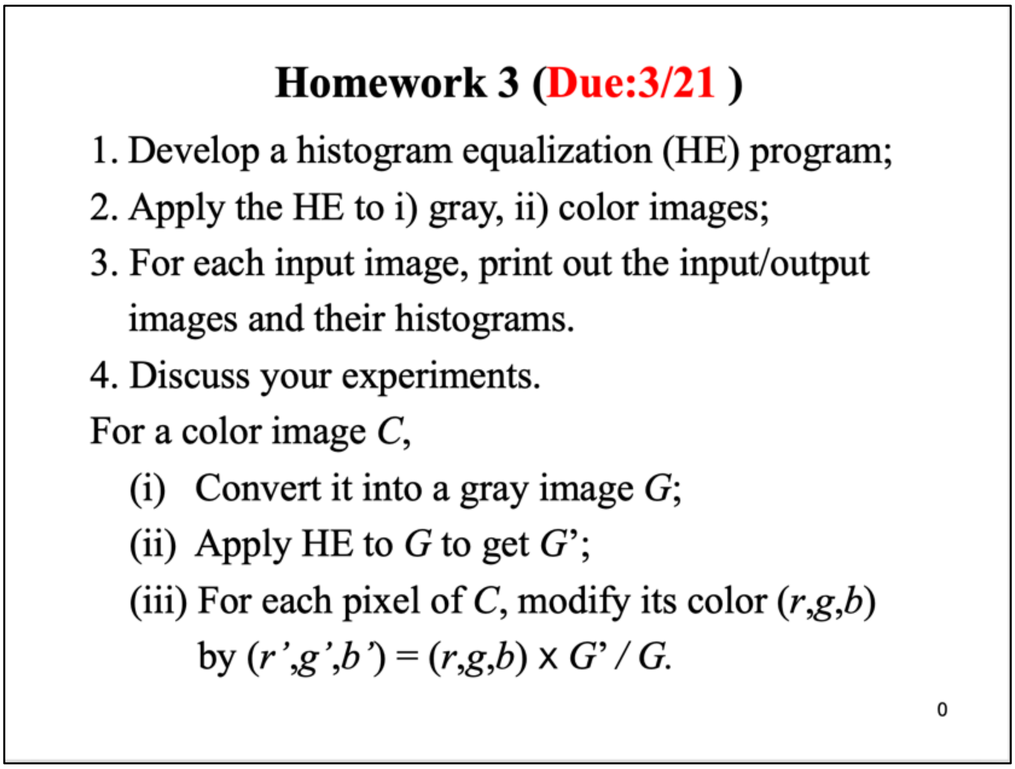
41047011S 資工 114 趙安庭

(1) Assignment statement



Homework 3 (Due:3/21)

1. Develop a histogram equalization (HE) program;
2. Apply the HE to i) gray, ii) color images;
3. For each input image, print out the input/output images and their histograms.
4. Discuss your experiments.

For a color image *C*,

(i) Convert it into a gray image *G*;

(ii) Apply HE to *G* to get *G*’;

(iii) For each pixel of *C*, modify its color (r,g,b) by $(r',g',b') = (r,g,b) \times G' / G$.

0

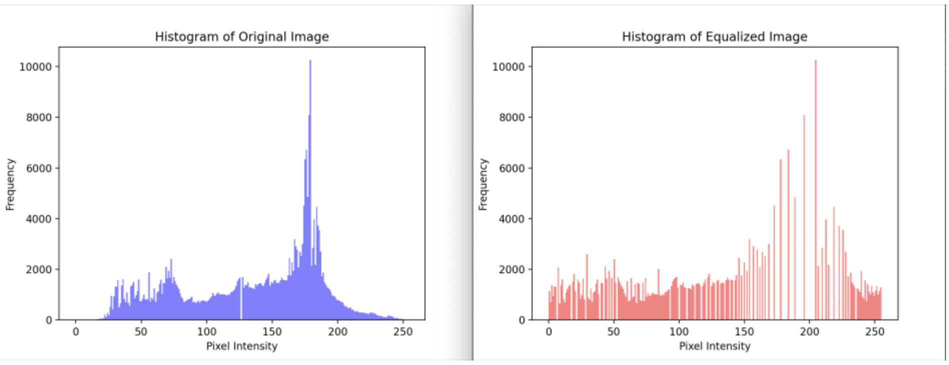
(2)

