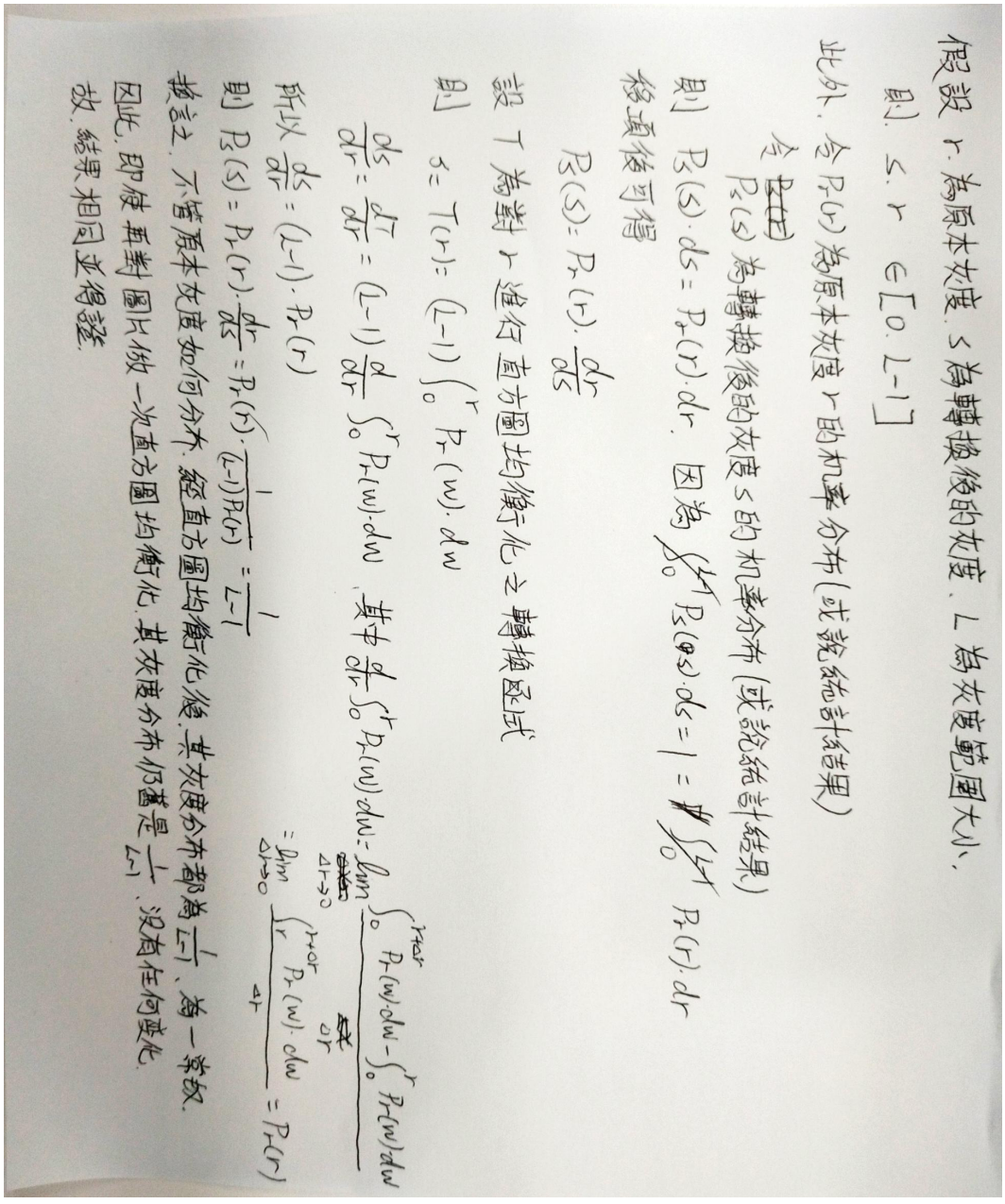
學號：D12922014

姓名：莊謹譽

1. Suppose that a digital image is subjected to histogram equalization. Show that a second pass of histogram equalization (on the histogram-equalized image) will produce exactly the same result as the first pass.

