

电工导实验报告 5

一、实验目的

1. 了解图像的特征抽取

二、实验内容

1. 学习数字图像在计算机中的存储和表示
2. 图像特征的抽取
3. 搭建编译环境

三、实验环境

1. Python 2.7 + opencv +numpy

四、实验步骤

首先安装 opencv, opencv 是一个跨计算机视觉库, 功能强大, 由一系列 C 函数和少量 C++ 类构成, 同时提供了 python 等语言的接口, 实现了图像处理和计算机视觉方面的很多通用算法。安装后 import 无错误信息则可继续。

图像又灰度表示, 和彩色表示, 我们学习了梯度的计算, 得到了图像的一些性质。图像的特征迄今位置都没有通用和精确的定义, 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。我们主要研究了颜色直方图, 灰度直方图和梯度直方图。

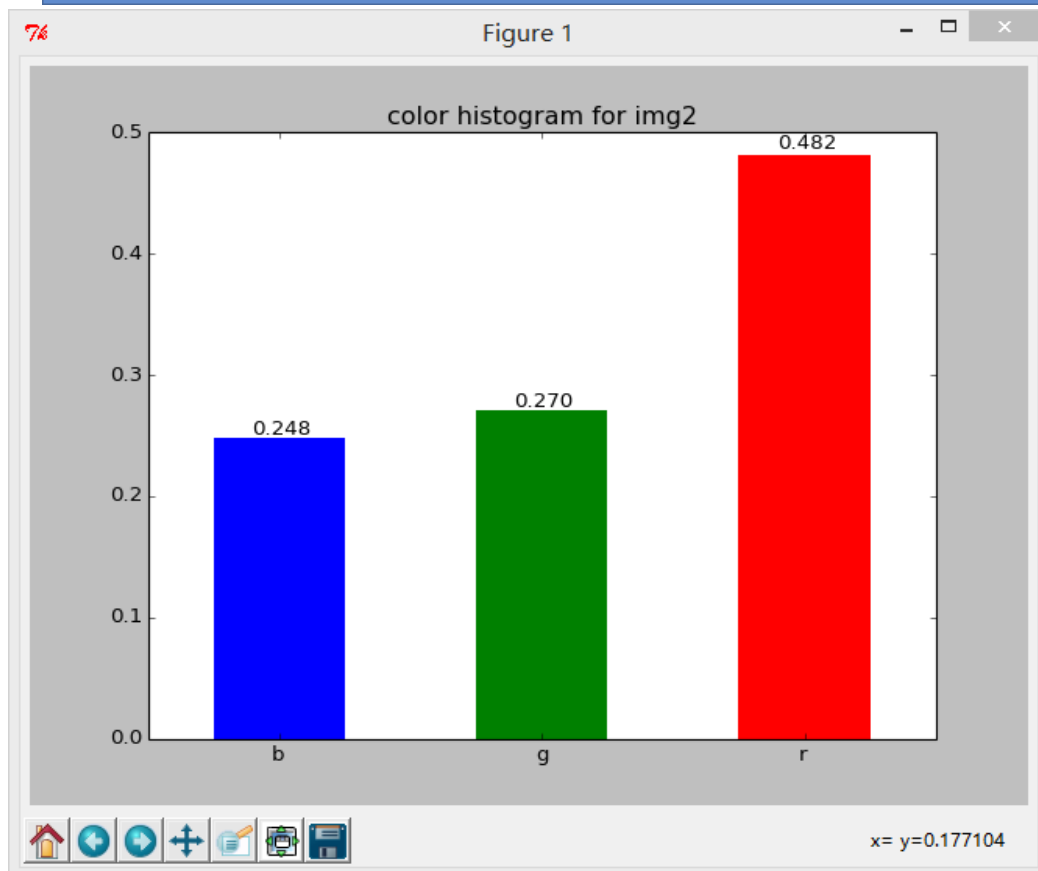
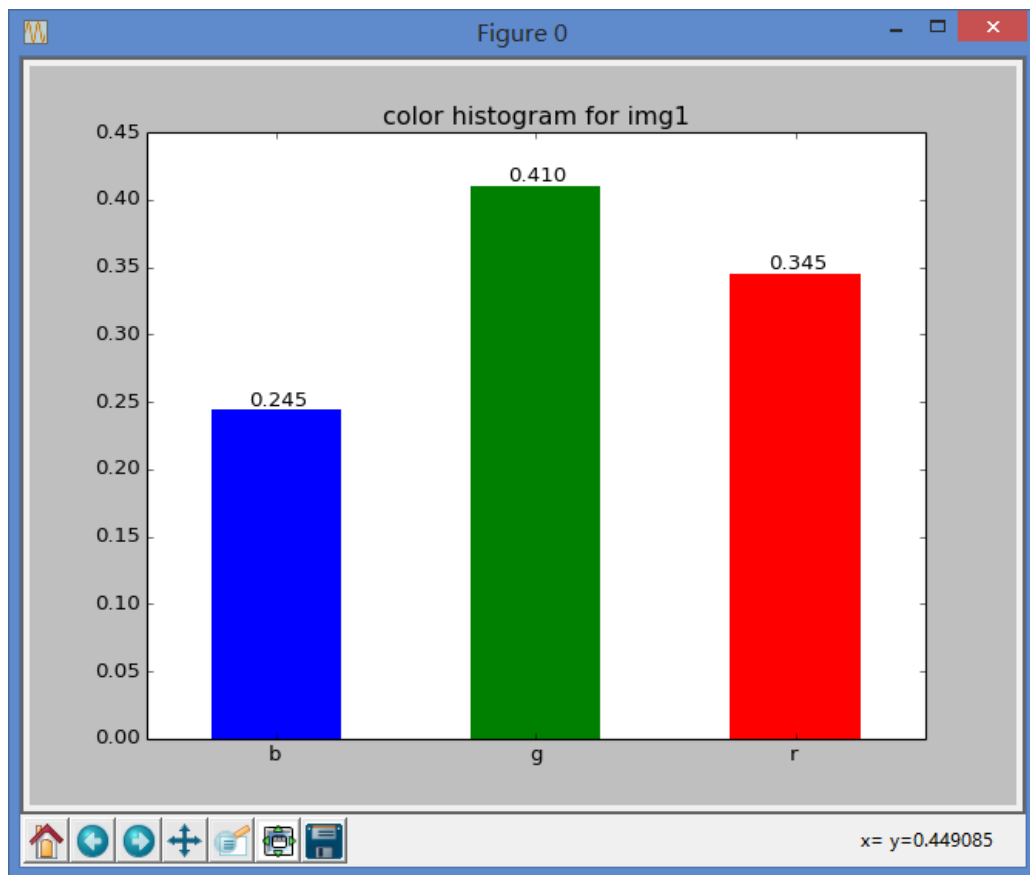
五、问题及其解决

