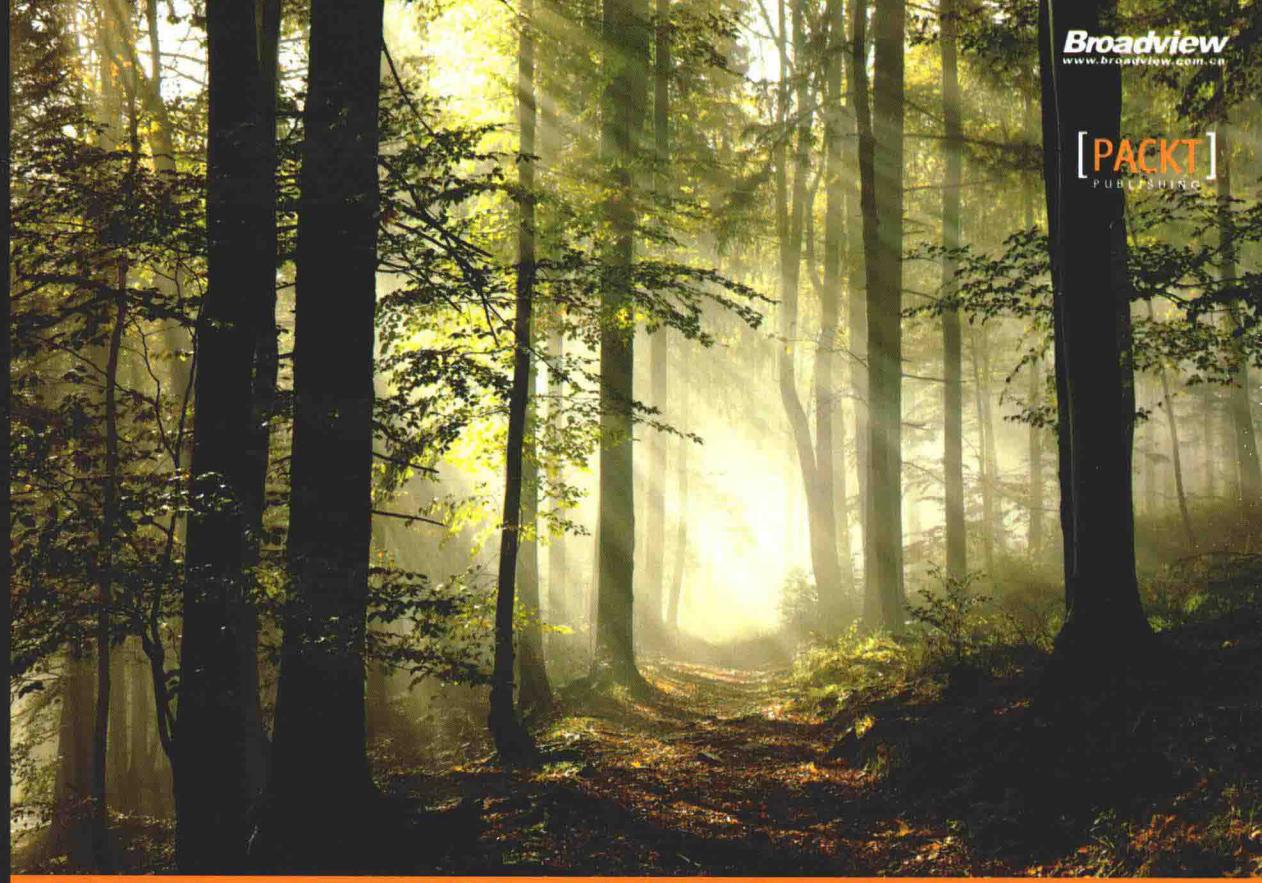


Mastering OpenCV Android Application Programming

深入OpenCV Android应用开发

掌握在Android平台实现计算机视觉算法的艺术，开发稳健、高效的应用

[印] Salil Kapur　Nisarg Thakkar　著
岳翰　译

中国工信出版集团
電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

Mastering OpenCV Android Application Programming

深入OpenCV Android应用开发

[印] Salil Kapur Nisarg Thakkar 著
岳翰 译

電子工業出版社
Publishing House of Electronics Industry
北京•BEIJING

内 容 简 介

本书以在 Android 平台上开发 OpenCV 应用为重点，详细介绍了计算机视觉技术的理论及其在移动平台的应用。本书由浅入深，囊括了从基本的开发环境部署，到基础的图像处理算法，再到目标检测、人脸检测、目标追踪、图像拼接等高级图像分析技术，以及用于图像分类的机器学习算法等各方面的知识。

本书虽然篇幅不多，但内容十分丰富，从理论到实践，从精辟的数学公式到翔实的源代码，从系统的算法解释到实用的编程技巧，完全能够满足读者从入门到进阶的求知需要。本书适合于有一定 Java 和 Android 开发基础，并对计算机视觉技术感兴趣的入门读者，亦可作为从事 Android 图像编程的开发人员，以及熟悉 OpenCV 开发并有意一试身手的编程爱好者的参考手册。

版权贸易合同登记号　图字：01-2015-8410

图书在版编目（CIP）数据

深入 OpenCV Android 应用开发 /（印）卡普尔，（印）塔卡尔著；岳翰译. —北京：电子工业出版社，2016.6
书名原文：Mastering OpenCV Android Application Programming
