**Contents**

Question 1 -------------------------------------------------------------------------------------------- 2

1. Image of code --------------------------------------------------------------------------------- 2

2. Code ---------------------------------------------------------------------------------------------- 2

3. Test ----------------------------------------------------------------------------------------------- 3

Question 2 -------------------------------------------------------------------------------------------- 3

1. Image of code --------------------------------------------------------------------------------- 4

2. Code ---------------------------------------------------------------------------------------------- 4

3. Test ----------------------------------------------------------------------------------------------- 5

4. Conclusion -------------------------------------------------------------------------------------- 7

Question 3 -------------------------------------------------------------------------------------------- 8

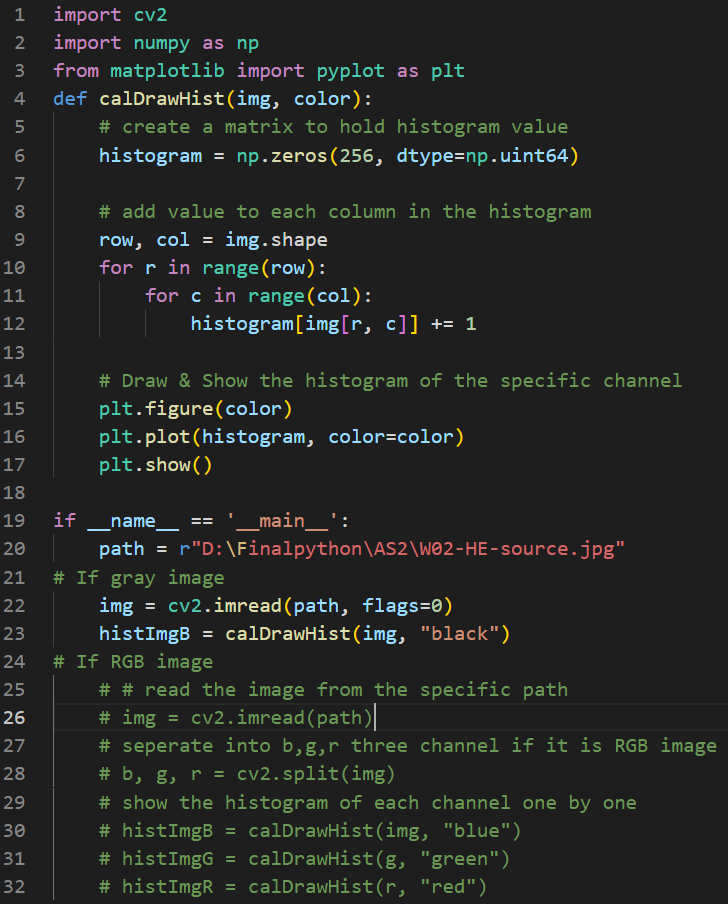
1. Image of code --------------------------------------------------------------------------------- 8

2. Code ---------------------------------------------------------------------------------------------- 8

3. Test ----------------------------------------------------------------------------------------------- 9

**Question 1**

1. **Image of code for the better reading experience**



1. **Code details**

import cv2

import numpy as np

from matplotlib import pyplot as plt

def calDrawHist(img, color):

# create a matrix to hold histogram value

histogram = np.zeros(256, dtype=np.uint64)

# add value to each column in the histogram

row, col = img.shape

for r in range(row):

for c in range(col):

histogram[img[r, c]] += 1

# Draw & Show the histogram of the specific channel

plt.figure(color)

plt.plot(histogram, color=color)

plt.show()

if \_\_name\_\_ == '\_\_main\_\_':

path = r"D:\Finalpython\AS2\W02-HE-source.jpg"

# If gray image

img = cv2.imread(path, flags=0)

histImgB = calDrawHist(img, "black")

# If RGB image

# # read the image from the specific path

# img = cv2.imread(path)

# seperate into b,g,r three channel if it is RGB image

# b, g, r = cv2.split(img)

# show the histogram of each channel one by one

# histImgB = calDrawHist(img, "blue")

# histImgG = calDrawHist(g, "green")

# histImgR = calDrawHist(r, "red")

1. **Test**

Use the image named “W02-HE-source.jpg” as an example to test the code above.

