上机实验十
实验报告

刘瀚文
517030910294

2018 年 11 月 26 日

目录

Part I

实验十

Chapter 1

实验准备

1.1 环境配置

1.1.1 OpenCV

OpenCV 的安装使用 pip install 的方法进行安装。直接就成功了。
环境说明：Python 2.7.15 OpenCV 3.4.3
使用 Ubuntu 14.04 VMware 14

1.2 背景知识

1.2.1 图像特征抽取

数字图像在计算机中的表示和存储

灰度图像的表示 图像的基本元素是“像素 (pixel)”。一幅宽为 W，高为 H 的灰度图像在计算机中用一个 W×H 的矩阵存储。矩阵的每个元素是图像对应位置像素的灰度值，范围在 0 到 255 之间。灰度值 0 表示黑色，灰度值 255 表示白色。图像坐标系以左上角为原点，横向为 x 方向，纵向为 y 方向。

彩色图像的表示 彩色图像的每个像素由红色 (R)、绿色 (G)、蓝色 (B) 三个颜色分量构成，从而能够呈现多种色彩。一幅宽为 W，高为 H 的彩色

图像在计算机中用一个 W×H×3 的矩阵存储，其中最后一个维度表示 RGB 颜色空间。

灰度图像的梯度 梯度：

灰度图像的梯度

假设I(x, y)表示一幅灰度图像，则它X方向的梯度定义为：

$$I_x(x,y)=\frac{\partial I(x,y)}{\partial x}=I(x+1,y)-I(x-1,y)$$

Y方向的梯度定义为：

$$I_y(x,y)=\frac{\partial I(x,y)}{\partial y}=I(x,y+1)-I(x,y-1)$$

梯度强度定义为

$$M(x,y)=\sqrt{I_x(x,y)^2+I_y(x,y)^2}$$

* 图像边界处的梯度无法根据上述梯度定义求出，并且边界的梯度的定义常常随不同应用场景有改变，如有的定义边界像素的梯度为零，有的定义边界像素的梯度等于它临近的非边界像素的梯度。本实验中涉及梯度的地方均不考虑边界像素的梯度。

图像特征抽取

至今为止, 图像特征没有通用和精确的定义，图像特征的定义往往由问题或者应用类型决定。图像特征提取的总体目标是用尽可能少的数据量最大程度地描述图像信息。

图像直方图特征 图像的直方图特征是一种全局统计特征, 描述了图像的某一数值特性的分布情况, 反映了图像的大体风格。

颜色直方图 彩色图像 RGB 三种颜色分量在整张图像中所占的相对比例，反映了图像全局的主体色调。

灰度直方图 灰度图像灰度值的分布情况，反映了灰度图像的明暗程度。

梯度直方图 灰度图像的梯度强度分布情况，反映了图像的纹理的疏密程度。

Chapter 2

Exercise

2.1 要求

1. 将 img1.png 和 img2.png 两幅图像以彩色图像方式读入，并计算颜色直方图；

2. 将 img1.png 和 img2.png 两幅图像以灰度图像方式读入，并计算灰度直方图和梯度直方图。

2.2 实验过程

在本此实验中，由于 opencv 的库比较多，所以尝试了很多种方法来获取图片的信息，来学习尽可能多的知识。

2.2.1 读入图片

```
img = cv2.imread("img2.png", cv2.IMREAD_GRAYSCALE)
```

2.2.2 读入灰度图像，计算灰度直方图

numpy.array () img.flatten()

使用 numpy 中的操作获取图片的灰度信息，记录在列表中。

```
    list = numpy.array(img).flatten()
```

matplotlib.pyplot

使用 matplotlib 进行图片的绘制。

```
import matplotlib.pyplot as plt


plt.hist(list, number, density=1)

plt.legend()

plt.xlabel("Gray")
plt.title("GrayScale")

plt.show()
```

其中的参数是 list 作为传入的原始数据数组，对应直方图中的原始分布数据；number 是将直方图分为多少个直方呈现；density=1 则是让整体的分布进行归一化调整之后，用频率代替频数更好的展现分布；xlabel 则是横坐标的标签；title，show 等信息的含义清晰、显然。

2.2.3 读入彩色图像，计算彩色直方图

cv.calcHist()