ISBN 978-7-121-28823-4

Ⅰ. ①深… Ⅱ. ①卡… ②塔… ③岳… Ⅲ. ①移动终端－应用程序－程序设计 Ⅳ. ①TN929.53

中国版本图书馆 CIP 数据核字(2016)第 103928 号

策划编辑：张春雨
责任编辑：葛　娜
印　　刷：三河市双峰印刷装订有限公司
装　　订：三河市双峰印刷装订有限公司
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编：100036
开　　本：787×980　1/16　印张：12.5　字数：257 千字
版　　次：2016 年 6 月第 1 版
印　　次：2016 年 6 月第 1 次印刷
定　　价：58.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888，88258888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

本书咨询联系方式：010-51260888-819　faq@phei.com.cn。

目录

1

为图像添加效果

在通常情况下，一幅图像所包含的信息量要多于任意一个特定任务所需要的。因此，我们要对图像进行预处理，以使图像包含的信息刚好满足应用的需要，从而缩减计算时间。

在本章中，我们会了解多种不同的预处理操作方式，包括：

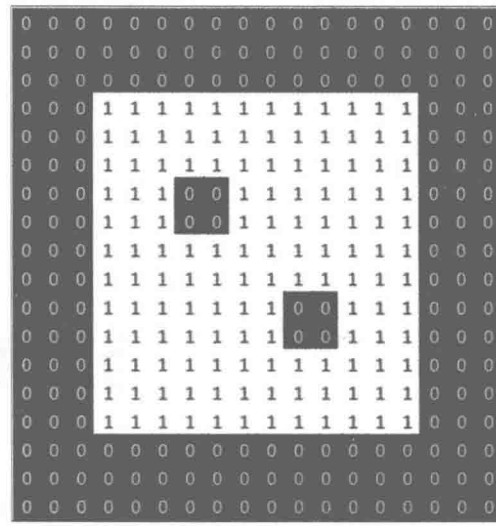
- 模糊
- 降噪
- 锐化
- 腐蚀和膨胀
- 阈值处理和自适应阈值

在本章末尾，我们会了解如何将 OpenCV 整合到现有的 Android 应用中。

在学习各种特征检测算法及其实现之前，我们先建立一个基础的 Android 应用，在本章的学习过程中会不断地将特征检测算法添加到这个应用中。

入门

当我们看到一幅图像时，我们感知的是颜色和物体。然而，对计算机视觉系统来说，它看到的是一个充满数字的矩阵（见下图）。依据所选颜色模型的差异，数字的含义也会有所不同。计算机无法直接从图像中提取模式或目标，计算机视觉系统的目的在于将这个数字构成的矩阵解释成某个特定类型的目标。