对两幅图分别进行分析, 得到灰度, 色彩和梯度的直方图。

色彩直方图的统计最为简单, 通过 split 将图像分成 RGB 三个矩阵, 分别累加然后计算占比。这里我开始天真的以为就是 RGB 的顺序, 实际上是 b, g, r 的顺序, 让我找错找了半天。

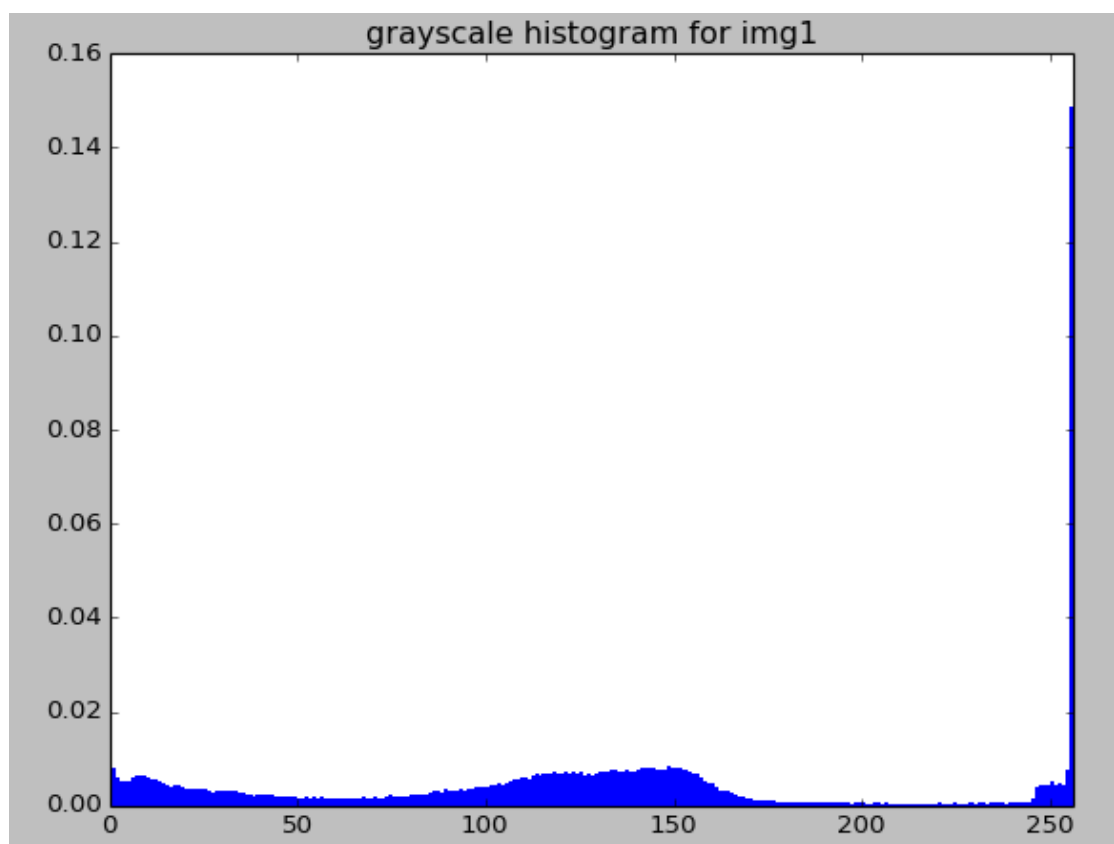
```
def RGB(img):  
    b, g, r = cv2.split(img)  
    x = b.sum()  
    y = g.sum()  
    z = r.sum()  
    total = x+y+z  
    rate = [float(x)/total, float(y)/total, float(z)/total]  
    return rate
```