在彩色图像的处理中，直接采用的 cv 的函数进行操作。

```
import numpy as np
import cv2 as cv
from matplotlib import pyplot as plt

img = cv.imread('img1.png')
color = ('b', 'g', 'r')
for i, col in enumerate(color):
histr = cv.calcHist([img], [i], None, [256], [0, 256],)
plt.plot(histr, color=col)
plt.xlim([0, 256])
plt.show()
```

对于 calcHist() 核心函数参数的解读：

image 输入图像，传入时应该用中括号 [] 括起来

channels 传入图像的通道，如果是灰度图像，就只有一个通道，值为 0；如果是彩色图像（有 3 个通道），那么取值在 0,1,2, 中选择一个，对应着 BGR 各个通道。这个值也需用 [] 传入。

mask 掩膜图像。如果统计整幅图，那么为 none。主要是如果要统计部分图的直方图，就得构造相应的膜来计算。

histSize 灰度级的个数，需要中括号，比如 [256]

ranges 像素值的范围，通常 [0,256]，有的图像如果不是 0-256，比如说你来回各种变换导致像素值负值、很大，则需要调整后才可以。

2.2.4 读入灰度图像，计算灰度梯度直方图

在这个实验中，使用的是老老实实的计算方式，用自己写的求平方根来进行操作。

```
H = len(img)
W = len(img[0])

for j in range(1, H - 2):
  for k in range(1, W - 2):
    Ix = int(img[j][k - 1]) - int(img[j][k + 1])
    Iy = int(img[j - 1][k]) - int(img[j + 1][k])
    I = round((Ix ** 2 + Iy ** 2) ** 0.5)
    list.append(I)
```

这个操作就是严格按照定义来进行，在写的过程中我也进行了思考，关于为什么横向差值为 2，但是不需要除以二。我猜想可能是因为在计算的过程中做出假设：相邻的块的边界的灰度值为该块的灰度，但对于本色块来说，距离只有单位的 1，所以无需除以看起来像的 2.

