

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： $s = cr^{\text{gamma}}$

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 $\text{gamma}=2, c=1$ 的 gamma 变换，得到污染图像

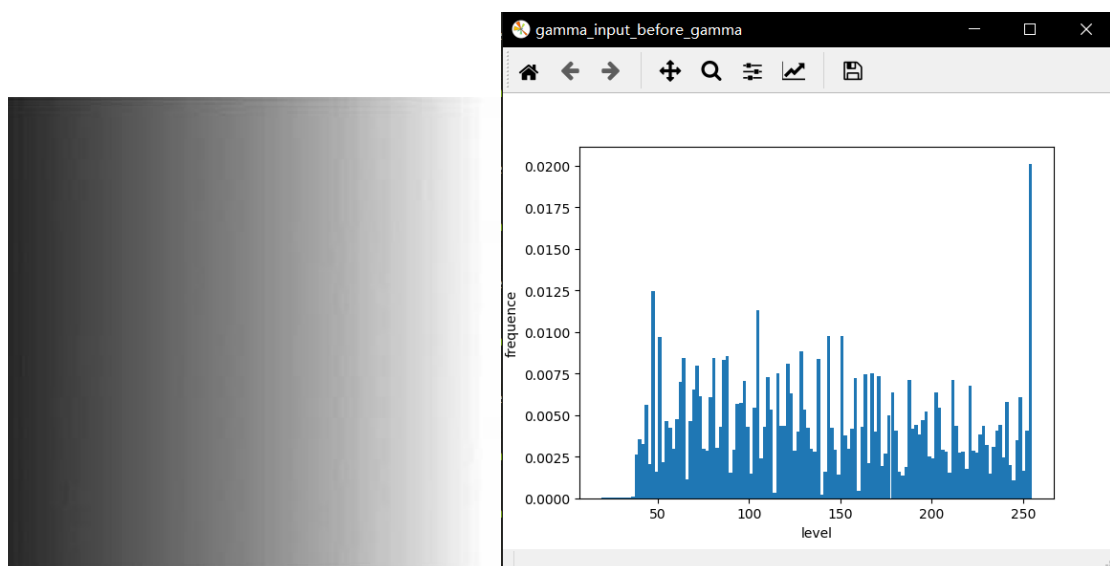
2 对输入图像进行 $\text{gamma}=0.5, c=1$ 的 gamma 变换得到中间图像