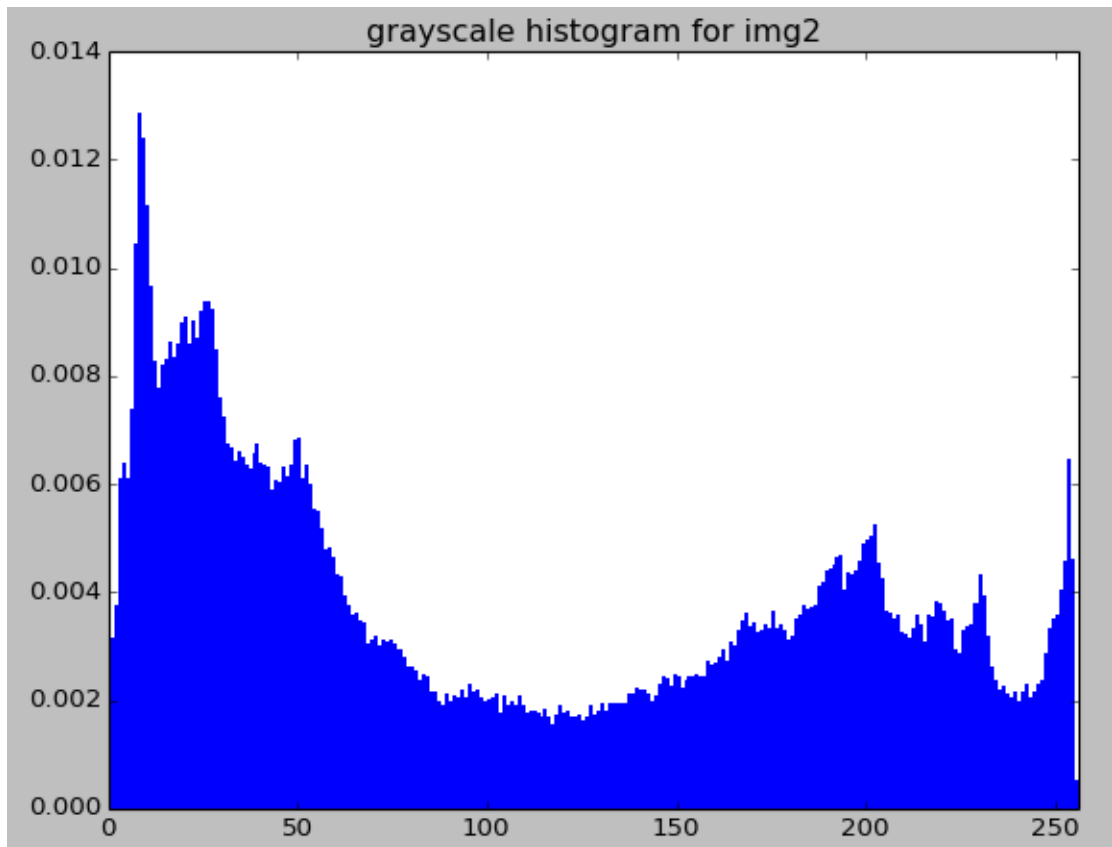
另一个问题则是如何绘制直方图, 虽说可以用 EXCEL 画, 但是还是用 python 一步到位会比较好一些。我通过调用 matplotlib 库中的 pyplot, 成功进行了绘图, 效果非常不错, 直方图的效果如下:



灰度直方图则通过灰度模式读取，把 0~255 的值分别进行统计，计算出占比为多少，最后画出直方图。画直方图这里我通过 `plt.axis([0, 256, 0, 0.16])` 的语句来限定 x 和 y 的范围，使的图像更加美观。

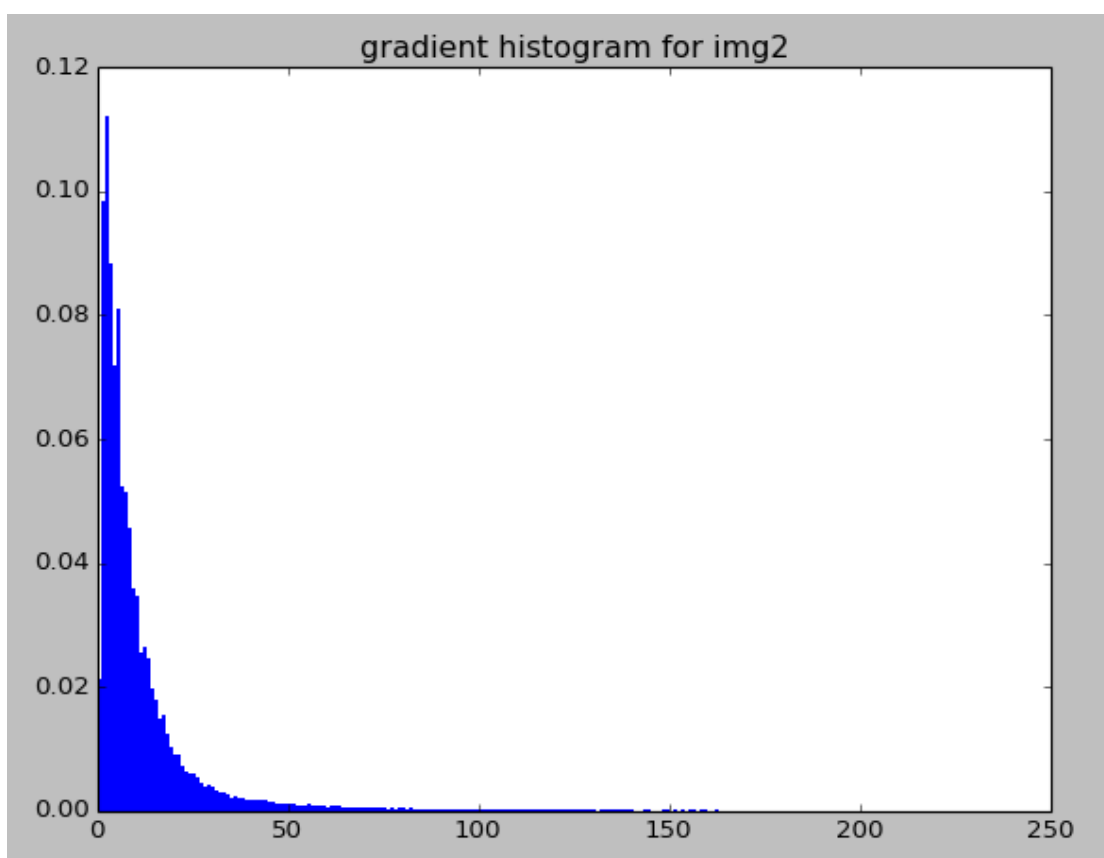
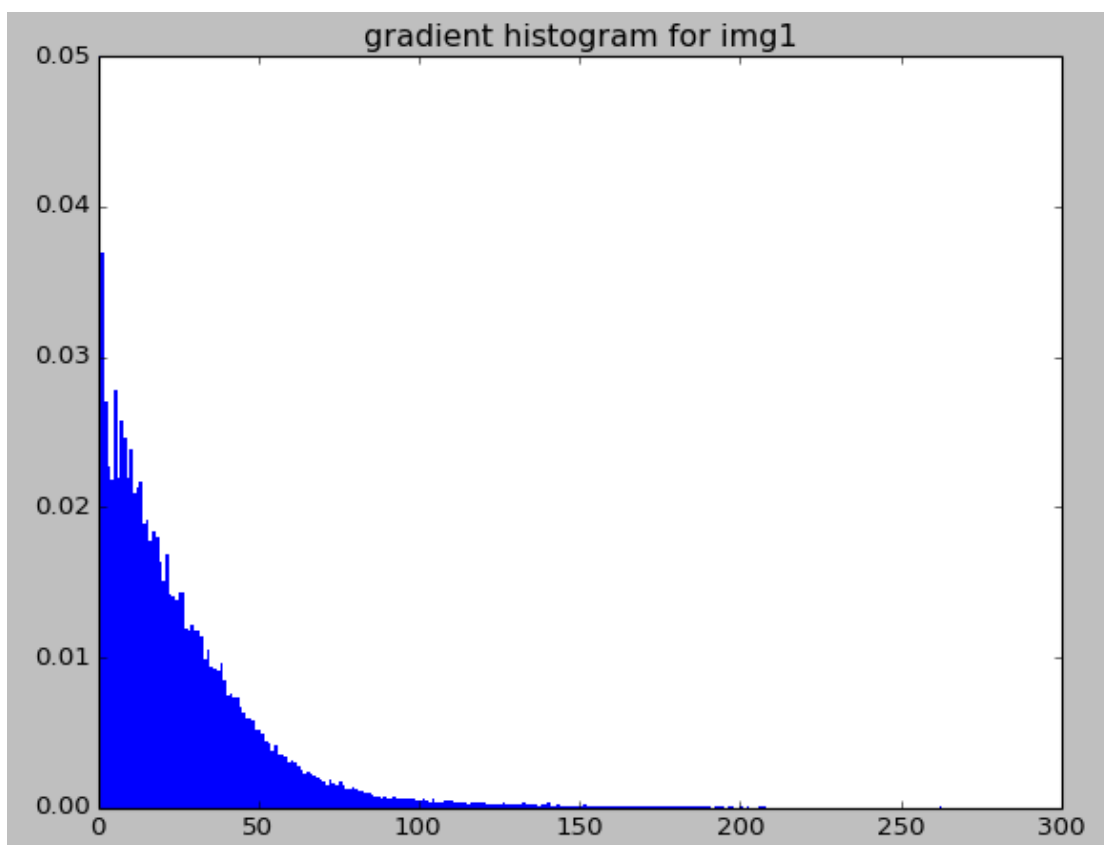
```
def GARY(img):  
    m=[0]*256  
    total = 0  
    for i in range(len(img)):  
        for j in range(len(img[i])):  
            m[img[i][j]]+=1  
            total += 1  
    for i in range(256):  
        m[i] = float(m[i])/total  
    return m
```





