数字图像处理第二次实验Gamma校正报告（文末附代码）

1 问题

实现一幅灰度图像的伽玛校正

2 问题分析

本次实验验证课堂上讲过的gamma变换，验证gamma变换的效果。

Gamma变换 ：

其中两个主要参数 c 和 gamma, r是输入像素值。

我先将图像进行归一化映射到[0,1]区间，用上述公式进行变换，最后再反变换得到输出图像

为了验证gamma校正的效果，我利用 ppt3.3.2内容模拟阴极射线管的gamma校正的过程。

1 首先对输入图像进行 gamma= 2, c=1的gamma变换，得到污染图像

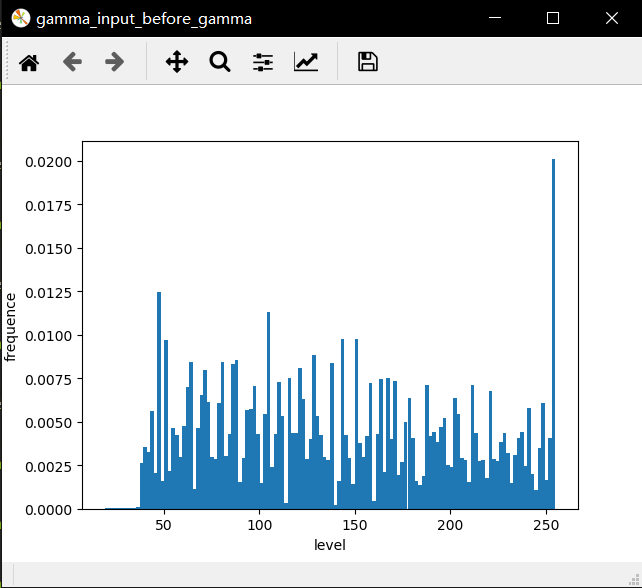
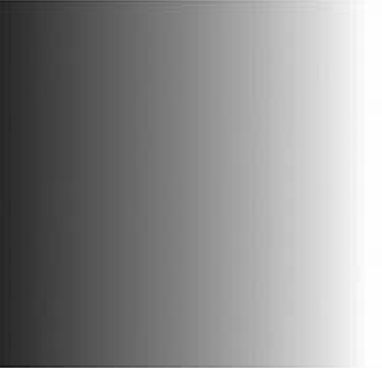
2 对输入图像进行 gamma=0.5,c =1 的 gamma变换得到中间图像

3 对中间图像进行 gamma=2,c=1的 gamma变换得到校准图像

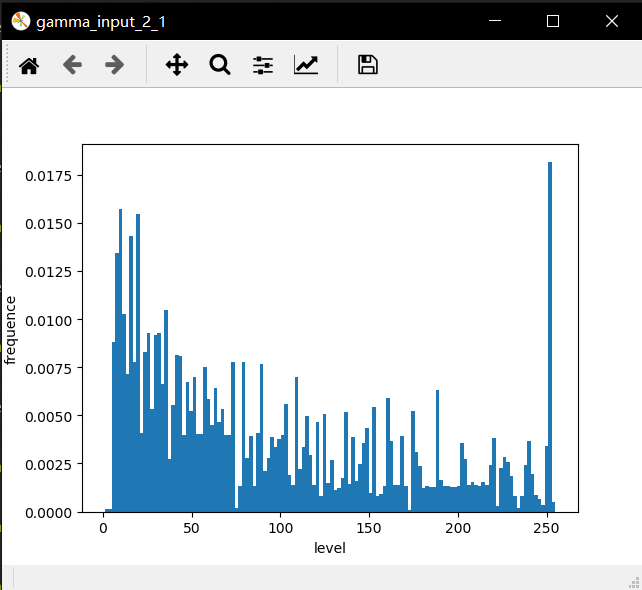
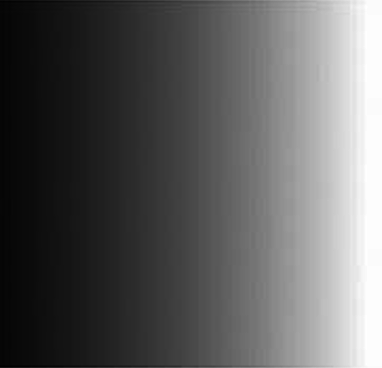
对比 输入图像，污染图像、较准图像理解gamma变换的实际效果

3 实验结果：

原图像

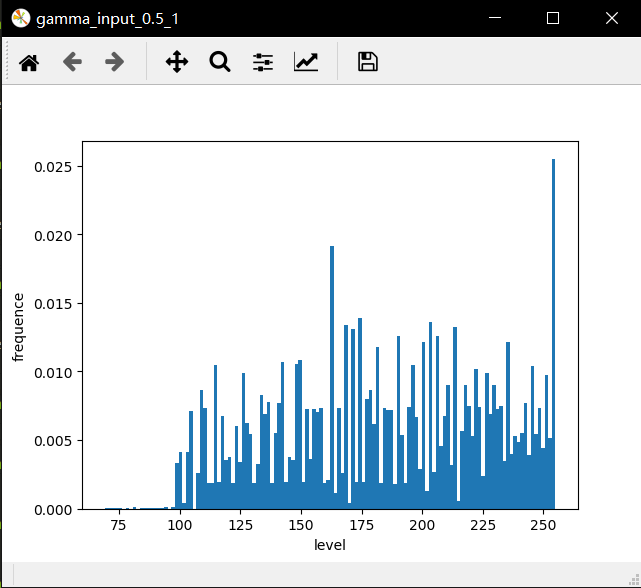
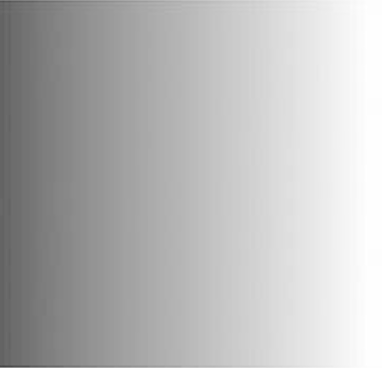