二值图像的表示

部署 OpenCV

OpenCV[1]是 Open Source Computer Vision library（开源的计算机视觉库）的缩写。它是使用最广泛的计算机视觉库。OpenCV 是计算机视觉领域常用的操作函数的集合，其自身由 C/C++编写而成，同时也提供了对 Python、Java 以及任意 JVM 语言的封装（JVM 语言指的是用于产生 Java 字节码的语言，比如 Scala 和 Clojure）。考虑到大部分 Android 应用是用 C++/Java 编写的，OpenCV 也被移植为供开发者使用的 SDK，以使他们开发的应用支持机器视觉。

现在，我们来看一下如何为 Android 平台部署 OpenCV，以便开启我们的旅程。我们选择 Android Studio[2]作为 IDE，不过任意一个 IDE 经过些许的调整也会胜任。执行下面的步骤：

1 本书代码使用 OpenCV 2.4.10。截至译文截稿前，OpenCV 2 的最新稳定版本为 2.4.11，同时 OpenCV 3 也已经迭代到了 3.1 版本。由于 OpenCV 3.x 版本相对 OpenCV 2.4.x 改动较大，出于学习的目的，推荐采用 2.4.x 版本。——译注

2 虽然 Eclipse 配合 ADT 插件也能进行 Android 开发，但谷歌官方已经不推荐采用这种开发流程，推荐开发者使用 Android Studio。截至译文截稿前，Android Studio 的最新稳定版本为 2.0.0，相比书中采用的 1.x 版本，其编译速度与 Android 模拟器的运行速度都有很大的提升。——译注

1. 从 https://developer.android.com/sdk/ 下载 Android Studio，从 http://sourceforge.net/projects/opencvlibrary/files/opencv-android/下载 OpenCV4Android SDK。

2. 把两个文件解压缩到已知的位置。

3. 创建一个一般的 Android 项目，命名为 FirstOpenCVApp。单击 **File | Import**。

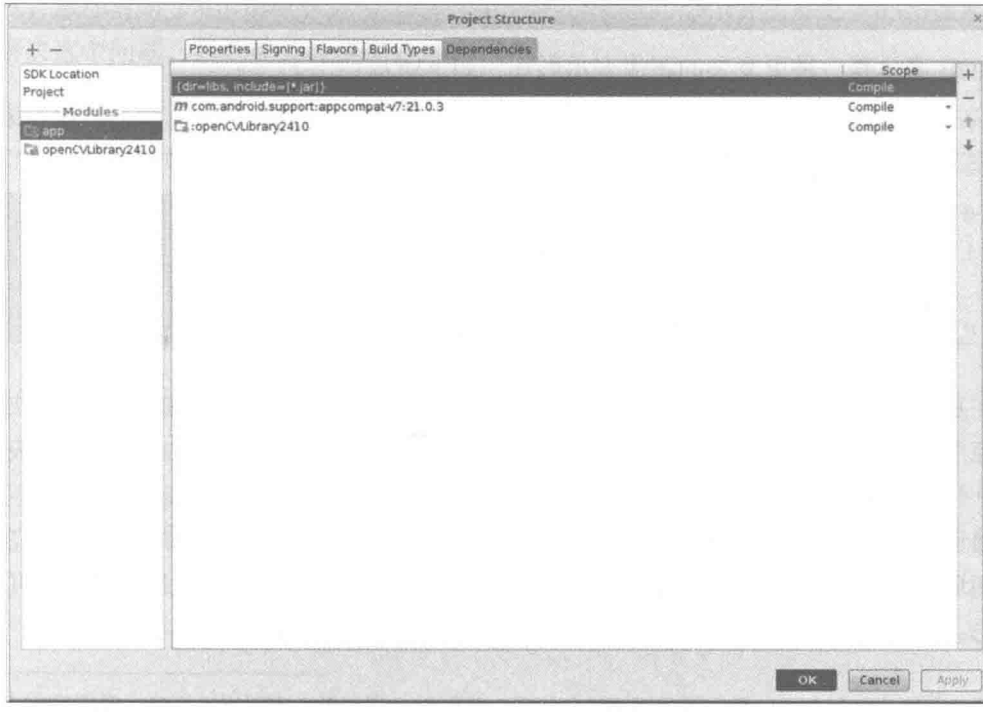
4. 选择 OpenCV4Android SDK 目录下的 sdk/java/文件夹。