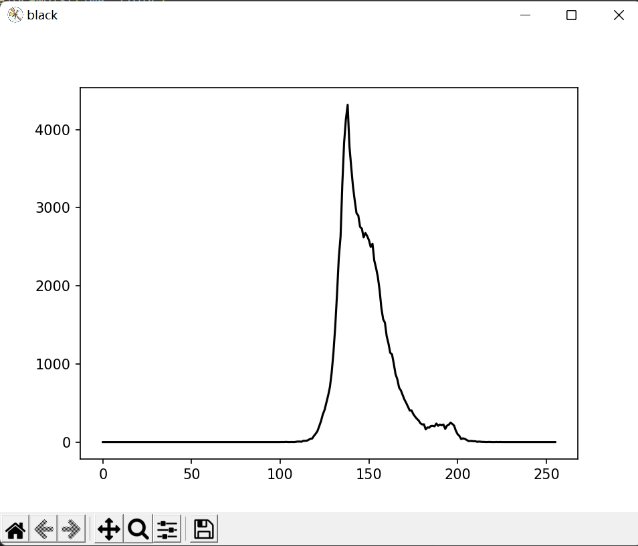
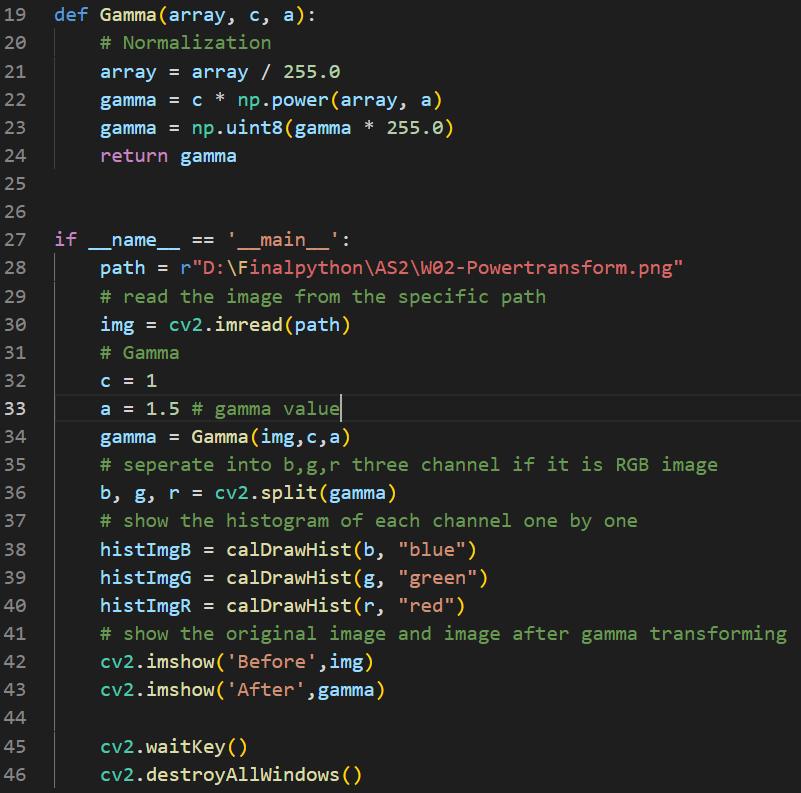


Figure 1: Histogram of gray image

**Question 2**

1. **Image of code for the better reading experience**

Because the function of calculate, draw and show the histogram of image has shown in the Question 1, the code below will only include the function of gamma and main function used to test.



1. **Code details**

The code here will only include the function of gamma transformation and main function as well.

def Gamma(array, c, a):

# Normalization

array = array / 255.0

gamma = c \* np.power(array, a)

gamma = np.uint8(gamma \* 255.0)

return gamma

if \_\_name\_\_ == '\_\_main\_\_':

path = r"D:\Finalpython\AS2\W02-Powertransform.png"

# read the image from the specific path

img = cv2.imread(path)

# Gamma

c = 1

a = 1.5 # gamma value

gamma = Gamma(img,c,a)

# seperate into b,g,r three channel if it is RGB image

b, g, r = cv2.split(gamma)

# show the histogram of each channel one by one

histImgB = calDrawHist(b, "blue")

histImgG = calDrawHist(g, "green")

histImgR = calDrawHist(r, "red")

# show the original image and image after gamma transforming

cv2.imshow('Before',img)

cv2.imshow('After',gamma)

cv2.waitKey()

cv2.destroyAllWindows()

1. **Test**

Because this is a **RGB** image, there are three histograms for each three channels.

