桂林电子科技大学

**实验1 直方图均匀化**  实验报告

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 实验名称 | **直方图均匀化** | | | | | | | |  | 辅导员意见：  成绩 辅导员  签 名 |
| 院 系 | 计算机与信息安全学院 | | | 专业 | | 计算机科学与技术 | | |
| 学 号 | 2000301320 | | | 姓名 | | 靳志凌 | | |
| 实验日期 | 2022 | 年 | 10 | | 月 | | 14 | 日 |
|  |  | | | | | | | |

## 一、实验目的

掌握直方图均匀化算法的基本原理。

## 二、实验内容与要求

1. 实现图像直方图均匀化的算法。
2. 读取MATLAB自带图像，对该图像进行直方图均衡化，显示处理前后的图像及直方图。
3. 调出MATLAB自带直方图均衡化函数，与你编写代码的效果进行比较。
4. 提供良好交互界面。

## 三、实验环境

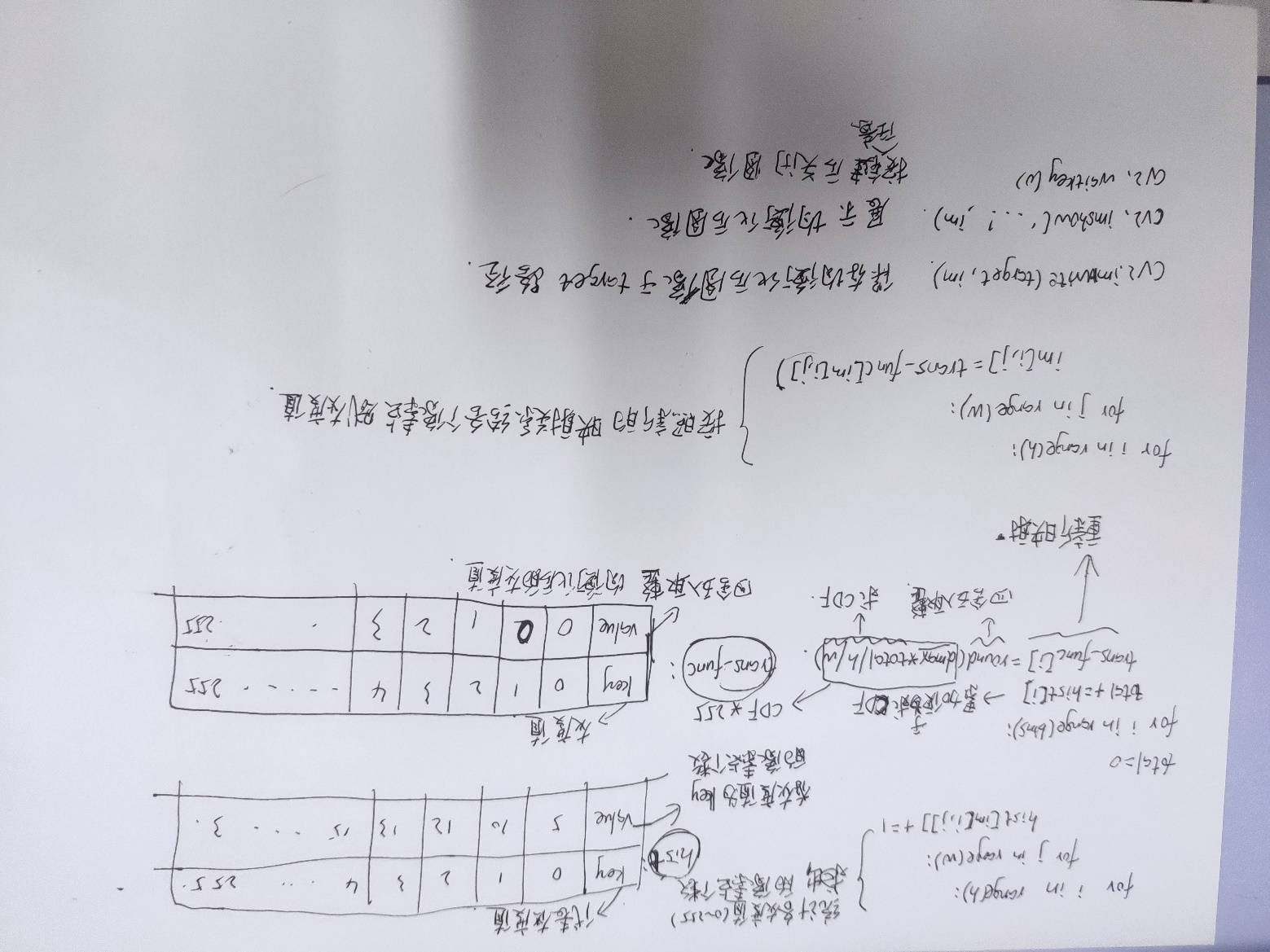
在实验室进行实验，学生可选Matlab、Python等等语言实现。

## 四、实验步骤

1.该实验主要涉及两个自写函数：GrayHist(path,hist\_save,bins),Histeq(path,target,bins)。

在GrayHist(path,hist\_save,bins)函数中，实现将存储在path中的灰度图片的直方图通过plt库显示到屏幕，并将直方图存储到hist\_save路径，bins参数用来决定直方图中小柱子的个数。

Histeq(path,target,bins)该函数用来将path路径上的图片均衡化之后存储到target路径，bins是该图片的灰度级数。均衡化的具体过程如下：

首先将原图片以灰度图的方式读出，并获取该图的像素点的行数列数。然后构造出两个字典，两个字典的key均为从0到255的灰度值。先求出该图片的PDF，以字典的形式保存在hist变量中，然后累加求出CDF，并将CDF\*255后四舍五入取整，将灰度值重新映射到原图像。

记录每个实验的代码（代码必须以文字形式粘贴，白底黑字，不得使用截图方式、不得使用语法高亮）、运行截图（截图中能够包含自己的学号和姓名）

实验代码：

import cv2

import numpy as np

import matplotlib.pyplot as plt

# 显示灰度图的直方图,并存储直方图

def GrayHist(path,hist\_save='Images\a.png',bins = 32):

gray\_image = cv2.imread(path,0)

h,w = gray\_image.shape

data = gray\_image.reshape((h\*w)) #转化为一维数组

plt.hist(data,density=True,stacked=True,bins=bins)

plt.savefig(hist\_save)

plt.show()

# 线性变换

def LinearTrans(path,a =1,b =-50):

"""

返回值：处理好的二维数组

"""

im = cv2.imread(path,0)

imarray = np.array(im)

h,w = im.shape

