

《数字图像处理》实验报告

时间: 2021 年 10 月 23 日 (第 8 周, 星期 六)

地点: 测绘工程实验室 (B 区) -- B 实验大楼 202

学生姓名: 魏子继 学号: 20194947

一、实验名称:

图像直方图统计

二、实验目的:

该实验的目的是图像直方图统计, 分为两个方面:

1. 灰度图像的灰度值统计
2. 彩色图像的三通道 B、G、R 值统计

要求输出的直方图图像简单而又美观, 同时各个参数齐全, 参数代表含义和参数值域简单给定。

三、实验步骤:

根据实验的目的和实现方法, 该实验由以下三步完成:

1. 灰度图像的像素每个灰度值的个数统计。统计过程中使用了 cv2 库中的 calcHist 函数, 该函数的参数是 images、channels、mask(掩膜)、histSize、ranges, 其中 channels 代表的是统计图像的通道个数。该函数返回一个数组, 数组的大小为[ranges, 1], 一般为[256, 1]。

2. 彩色图像三通道像素 B、G、R 值的统计。在该统计的过程中，依然是选择采用 cv2 库中的 calcHist 函数，与统计灰度图像灰度值的区别是在本次统计中，参数 channels 由 0 变为 3。
3. 利用 matplotlib.pyplot 绘制直方图。这个过程中使用了该库中的 hist 函数，这个函数能够方便的对输入参数数组返回其直方图表示。虽然也可以使用 cv2 来绘制直方图，但那个方法绘制直方图相比之下更加困难，因此选择简单的方法。

四、 实验中的关键点分析（包括关键算法与代码实现）：

直方图统计（Hist.py）

```
1. """
2. Created by Chloe on 2021/10/30
3. """
4.
5. import cv2.cv2 as cv2
6. # import numpy as np
7. import matplotlib.pyplot as plt
8.
9. plt.rcParams['font.sans-serif'] = ['SimHei'] # 图像字体汉化
10.
11.
12. # 统计直方图参数-matplotlib
13. def calHist_mat(img):
14.     # 图片转化为灰度图
15.     gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
16.     # calcHist 返回一个数组，该例中为 256*1 的数组，值为对应灰度的像素个数，下标即为像素值
17.     # calcHist 函数的参数依次为:images, channels, mask(掩膜), histSize, ranges
18.     hist = cv2.calcHist([gray], [0], None, [256], [0, 255]) # mask 为 None 表示处理整幅图像
19.     return hist
20.
```

```

21.
22. # 绘制三通道直方图 (BGR) -matplotlib