Besides, since the image is quite bright, I choose to use the **value γ** that is **greater than 1**. Hence, the values used are **1.5, 2.5 and 5**, which are also from the lecture PowerPoint.

1. Before the power-law (Gamma) transform, the histograms are,

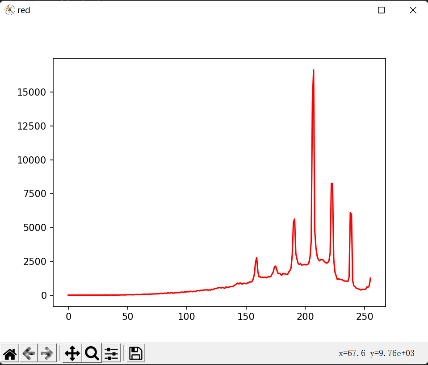
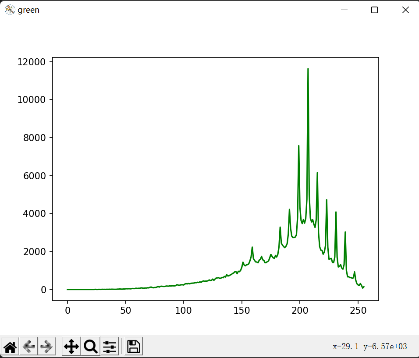
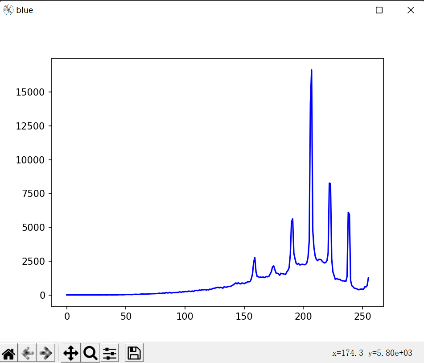