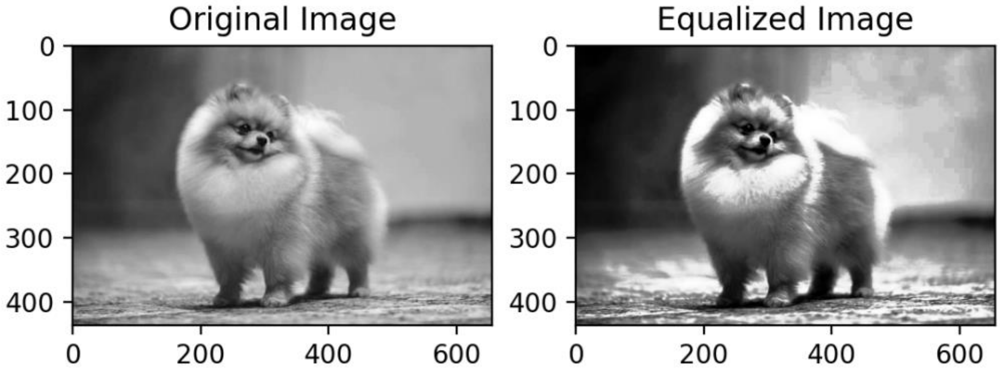
(a) Input/output images

(a-i) gray image

Input:

Output:

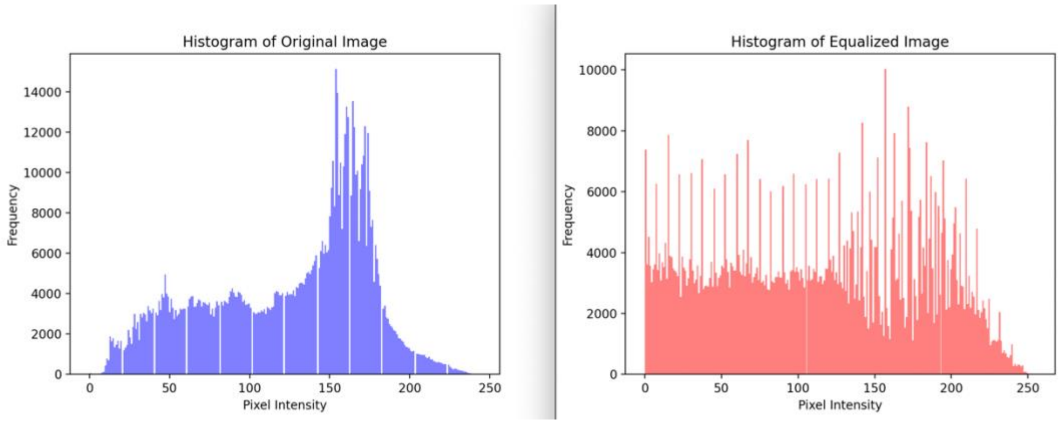


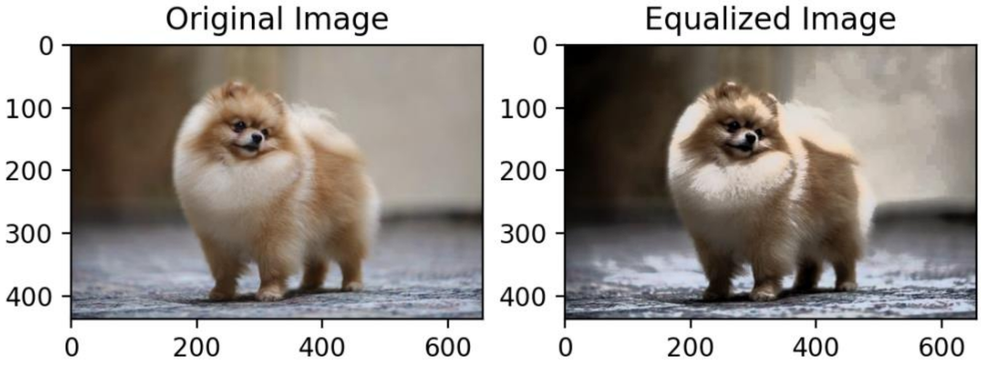


(a-ii) color image

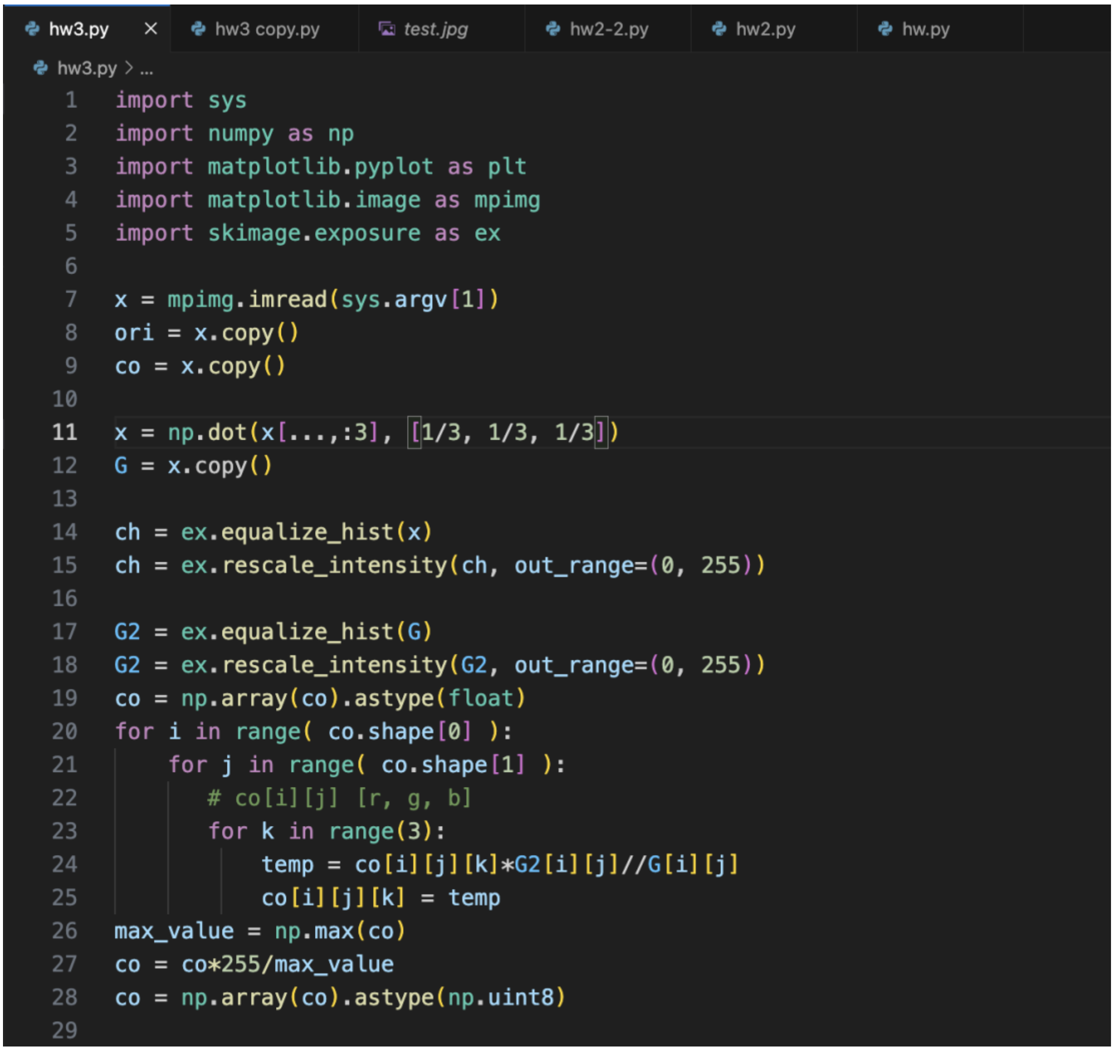
Input:

Output:

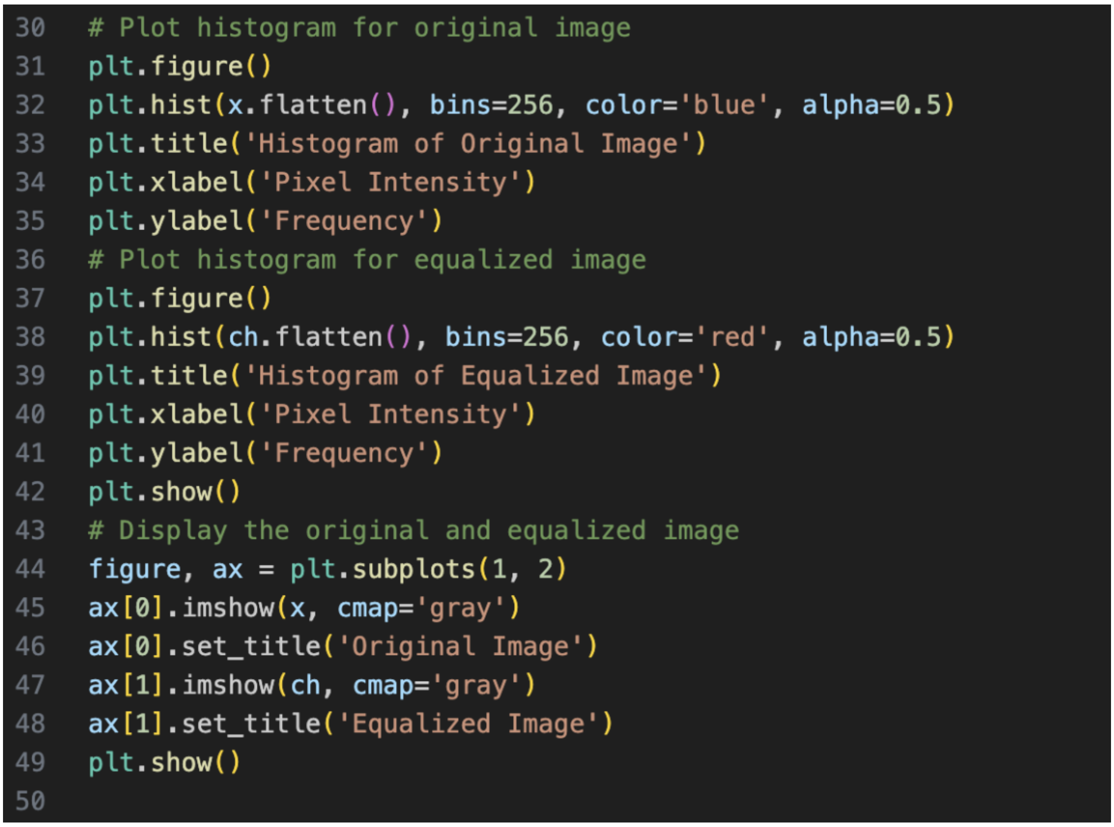




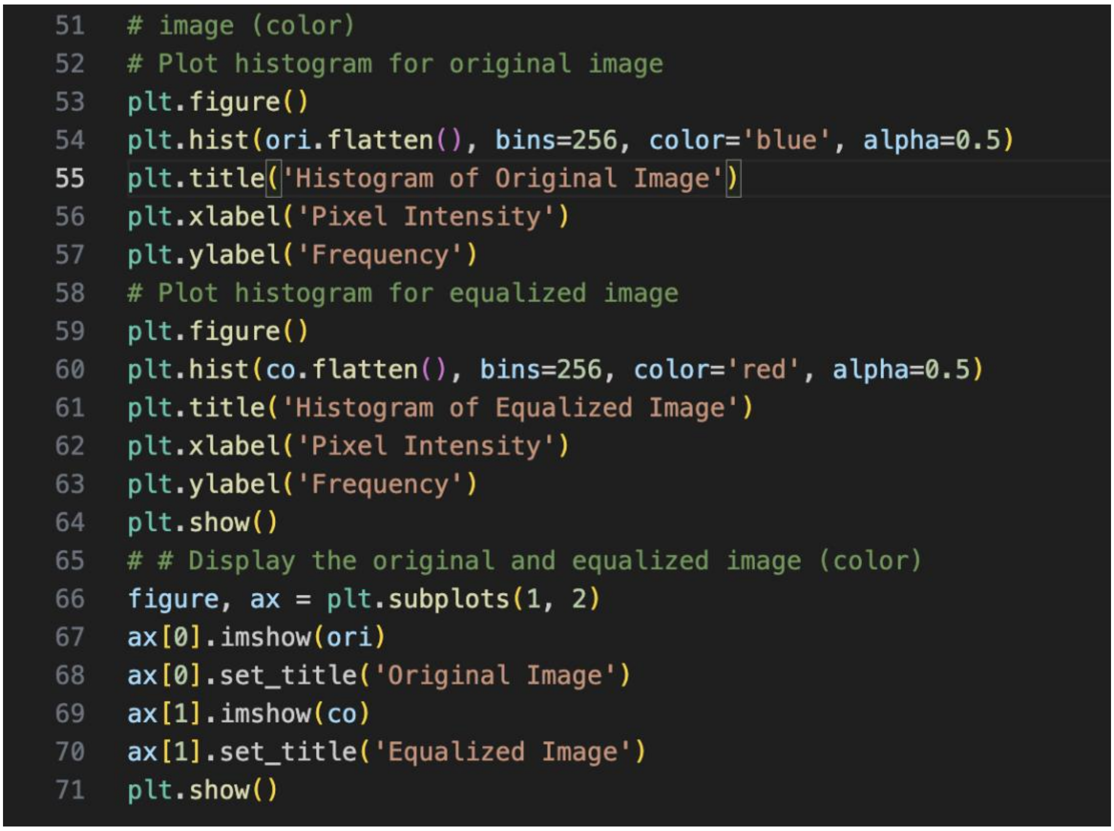
(b) Source code



```python
import sys
import numpy as np
import matplotlib.pyplot as plt
import matplotlib.image as mpimg
import skimage.exposure as ex

x = mpimg.imread(sys.argv[1])
ori = x.copy()
co = x.copy()

x = np.dot(x[...,:3], [1/3, 1/3, 1/3])
G = x.copy()

ch = ex.equalize_hist(x)
ch = ex.rescale_intensity(ch, out_range=(0, 255))

G2 = ex.equalize_hist(G)
G2 = ex.rescale_intensity(G2, out_range=(0, 255))
co = np.array(co).astype(float)
for i in range( co.shape[0] ):
    for j in range( co.shape[1] ):
        # co[i][j] [r, g, b]
        for k in range(3):
            temp = co[i][j][k]*G2[i][j]//G[i][j]
            co[i][j][k] = temp
max_value = np.max(co)
co = co*255/max_value
co = np.array(co).astype(np.uint8)

```



```python
# Plot histogram for original image
plt.figure()
plt.hist(x.flatten(), bins=256, color='blue', alpha=0.5)
plt.title('Histogram of Original Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
# Plot histogram for equalized image
plt.figure()
plt.hist(ch.flatten(), bins=256, color='red', alpha=0.5)