23. def calHistThree_mat(img):
24.     color = ('b', 'g', 'r')
25.     for i, col in enumerate(color):
26.         hist_three = cv2.calcHist([img], [i], None, [256], [0, 255])
27.         # 直接绘制三通道直方图的图像
28.         plt.subplot(224), plt.plot(hist_three, color=col)
29.         plt.xlim([0, 256]), plt.title('Histogram')
30.     return 0
31.
32.
33. # 主程序
34. if __name__ == '__main__':
35.     # 读入图片
36.     image = cv2.imread("D:\\pyfiles\\DigitalImageProcessing\\image\\sylm1
        .jpeg")
37.
38.     # 转换为灰度图像
39.     grayImg = cv2.cvtColor(image, cv2.COLOR_BGR2GRAY)
40.
41.     # 统计直方图参数
42.     hist_num = calHist_mat(image)
43.     # print(hist_num)
44.
45.     # 绘制直方图
46.     plt.subplot(221), plt.imshow(grayImg, 'gray')
47.     plt.title('grayImage'), plt.axis('off') # 221 处为灰度图像
48.     plt.subplot(222), plt.hist(grayImg.ravel(), 256, [0, 256]) # 使用
        hist 算法绘制直方图
49.     plt.title('Histogram'), plt.xlim([0, 256]) # 222 处为灰度直方图
50.     plt.subplot(223), plt.imshow(cv2.cvtColor(image, cv2.COLOR_BGR2RGB))
51.     plt.title('OriginalImage'), plt.axis('off') # 223 为原图
52.     calHistThree_mat(image) # 224 为三通道直方图
53.
54.     plt.tight_layout() # 自动调整子图间距
55.     plt.savefig("D:\\pyfiles\\DigitalImageProcessing\\result\\Hist.png")
56.     plt.show()

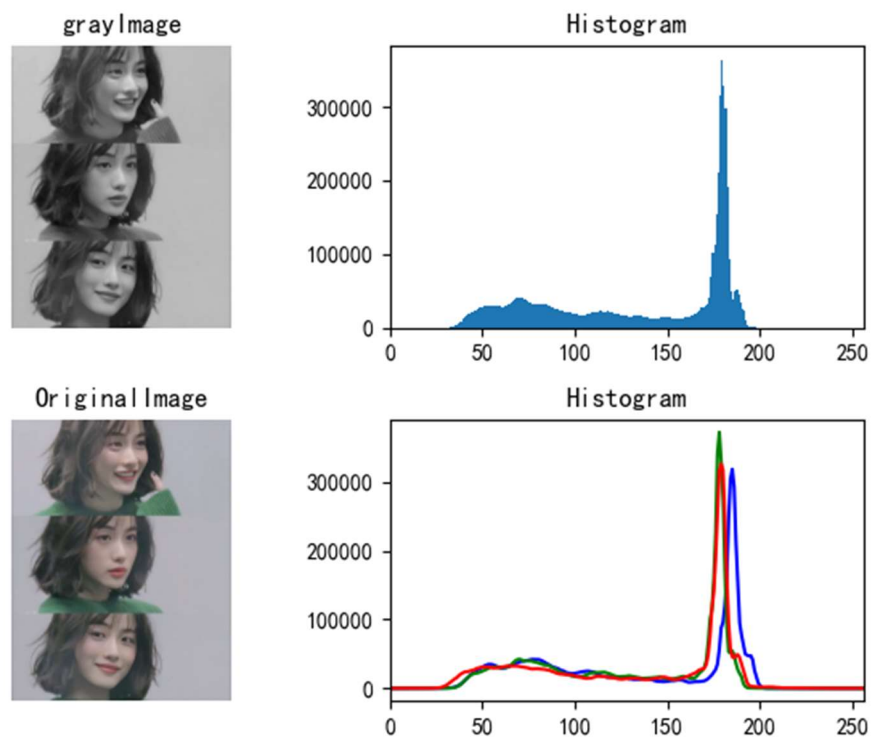
```