5. 单击 **Build | Rebuild Project**。

6. 单击 **File | Project Structure**。

7. 把 OpenCV 模块加入到应用中，方法是在左侧栏中选择 **app** 模块，然后单击 **Dependencies** 选项卡中的绿色加号，最后选择 OpenCV 模块。

8. 现在，你已经准备好在 Android 项目里使用 OpenCV 了。界面应该是这个样子：



在 OpenCV 中存储图像

OpenCV 使用称为 **Mat** 的自定义对象存储图像，该对象保存了行数、列数、数据等能唯一标识该图像的信息，并能在需要的时候重新创建图像。不同的图像所包含的信息量也不同。例如，彩色图像包含的信息就比相同图像的灰度版本要多，这是因为一幅采用 RGB 颜色模型的彩色图像是三通道的，而灰度图像是单通道的。下面的图表展示了单通道和多通道（这里为 RGB）图像是如何存储的（图表摘自 `docs.opencv.org`）。

某图像的一种单通道表示如下：

图像的灰度（单通道）表示

	列 0	列 1	列…	列 *m*
行 0	0,0	0,1	0,…	0,*m*
行 1	1,0	1,1	…,…	1,*m*
行…	…,0	…,1	…,…	…,*m*
行 *n*	*n*,0	*n*,1	*n*,…	*n*,*m*

RGB 表示是一种更复杂的图像表示形式，如下所示：

图像的 RGB（三通道）表示

	列 0			列 1			列…			列 *m*		
行 0	0,0	0,0	0,0	0,1	0,1	0,1	0,…	0,…	0,…	0,*m*	0,*m*	0,*m*
行 1	1,0	1,0	1,0	1,1	1,1	1,1	…,…	…,…	…,…	1,m	1,*m*	1,*m*
行…	…,0	…,0	…,0	…,1	…,1	…,1	…,…	…,…	…,…	…,*m*	…,*m*	…,*m*
行 *n*	*n*,0	*n*,0	*n*,0	*n*,1	*n*,1	*n*,1	*n*,…	*n*,…	*n*,…	*n*,*m*	*n*,*m*	*n*,*m*

在灰度图像中，数字代表指定颜色的亮度。当以整数表示时，其范围为 0~255，其中 0 代表纯黑色，255 代表纯白色。如果用浮点数来表示，那么像素就以 0~1 之间的数值表示，其中 0 代表纯黑色，1 代表纯白色。在 OpenCV 的 RGB 图像中，第一个通道代表蓝色，第二个通道代表绿色，第三个通道代表红色。这样，每个通道都能代表任何特定颜色的亮度。我们知道，红、绿、蓝是三原色，它们按照不同的比例混合可以构成任意一种人眼可感知的颜色。下图展示了不同颜色及其对应整数形式的 RGB 等价表示：