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
# Display the original and equalized image
figure, ax = plt.subplots(1, 2)
ax[0].imshow(x, cmap='gray')
ax[0].set_title('Original Image')
ax[1].imshow(ch, cmap='gray')
ax[1].set_title('Equalized Image')
plt.show()

```



```python
# image (color)
# Plot histogram for original image
plt.figure()
plt.hist(ori.flatten(), bins=256, color='blue', alpha=0.5)
plt.title('Histogram of Original Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
# Plot histogram for equalized image
plt.figure()
plt.hist(co.flatten(), bins=256, color='red', alpha=0.5)
plt.title('Histogram of Equalized Image')
plt.xlabel('Pixel Intensity')
plt.ylabel('Frequency')
plt.show()
# # Display the original and equalized image (color)
figure, ax = plt.subplots(1, 2)
ax[0].imshow(ori)
ax[0].set_title('Original Image')
ax[1].imshow(co)
ax[1].set_title('Equalized Image')
plt.show()
```

(c) Comments

處理彩色圖像的時候，有一度我做出來的輸出圖片會有一塊一塊青色、黃色（如下圖），後來發現應是程式碼第 24 行的計算會讓某些像素的值超過 255，因此我改為先將原圖轉型為 float，迴圈計算完第 24 行公式以後，在迴圈外面我再將整張圖的值乘以 255、除以圖片最大值，確保圖片的值都落在 0 到 255 之間，如此便解決了有色塊的問題。