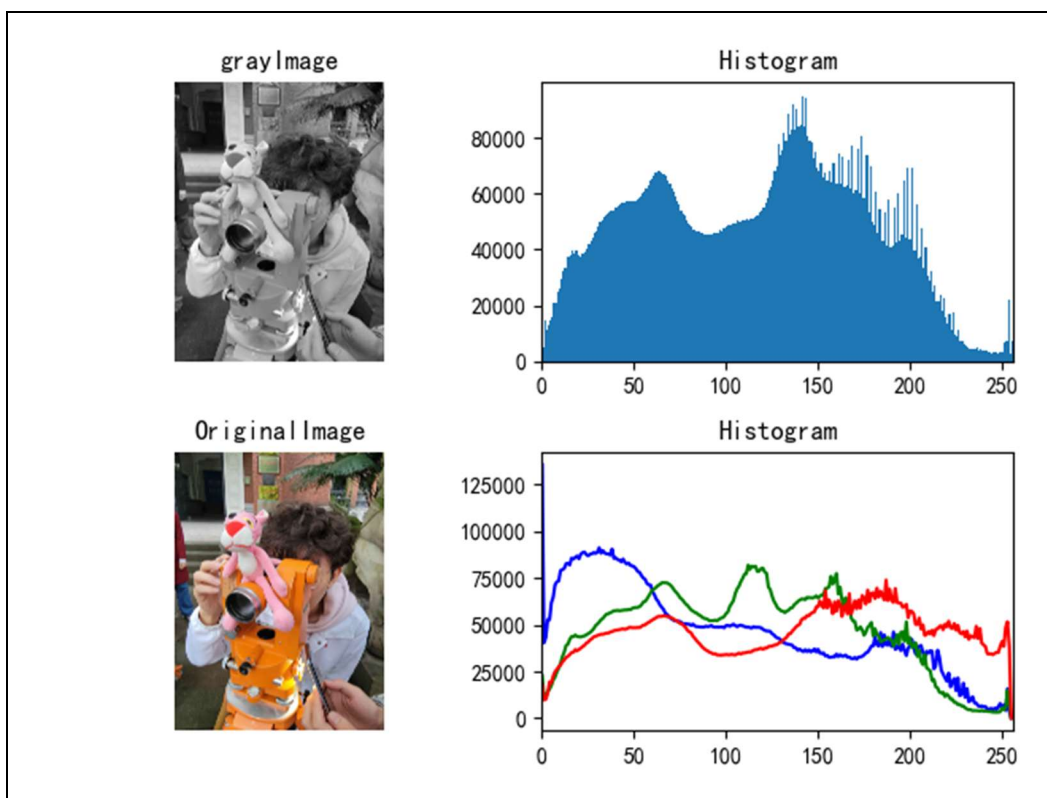
五、 实验原始数据与实验结果：

源图像：



结果图像：





六、 问题分析与心得体会：

在实验过程中，主要遇到了以下问题并做出总结：

1. 在进行本实验之前，我的计划是用 Java 来完成本学期数字图像处理实验的，但是当使用 Java 绘制直方图后打印直方图时，出现了很明显的错误（在 255 处显示有很多，但是在 0 处显示几乎没有像素值。但是当遍历灰度化图像数组灰度值后，发现实际情况并非如此）。因此，在考虑到 Java 目前图形化设计编程（包括界面设计和此次的直方图绘制）等还不是特别熟悉的情况下，选择使用易于绘图的 Python 语言重新编写代码。另一方面，在使用 Python 语言实现算法时，发现 Python 对于 opencv 是易于安装的。Opencv 库对于图像处理方面有许多现成的函数供使用者使用，利用这些函数能够与

自己写的函数做一个参照，检查自己写的函数是否有较大纰漏。综上所述，选择 Python 语言为本系列实验的实现语言。

2. 图像选择方面，本身也是选择最爱的十元。但是那个图像的背景颜色极致的统一，并呈现明显的暗灰色，致使统计出的直方图出现了靠近 0 和 255 处的数值非常地少。最初以为是语法设计的问题，但遍历灰度图像数组和 `calcHist` 统计的结果后发现事实确是如此，因此对图像进行重新选择。选择了在社群内传播广泛的、色彩丰富的邹野同学使用光学经纬仪进行测量时的照片，这个图像的色彩相比最爱的十元是极大丰富的，因此绘制出的直方图也是非常均匀的在整个坐标轴上的，同时彩色图像的直方图统计数目也是很多的。
3. 在该代码运行后遇到了一个不影响最后出图的问题，先将此问题的报错截图如下：

```
E:\anaconda\anaconda3\python.exe "E:\Pycharm\PyCharm 2021.2.2\plugins\python\helpers\pydev\pydevconsole.py"
--mode=client --port=51358

import sys; print('Python %s on %s' % (sys.version, sys.platform))
sys.path.extend(['D:\\pyfiles', 'D:/pyfiles'])

Python 3.8.8 (default, Apr 13 2021, 15:08:03) [MSC v.1916 64 bit (AMD64)]
D:/pyfiles/DigitalImageProcessing/code/Hist.py:27: MatplotlibDeprecationWarning: Adding an axes using the same
arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will
always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by
passing a unique label to each axes instance.
  plt.subplot(224), plt.plot(hist_three, color=col)
D:/pyfiles/DigitalImageProcessing/code/Hist.py:27: MatplotlibDeprecationWarning: Adding an axes using the same
arguments as a previous axes currently reuses the earlier instance. In a future version, a new instance will
always be created and returned. Meanwhile, this warning can be suppressed, and the future behavior ensured, by
passing a unique label to each axes instance.
  plt.subplot(224), plt.plot(hist_three, color=col)
```

根据资料查询和对于该错误的翻译初步认为是因为在彩色图像的直方图统计中没有添加三条曲线的标签而导致的，但没有影响出图结果，并且在加入标签后还是在报错，因此

没有办法确定错误情况的条件下，就没有管他了。

4. 该代码还有一个更加遗憾的地方，即没有将绘制彩色图像直方图统计的那部分抽象出来，即他是依托于本图像 subplot 的，如果想单独输出一张彩色图像还需要对函数进行修改，这是我们不乐意看到的。这个问题影响到了后期界面设计的完整性，是很遗憾的一个地方。一个较好的方法是将彩色图像直方图的结果存入一个适当格式的数组之中，再利用这个数组进行 plot 画图。