Figure 2: Image histograms of blue, green and red channels

1. After the power-law (Gamma) transform, the histograms and both before and after images are,
2. When γ=1.5,

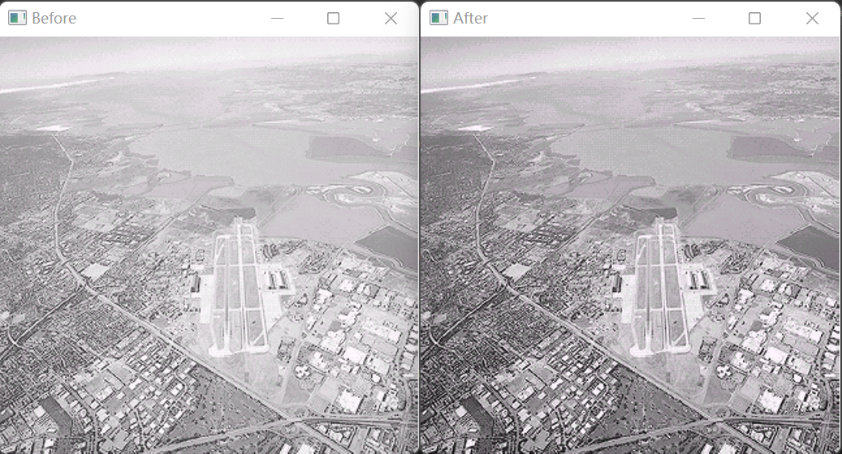


Figure 3: Before and after image when γ=1.5

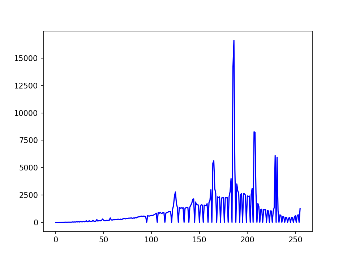
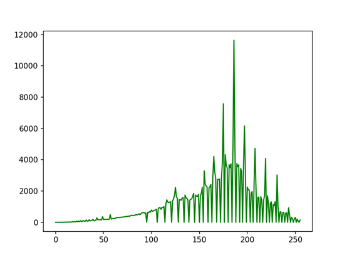
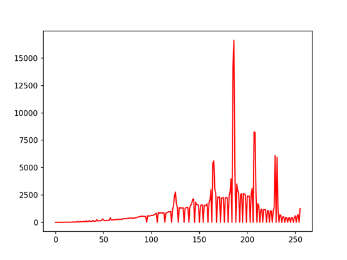
  

Figure 4: Image histograms of blue, green and red channels

1. When γ=2.5

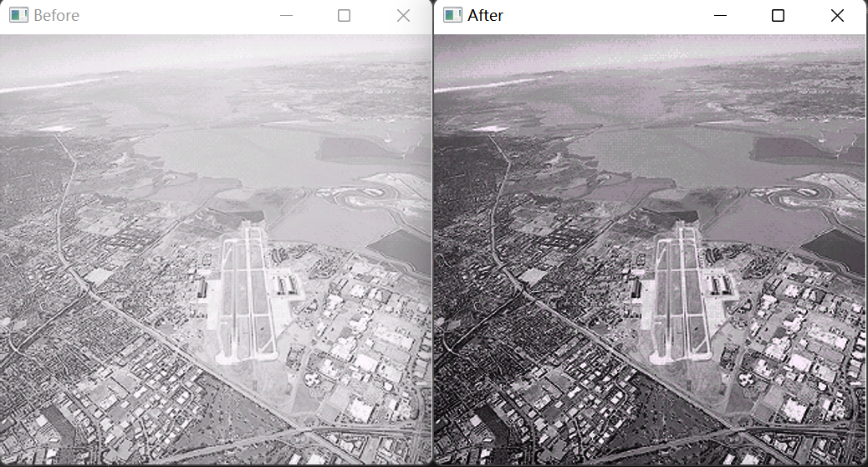


Figure 5: Before and after image when γ=2.5

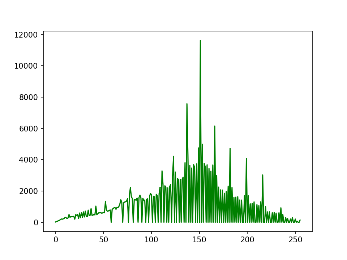
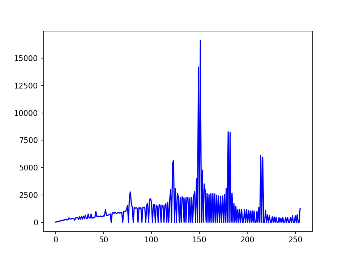
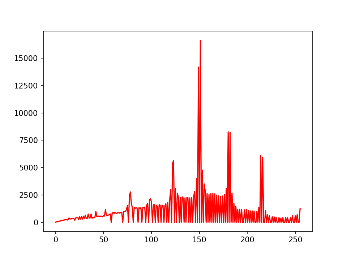
 