最后是梯度的直方图，方法差不多，无非多了一步计算梯度。

```
def GRADIENT(img):
    gradient=361*[0]
    total = 0
    for i in range(len(img)):
        if i==0 or i==len(img)-1:
            continue
        for j in range(len(img[i])):
            if j==0 or j==len(img[i])-1:
                continue
            x=math.fabs(int(img[i+1][j])-int(img[i-1][j]))
            y=math.fabs(int(img[i][j+1])-int(img[i][j-1]))
            m = (x**2+y**2)**.5
            gradient[int(m)] += 1
            total += 1
    for i in range(len(gradient)):
        gradient[i]=float(gradient[i])/total
    return gradient
```



为了显示的方便，我使多张图像同时显示，而且对每个直方图还能进行相应的调整。

```
def draw(data, name):
    global count
    plt.figure(count)
    plt.bar(left=range(256), height=data, width=1, linewidth=0)
    if count==0:
        plt.axis([0,256,0,0.16])
    if count==1:
        plt.axis([0,256,0,0.014])
    plt.title(name)
    count += 1

N = 2
count = 0
for i in range(N):
    plt.figure(i)
    img1 = cv2.imread("img1.png", cv2.IMREAD_GRAYSCALE)
    data1 = GARY(img1)
    draw(data1, "grayscale histogram for img1")
    img2 = cv2.imread("img2.png", cv2.IMREAD_GRAYSCALE)
    data2 = GARY(img2)
    draw(data2, "grayscale histogram for img2")
plt.show()
```

六、实验总结

最近的课程方向主要是图像处理。数据不仅存在数据，图像里也有大量的数据，但是对图像来说，图像的特征等很难被计算机进行处理，这也就是我们学习这方面知识的原因。通过这次的实验，我对图像的基本处理了解了很多，我很期待之后的学习。

F1403023 5140309534 韩坤言