print(im.shape,imarray.shape,sep='\n')

for i in range(h):

for j in range(w):

aft = int(a\*im[i,j]+b)

if(aft<=255 and aft >=0):

im[i][j] = aft

elif aft>255:

im[i][j] = 255

elif aft<0:

im[i][j] = 0

# 指数对数变换

def ExpTrans(path,esp = 0,gama = 1):

"""

返回值：处理好的二维数组

"""

im = cv2.imread(path,0)

h,w = im.shape

for i in range(h):

for j in range(w):

tmp = im[i,j]/255

tmp = int(pow(tmp+esp,gama)\*255)

if tmp >= 0 and tmp <= 255:

im[i][j] = tmp

elif tmp < 0:

im[i][j] = 0

else:

tmp = 255

# 均衡化图像，并显示存储图像

def Histeq(path,target,bins=256):

dmax = 255

im = cv2.imread(path,0)

h,w = im.shape

trans\_func = dict(zip([i for i in range(bins)],[0 for i in range(bins)]))

hist = dict(zip([i for i in range(bins)],[0 for i in range(bins)]))

for i in range(h):

for j in range(w):

hist[im[i,j]] +=1

total = 0

for i in range(bins):

total += hist[i]

trans\_func[i] = round(dmax\*total/h/w)

for i in range(h):

for j in range(w):

im[i,j] = trans\_func[im[i,j]]

cv2.imwrite(target,im)

cv2.imshow('2000301320-jinzhiling\_histeq',im)

cv2.waitKey(0)

origin\_image = 'Images\girl.jpg'

gray\_image = 'Images\gray\_image.jpg'

histeq\_image = 'Images\histeq\_image.jpg'

origin\_hist = 'Images\origin\_hist.png'

equalize\_hist = 'Images\equalize\_hist.png'

origin = cv2.imread(origin\_image,0)

cv2.imshow('origin\_gray',origin)

cv2.waitKey(0)

GrayHist(origin\_image,origin\_hist,256)

Histeq(origin\_image,histeq\_image,256)

GrayHist(histeq\_image,equalize\_hist,256)