同时在梯度的计算中，使用的 number=360，进行分割。

其余的部分比较相似。

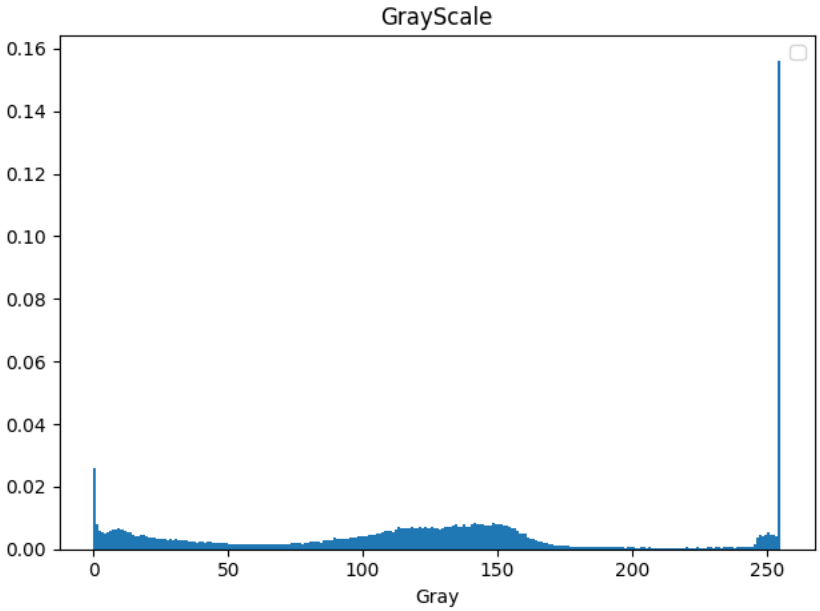
2.3 结果展示

2.3.1 先呈现原始的图片

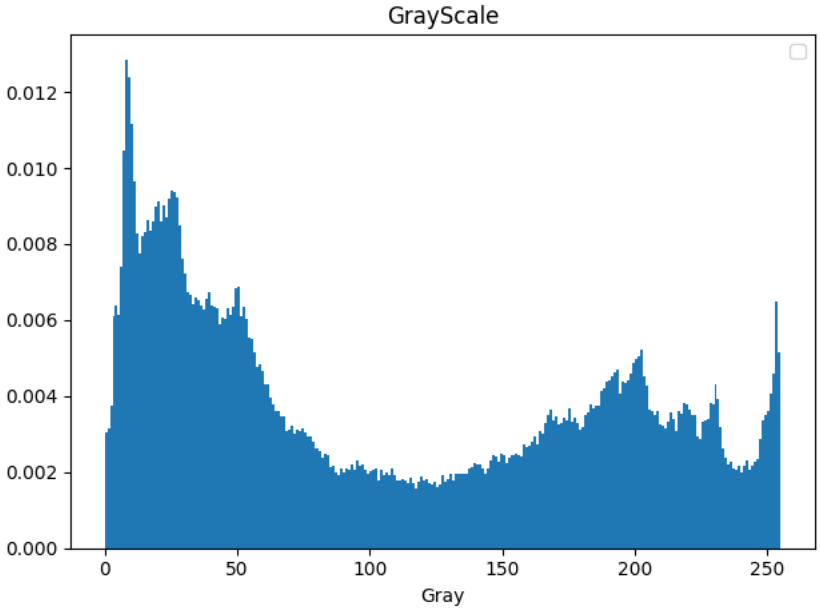
img1

img2

2.3.2 读入灰度图像，计算灰度直方图

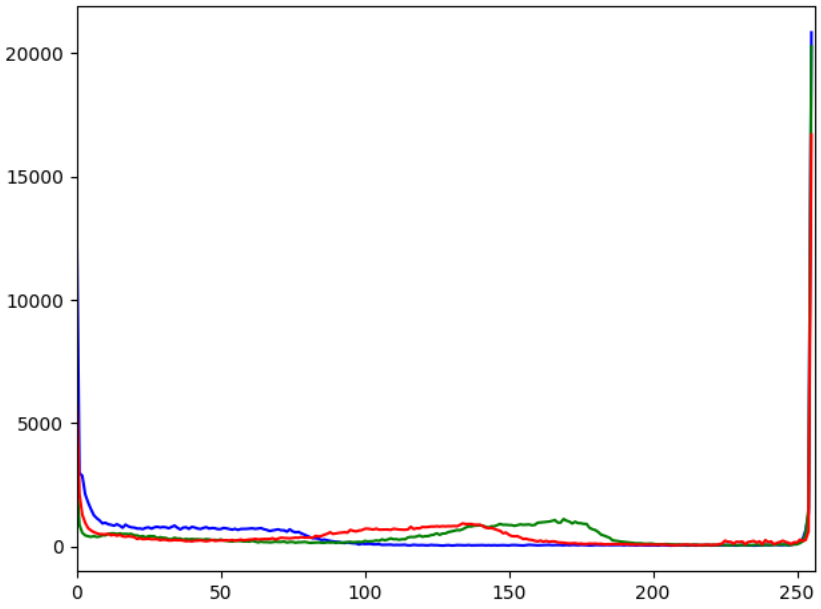


Gray1

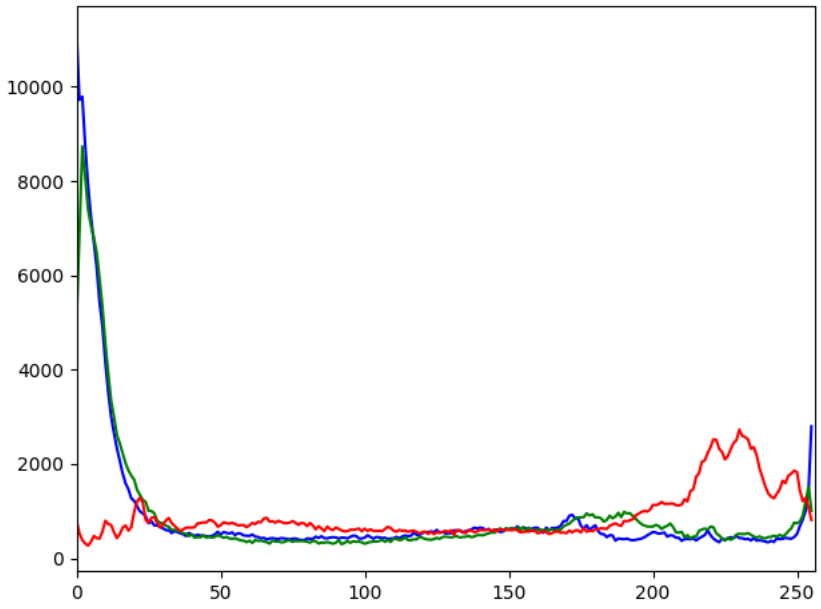


Gray2

2.3.3 读入彩色图像，计算彩色直方图

