

实验 1-A. 设置 Python+OpenCV+VS Code 开发环境

1. 参考提供的视频“Python-Open-环境.mp4”,设置 Python+OpenCV+VS Code 开发环境;
2. 在新建的.py 文件中输入如下代码,观察是否能运行成功。

```
import cv2
im = cv2.imread('Flower.png')
cv2.imshow('The 1st Window', im)
cv2.waitKey()
cv2.destroyAllWindows('The 1st Window')
```

实验 1-B. 练习图像分离、截取、转换等函数的使用

1. 打开彩色图像 lenna.jpg,使用 cv2.split()分离红色、绿色和蓝色通道,分别用 imshow() 显示;使用 cv2.merge()将分离后的三通道图像合并成一幅彩色图像,尝试不同的顺序时的结果:

```
im = cv2.imread('lenna.jpg')
b,g,r = cv2.split(im)
...
xim = cv2.merge([b,g,r])
...
```

2. 使用 cvtColor()将彩色图像转换成灰度图像,与上述通过分离获得的灰度图像作比较,观察与哪个分量图像接近。

```
xim = cv2.cvtColor(src, cv2.COLOR_BGR2GRAY);
```

3. 打开图像 lenna.jpg,保存到矩阵 im,使用如下形式的定义,截取从第 20 行至第 180 行,第 30 列至 220 列的区域图像,且使用 cv2.imwrite('FileName', rim) 保存区域图像

```
rim = im[20:180, 30:220, :];
cv2.imwrite(...)
...
```

4. 打开图像 lenna.jpg,使用 cv2.flip()产生水平镜像、垂直倒影和对角翻转图像,且加以显示

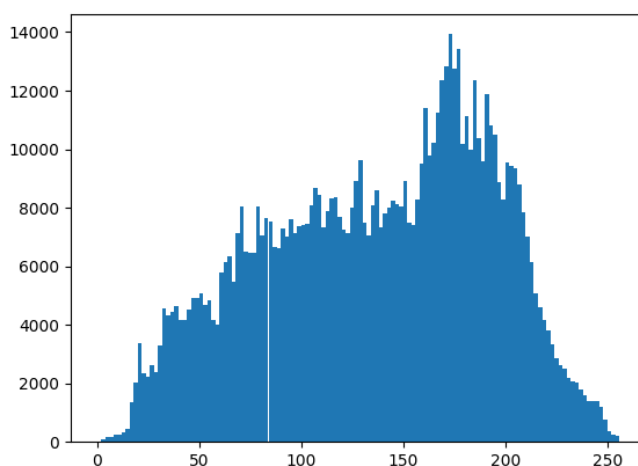
实验 1-C. 图像颜色变换

打开图像 Flower.png,将左上角坐标 (400, 360)、右下角 (670, 660) 矩形范围内的颜色求其补色 (即 Red、Green、Blue 分别计算为 255-Red, 255-Green, ...) 产生形如下图的结果;



实验 1-D. 计算和显示直方图

1. 打开图像 `lenna.jpg`，使用 `cv2.calcHist()` 函数计算灰度图像的直方图，以屏显打印输出的形式显示灰度值介于 100-128 的像素数目；
2. 通过设置 `cv2.calcHist()` 的参数，计算 `binwidth` 等于 8、16 的直方图；
3. 设计算所得的直方图保存在数组：`xhist`，搜索查找具有最多像素数目的灰度级。
4. 使用 `matplotlib` 包画出如下的直方图（可参考视频 `me_hist_plot.mp4`）。



实验 1-E. 图像统计处理

1. 打开 Flower.png，使用 `minMaxLoc()` 统计绿色分量图像的最大值和最小值，`print` 输出；
2. 使用 `meanStdDev()` 统计上述绿色分量图像的均值和标准差，且 `print` 输出；
3. 请自行 Internet 搜索上述函数的使用。

实验 1-F. 随机截取子图像且拼接成新图像

打开 Lenna.jpg，以随机选定位置为左上角截取 4 个 120×80 的子图像，从左到右排列成一个 120×320 的新图像（如下图示意），并加以显示。