3 对中间图像进行 $\text{gamma}=2, c=1$ 的 gamma 变换得到校准图像

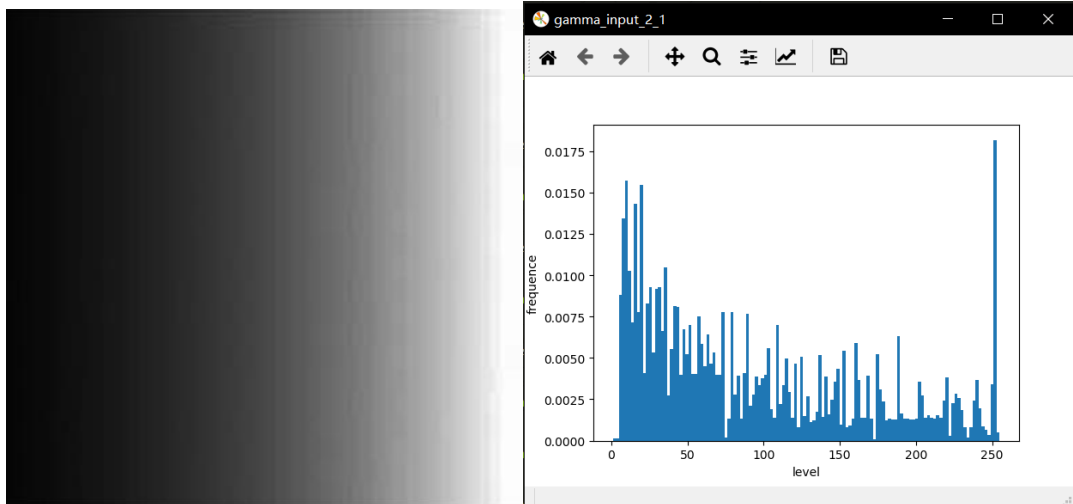
对比 输入图像，污染图像、校准图像理解 gamma 变换的实际效果

3 实验结果：

原图像

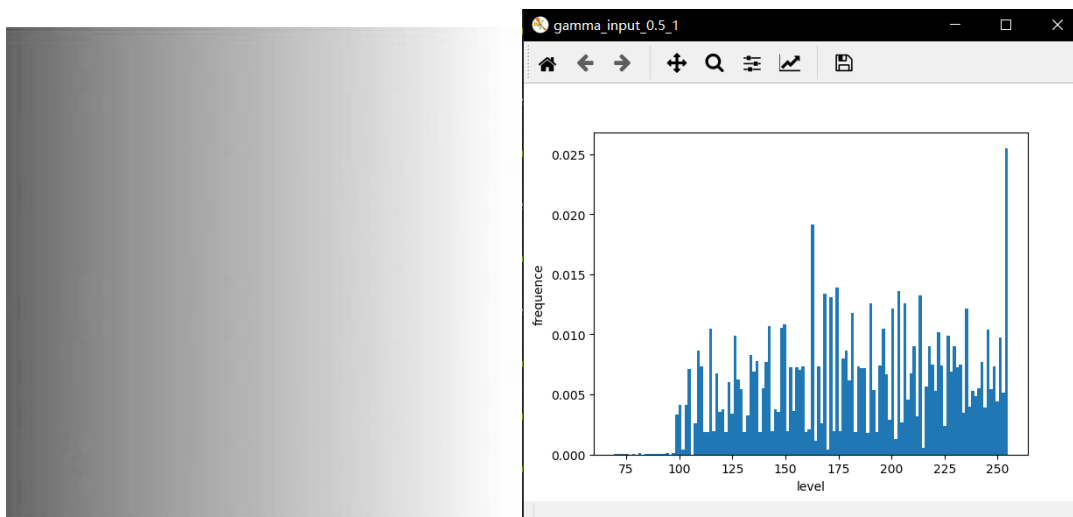


Step1(和 2 中所述相对应) $\text{gamma}=2, c=1$



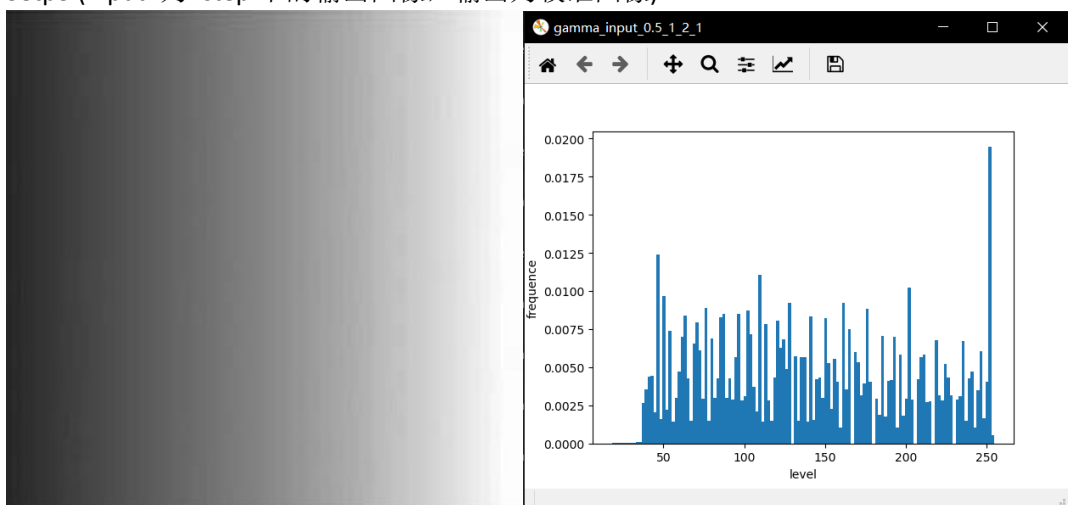
可以看到 γ 大于 1 是黑色像素压缩

Step2 (input 为原图像 output 为中间图像) $\gamma=0.5$ $c=1$



可以看到 γ 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 γ 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("frequency")
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)

```