Step1(和2中所述相对应) gamma =2 c=1



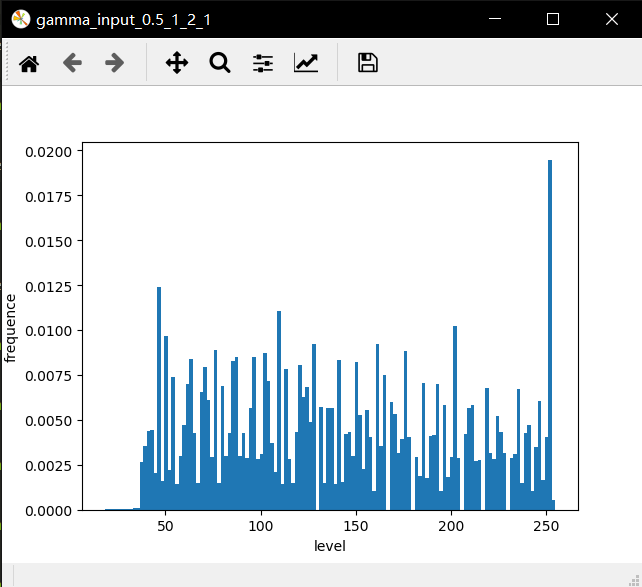
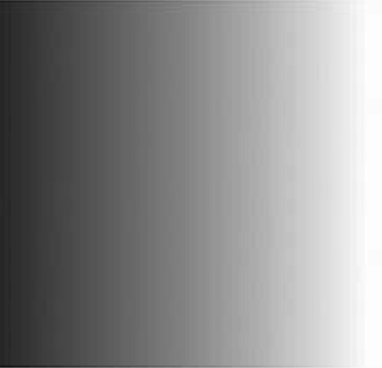
可以看到 gamma大于1是黑色像素压缩

Step2 (input 为原图像 ouput为中间图像) gamma=0.5 c=1



可以看到 gamma小于1时，白色像素压缩

Setp3 (input 为 step中的输出图像，输出为较准图像)



可以看到经过变换再进行反变换能很好地恢复原图像，通过这种方式模拟了通过gamma变换消除阴极射线管设备的影响

4 实验结果

Gamma 大于1时将压缩暗像素值, 扩展亮像素值

Gamma 小于1时扩展暗像素值, 压 缩亮像素值

5代码

**import** numpy **as** np  
**import** cv2 **as** cv  
**import** matplotlib.pyplot **as** plt  
**import** os  
**def** gamma\_transformation(input,gamma,c):  
 *'''* **:param** *input: 输入图片为256灰度图* **:param** *gamma: gamma校正的参数* **:param** *c: gamma校正的常数* **:return***:  
 '''* input = np.array(input,dtype=np.float32)  
 *#对输入进行归一化归到（0，1）* input = input/255 *# 映射到[0,1]上* input = np.power(input,gamma) *#计算 r^gamma* input = c\* input *#计算 c\*r^gamma  
 #反映射* output = input\*(255/c)  
 output = np.array(output,dtype= np.int32)  
 **return** output  
  
**def** plot\_hism(input,name):  
 *'''  
 统计输入的直方图并打印，输出的直方图不进行归一化  
 '''* input = np.array(input,dtype= np.int32)  
 input = input.flatten()  
 plt.figure(name)  
 plt.xlabel(**'level'**)  
 plt.ylabel(**"frequence"**)  
 n,bins,patches = plt.hist(input,bins=128,density=1)  
 plt.show()  
  
**if** \_\_name\_\_ ==**'\_\_main\_\_'**:  
 BASE\_PATH = os.path.dirname(os.path.abspath(\_\_file\_\_))  
 RESULT\_PATH = os.path.join(BASE\_PATH,**"result"**)  
 **if not** os.path.exists(RESULT\_PATH):  
 os.mkdir(RESULT\_PATH)  
 **import** argparse  
 parser = argparse.ArgumentParser()  
 parser.add\_argument(**'--img'**,help=**"the path of the input img"**)  
 parser.add\_argument(**'--gamma'**,help=**'the param of gamma transformation'**)  
 parser.add\_argument(**'--c'**, help=**'the param of gamma transformation'**)  
 args = parser.parse\_args()  
 *#获取图片名称* img\_name = os.path.split(args.img)[-1]  
 img\_name, ext = os.path.splitext(img\_name)  
 input = cv.imread(args.img,flags=cv.IMREAD\_GRAYSCALE)  
 plot\_hism(input, img\_name+**"\_before\_gamma"**)  
 output = gamma\_transformation(input,float(args.gamma),float(args.c))  
 img\_name = img\_name+**"\_"**+str(args.gamma)+**"\_"**+str(args.c)  
 plot\_hism(output,img\_name) *# 打印直方图* img\_name = img\_name+ext  
 img\_name = os.path.join(RESULT\_PATH,img\_name)  
 cv.imwrite(img\_name,output)