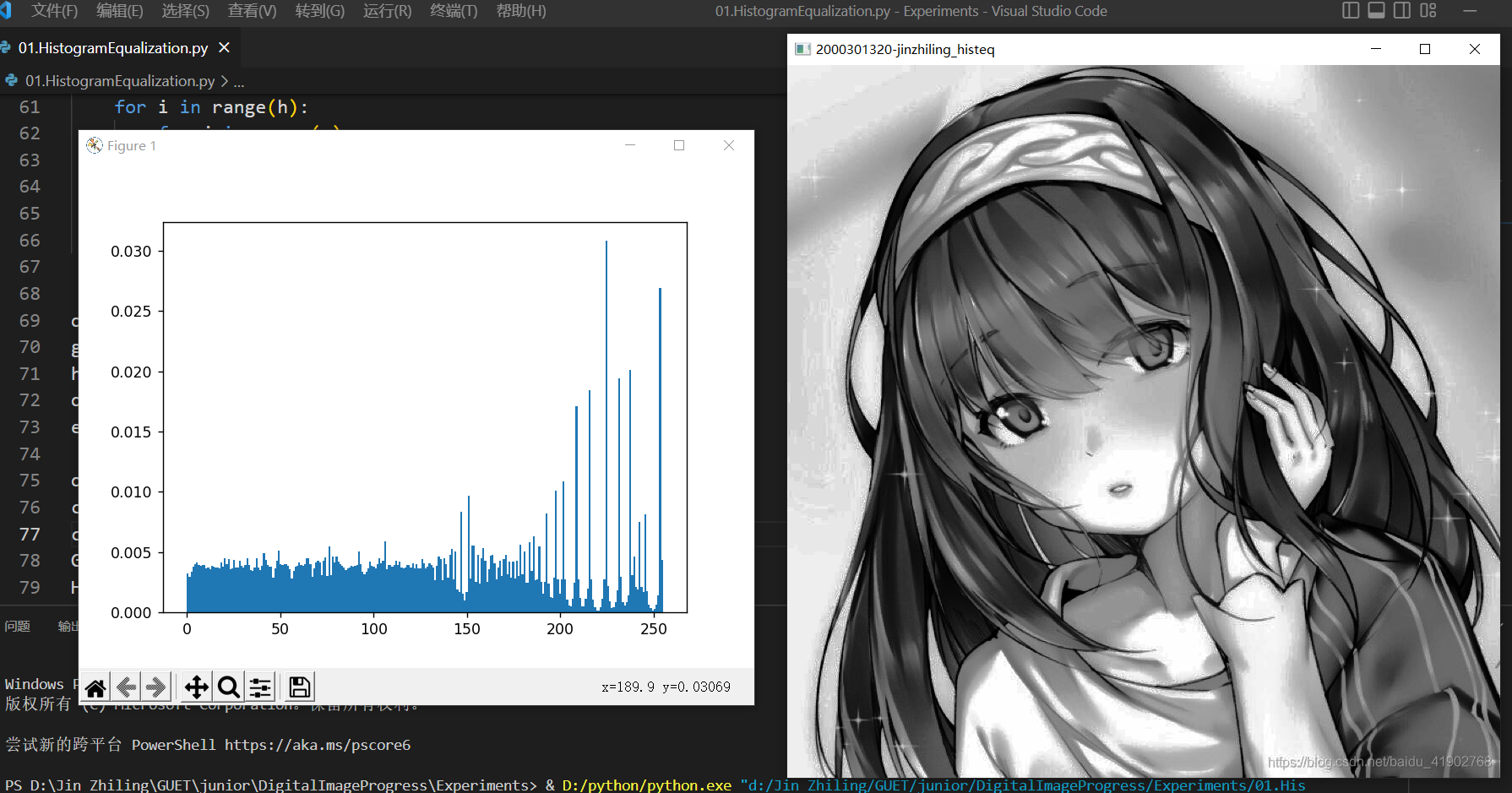
im2=cv2.equalizeHist(cv2.imread(origin\_image,0))

cv2.imshow('opencv',im2)

cv2.waitKey(0)

实验图片：





下图是opencv库自带的均衡化函数和自己编写的均衡化函数处理图像后进行对比：



## 五、问题记录和实验总结（必写）

通过直方图均衡化的实验学习，我学习到图像均衡化的原理，以及直方图均衡化的适用场景。直方图均衡化主要用来处理一些整体偏亮或偏暗的图像。但直方图均衡化是一种全局处理的方式，对于处理的数据不加选择，可能会增加背景干扰信息的对比度还可能降低有用信号的对比度。而且，均衡化后的图像的灰度级减少，图像中的某些细节将会消失。