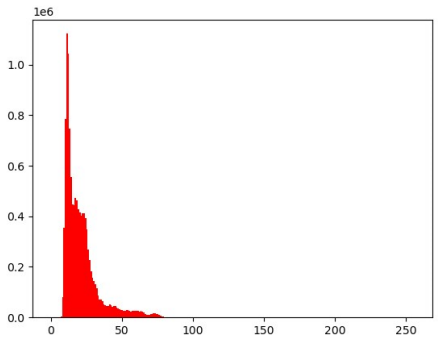
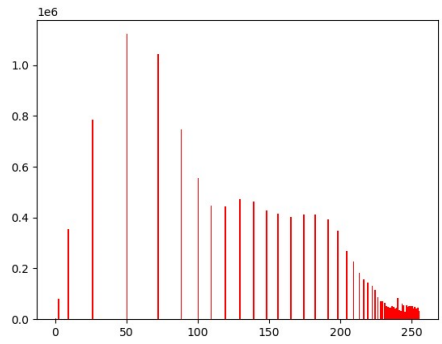
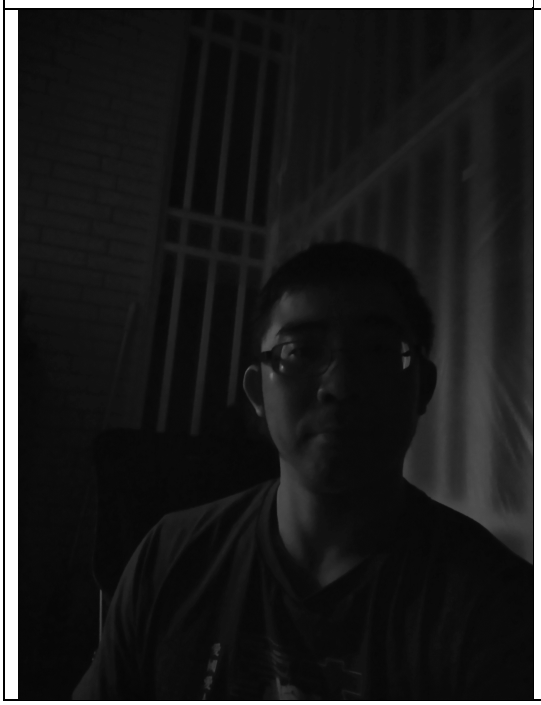


(實際執行程式碼也可得到相同結果，可參見附錄執行結果)

2. Write a program for histogram equalization, and test it with your own selfie took in a relatively dark environment so that we can clearly see

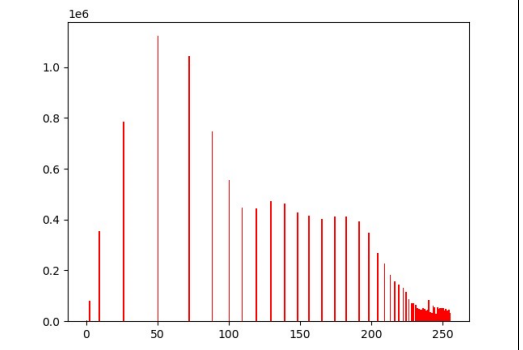
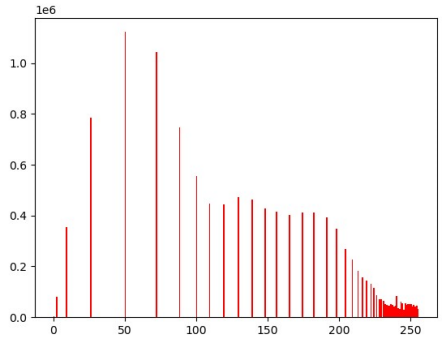
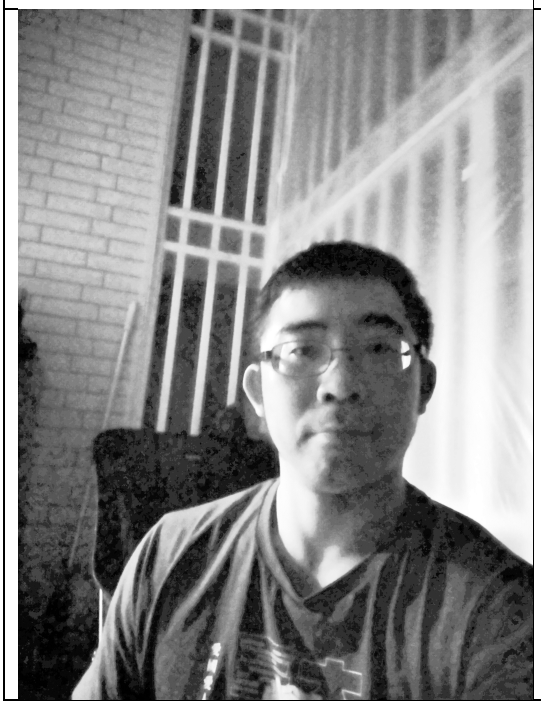
the effect of histogram equalization in image enhancement. Please show the histograms of your selfie before and after histogram equalization and explain your results. (Note: You only have to work on the gray scale image.)