(a) R=0 G=0 B=0

(b) R=10 G=20 B=100

(d) R=255 G=255 B=0

(c) R=255 G=0 B=0

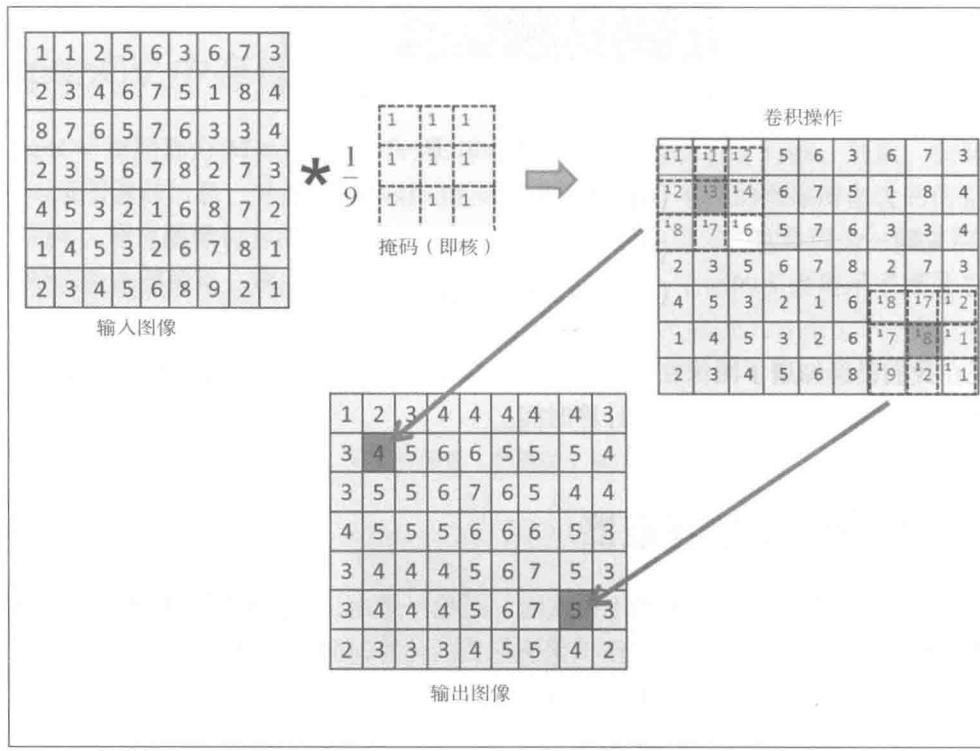
现在我们已经知道了图像在计算机中是怎样表示的，接下来我们要了解怎样调整像素值，使得在实际工作中花费更少的计算时间。

OpenCV 中的线性滤波器

我们都喜欢清晰的图像——谁不喜欢呢，是吧？然而，还需要进行权衡。图像的信息越丰富，意味着对于相同的操作，该图像所耗费的计算时间比包含信息更少的图像要长。为了解决这个问题，我们要进行模糊处理。

很多线性滤波算法都利用了称为**核**（kernel）的数字向量。核可以看作是沿着像素滑动的窗口，并把计算结果输出给该像素。通过下图可以更清晰地理解这个概念（这张表示线性滤波或卷积[3]的图片来自 `http://test.virtual-labs.ac.in/labs/cse19/neigh/convolution.jpg`）：

3 对于数字图像，卷积与线性滤波的计算过程相似。可以这样理解卷积过程：将核经过上下及左右两个方向的翻转，再做线性滤波计算。——译注



在上面的图中，一个 3×3 的核作用在了一幅 10×10 的图像上。

卷积是用于线性滤波的最常见操作之一，卷积核中的数值是对应像素做乘法运算的系数。最终结果保存在锚点——通常是卷积核的中心点中[4]。公式中的 src 代表原始图像，dst 代表目标图像，后文亦如此。

$$\mathrm{dst}(x,y)=\frac{\sum[\mathrm{src}(x+i,y+i)*\mathrm{kernel}(A+i,B+j)]}{\sum[\mathrm{kernel}(A+i,B+j)]}$$