Figure 6: Image histograms of blue, green and red channels

1. When γ=5,

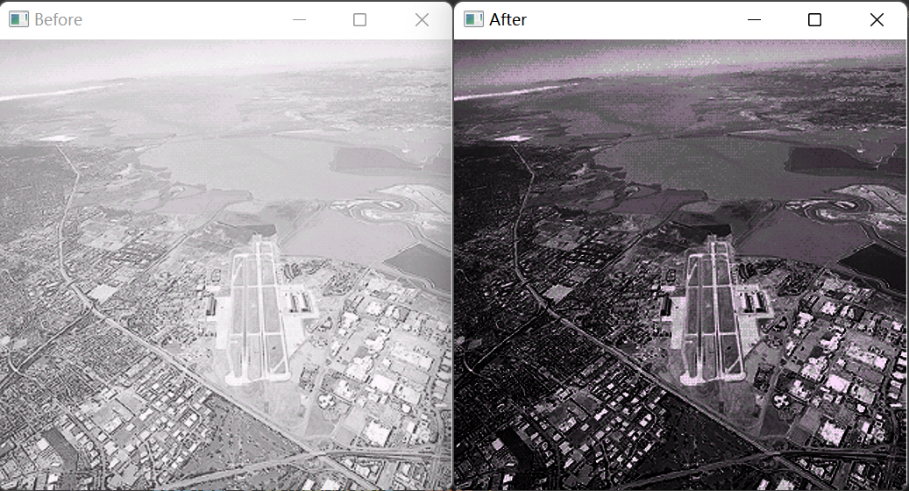


Figure 7: Before and after image when γ=5

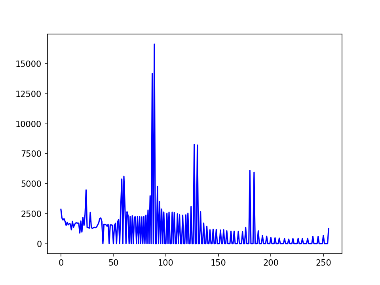
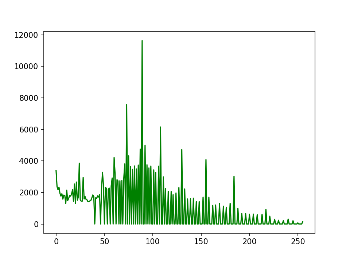
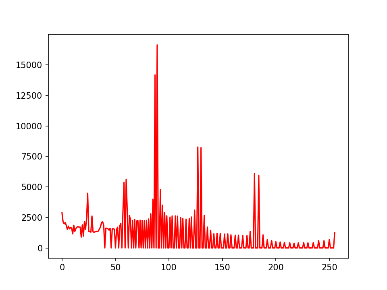
  

Figure 8: Image histograms of blue, green and red channels

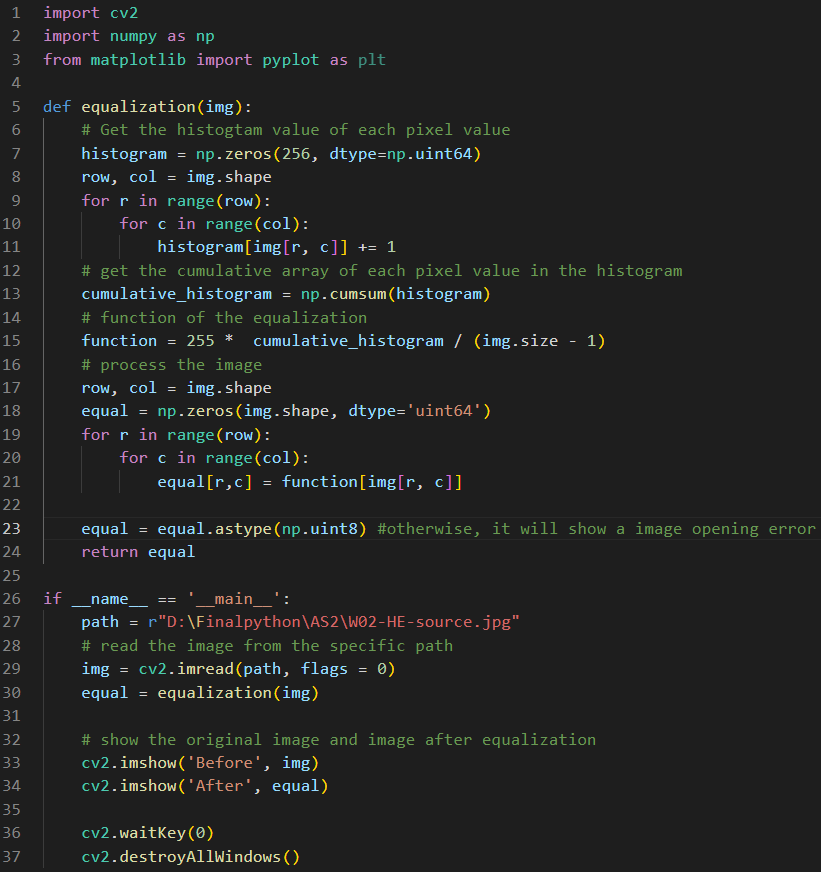
1. **Conclusion**

Obviously, the optimum value of γ is **2.5**, because the processed image has moderate light and shade, more details can be seen.