從下方對比可以很明顯看到，在陰暗環境下拍攝的圖片，其灰度全集中在很低的範圍，這導致整張圖片看起來很暗，對比很差；而透過直方圖均衡化，其灰度得以擴展到高亮度，且盡可能均勻分布在不同範圍，因此整張圖片不僅變亮，對比也明顯起來，可以更好地識別出畫面中的細節。

執行環境:	使用套件
● Windows 10 專業版 22H2 ● Visual Studio Code - 1.83.1 ● Git for Windows - 2.42.0 ● Python - 3.12.0	● matplotlib - 3.8.0 ● opencv-python - 4.8.1.78
Before	**After**
	
	

附錄：

- 套用兩次直方圖均衡化的結果對比

After	After Twice
	
	

- 程式碼執行方式與說明

```
~/Doc/cod/ntu-csie-digital-image-processing/hw3 on hw3 !4 ?3
cd d12922014

~/Doc/cod/ntu-csie-digital-image-processing/hw3/d12922014 on hw3 !4 ?3
python code/hw3.py
```

```python
# 程式碼進入點
if __name__ == "__main__":
    # 以灰階形式讀取「原始圖片」
    src = cv2.imread(os.path.abspath(ORIGIN_JPG_PATH), cv2.IMREAD_GRAYSCALE)
    # 以 png 格式儲存「處理前」圖片，以供後續對比
    cv2.imwrite(os.path.abspath(Path(RESULTS_DIR_PATH, "before.png")), src)
    # 繪製、儲存並顯示「處理前」histogram
    fig, ax = plt.subplots()
    ax.hist(src.ravel(), 256, [0, 256],color='r')
    plt.savefig(os.path.abspath(Path(RESULTS_DIR_PATH, "before_histogram.png")))
    plt.show()

    # 對「原始圖片」進行 histogram equalization 處理
    dst = cv2.equalizeHist(src)
    # 以 png 格式儲存「處理後」圖片，以供後續對比
    cv2.imwrite(os.path.abspath(Path(RESULTS_DIR_PATH, "after.png")), dst)
    # 繪製、儲存並顯示「處理後」histogram
    fig, ax = plt.subplots()
    ax.hist(dst.ravel(), 256, [0, 256],color='r')
    plt.savefig(os.path.abspath(Path(RESULTS_DIR_PATH, "after_histogram.png")))
    plt.show()

    # 對「處理後」圖片再次進行 histogram equalization
    dst2 = cv2.equalizeHist(dst)
    # 以 png 格式儲存「二次處理後」圖片，以供後續對比
    cv2.imwrite(os.path.abspath(Path(RESULTS_DIR_PATH, "twice_after.png")), dst2)
    # 繪製、儲存並顯示「二次處理後」histogram
    fig, ax = plt.subplots()
    ax.hist(dst2.ravel(), 256, [0, 256],color='r')
    plt.savefig(os.path.abspath(Path(RESULTS_DIR_PATH, "twice_after_histogram.png")))
    plt.show()
```