其中，A,B 是锚点像素的位置；i,j 的范围取决于锚点像素。

线性滤波操作通常并不是就地操作，因为对每一个像素我们使用的都是原始图像的值，而不是修改后的值。

4 当锚点位于核的左上角时，该过程称作相关操作。——译注

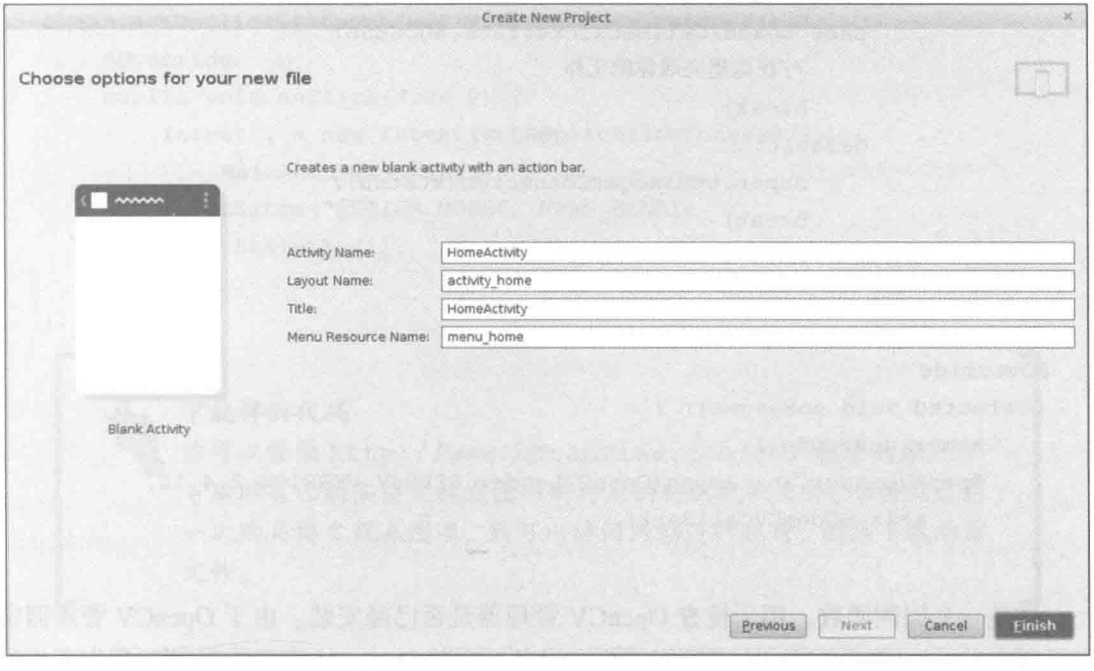
线性滤波最常见的一种用途是降噪。噪声是图像中亮度或色彩信息的随机变化，我们用模糊操作来减少图像中的噪声。

均值模糊方法

均值滤波是最简单的模糊，它对给定核所覆盖的所有像素计算均值。这种操作所采用的核只是一个所有元素为 1 的 Mat[5]，这说明每一个邻域像素都有相同的权值。

在这一章中，我们会从相册中选择一张图片，然后对其做相应的图像变换。为此，我们要添加基本的代码。我们假定 OpenCV4Android SDK 已经部署完毕并且工作正常。

为了满足本章需要，我们可以使用本章开头所创建的第一个 OpenCV 应用。在创建项目的过程中，一些默认名称如下面的截屏所示：



右击 Java 文件夹，单击 New | Activity，创建一个新的 Activity。然后选择 Blank Activity，将 Activity 命名为 `MainActivity.java`，将对应的 XML 文件命名为 `activity_`

5 还要除以核所包含的像素个数，以保证滤波前后的亮度均值一致。——译注

main.xml。打开 res/menu/menu_main.xml，键入如下内容：

```
<item android:id="@+id/action_load_image"
    android:title="@string/action_load_image"
    android:orderInCategory="1"
    android:showAsAction="ifRoom" />
```