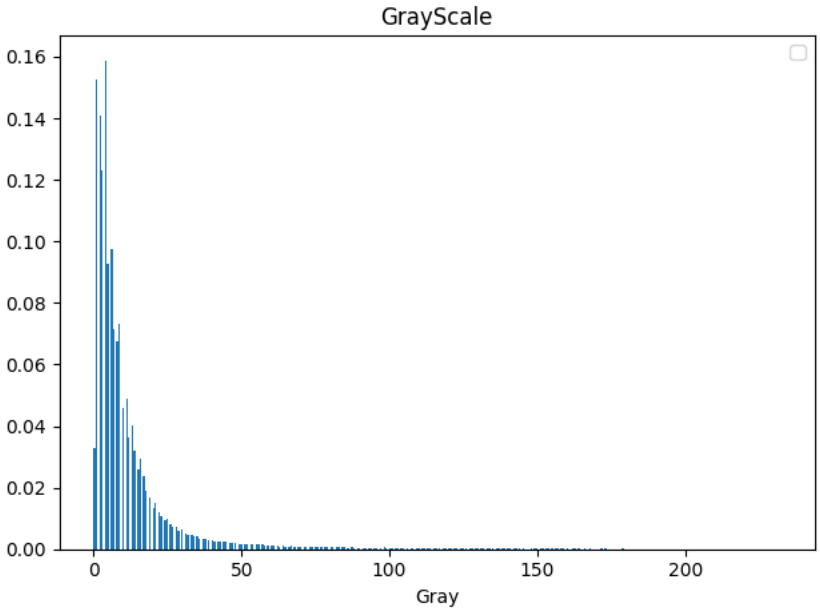


Color1

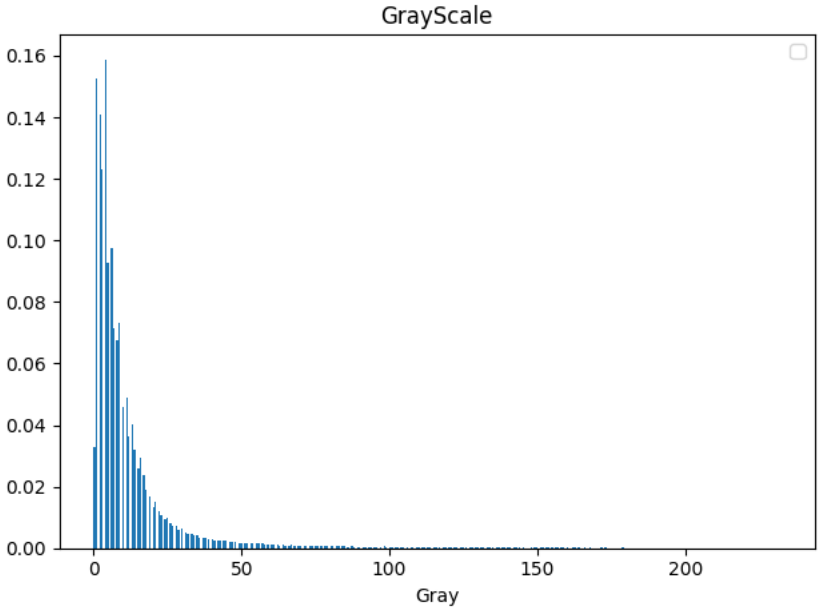


Color2

2.3.4 读入灰度图像，计算灰度梯度直方图



GrayGrad1



GrayGrad2

Part II

实验总结

这次实验开始学习对图像的应用，很有趣！

有一些同学第一个学期大作业做过和 python 相关的图像处理，那个时候我选择的是另一个方向。当时一直想能有个机会学习 OpenCV 相关的知识，现在终于能有机会系统的接触了，很开心！

期待在下一次实验中学习更多的知识！