**Question 3**

1. **Image of code for the better reading experience**

Complete code for complement equalization



1. **Code details**

import cv2

import numpy as np

from matplotlib import pyplot as plt

def equalization(img):

# Get the histogtam value of each pixel value

histogram = np.zeros(256, dtype=np.uint64)

row, col = img.shape

for r in range(row):

for c in range(col):

histogram[img[r, c]] += 1

# get the cumulative array of each pixel value in the histogram

cumulative\_histogram = np.cumsum(histogram)

# function of the equalization

function = 255 \* cumulative\_histogram / (img.size - 1)

# process the image

row, col = img.shape

equal = np.zeros(img.shape, dtype='uint64')

for r in range(row):

for c in range(col):

equal[r,c] = function[img[r, c]]

equal = equal.astype(np.uint8) #otherwise, it will show a image opening error

return equal

if \_\_name\_\_ == '\_\_main\_\_':

path = r"D:\Finalpython\AS2\W02-HE-source.jpg"

# read the image from the specific path

img = cv2.imread(path, flags = 0)

equal = equalization(img)

# show the original image and image after equalization

cv2.imshow('Before', img)

cv2.imshow('After', equal)

cv2.waitKey(0)

cv2.destroyAllWindows()

1. **Test**
2. Histograms of image before and after equalization

Since the image is a grayscale image, the histogram of each will be one.

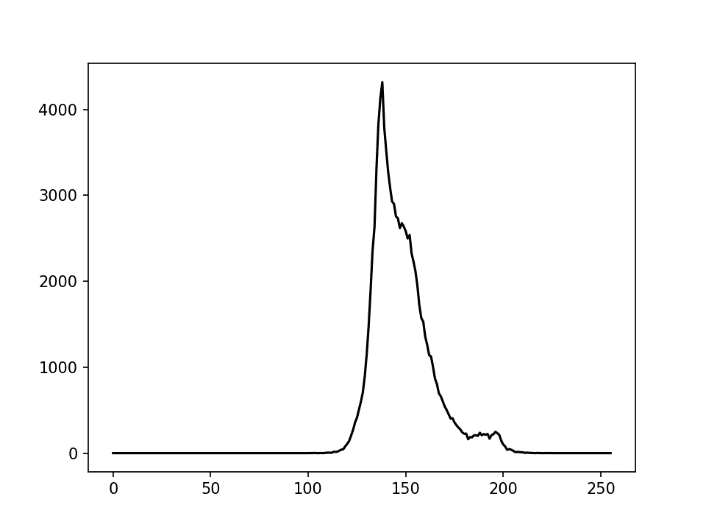


Figure 9: Histogram of image before equalization

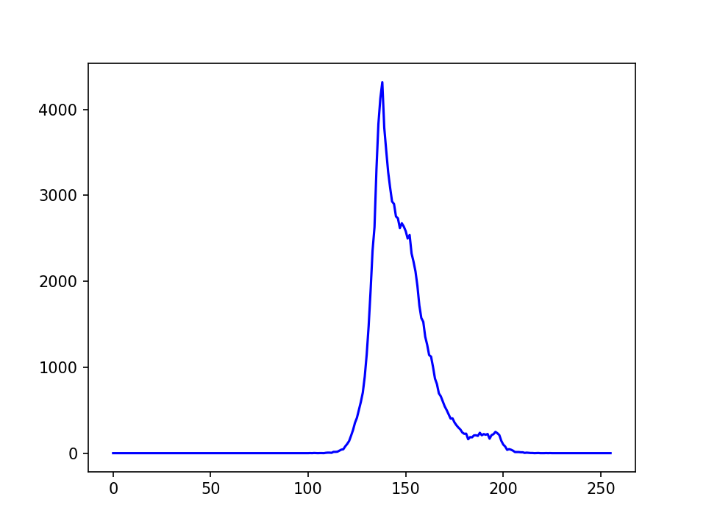


Figure 10: Histogram of image after equalization

1. Images before and after equalization



Figure 11: Images before and after equalization