考虑到 MainActivity 是我们准备用来执行 OpenCV 具体任务的 Activity，所以需要将 OpenCV 实例化，使其成为 MainActivity.java 的一个全局成员：

```
private BaseLoaderCallback mOpenCVCallBack = new
BaseLoaderCallback(this) {
    @Override
    public void onManagerConnected(int status) {
        switch (status) {
            case LoaderCallbackInterface.SUCCESS:
                //在这里完成你的工作
                break;
            default:
                super.onManagerConnected(status);
                break;
        }
    }
};
@Override
protected void onResume() {
    super.onResume();
    OpenCVLoader.initAsync(OpenCVLoader.OPENCV_VERSION_2_4_10,
        this,mOpenCVCallBack);
}
```

这是一个回调函数，用于检查 OpenCV 管理器是否已经安装。由于 OpenCV 管理器定义了所有的 OpenCV 函数，所以我们需要在设备上安装 OpenCV 管理器应用[6]。假如不想使用 OpenCV 管理器，也可以在本地包含函数，但是 APK 的大小就会显著地增加。如果没有

6 OpenCV 管理器应用（OpenCV Manager）的安装包位于 OpenCV Android SDK 目录下的 apk 文件夹中，需要根据设备的 CPU 架构来选择具体的 apk 文件。——译注

找到 OpenCV 管理器，应用会跳转到 Play Store 去下载。在 onResume() 中调用的函数加载 OpenCV 以供使用。

接下来，我们为 activity_main.xml 添加一个按钮：

```
<Button
    android:id="@+id/bMean"
    android:layout_height="wrap_content"
    android:layout_width="wrap_content"
    android:text="Mean Blur" />
```

然后，在 HomeActivity.java 中，将该按钮实例化，并为这个按钮设置 onClickListener()：

```
Button bMean = (Button)findViewById(R.id.bMean);
bMean.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        Intent i = new Intent(getApplicationContext(),
            MainActivity.class);
        i.putExtra("ACTION_MODE", MEAN_BLUR);
        startActivity(i);
    }
});
```

下载样例代码

你可以登录 http://www.broadview.com.cn，在你的账户下下载所有已购买博文视点图书所对应的样例代码文件。如果你是第一次购买博文视点图书，则可以访问网站进行注册，然后下载所需文件。

在上面的代码中，MEAN_BLUR 是值为 1 的常量，它规定了我们要进行何种操作。

在这里我们为 Activity 包增加了一些用以区分操作类型的部分。

打开 activity_main.xml，并用下面的代码片段替换所有内容。这段代码添加了两个 ImageView 元素：一个用来显示原始图像，一个用来显示处理后的图像。

```
<?xml version="1.0" encoding="utf-8"?>
   <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
       android:orientation="vertical"
       android:layout_width="match_parent"
       android:layout_height="match_parent">

   <ImageView
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:layout_weight="0.5"
       android:id="@+id/ivImage" />

   <ImageView
       android:layout_width="match_parent"
       android:layout_height="match_parent"
       android:layout_weight="0.5"
       android:id="@+id/ivImageProcessed" />

   </LinearLayout>
```

我们需要用 Java 以编程方式链接这些 ImageView 元素和 MainActivity.java 中的 ImageView 元素：

```
private final int SELECT_PHOTO = 1;
private ImageView ivImage, ivImageProcessed;
Mat src;
static int ACTION_MODE = 0;

@Override
protected void onCreate(Bundle savedInstanceState) {
    //Android的相关代码
    ivImage = (ImageView)findViewById(R.id.ivImage);
    ivImageProcessed =
    (ImageView)findViewById(R.id.ivImageProcessed);
    Intent intent = getIntent();
    if(intent.hasExtra("ACTION_MODE")){
        ACTION_MODE = intent.getIntExtra("ACTION_MODE", 0);
}
```