

## I. Method

## A. Histogram Equalization

```

19 def histogram_equalization(img):
20     histogtam, bins = np.histogram(img.flatten(), 256, [0, 256])
21     cdf = histogtam.cumsum()
22     masked_cdf = np.ma.masked_equal(cdf, 0)
23     masked_cdf = (masked_cdf - masked_cdf.min()) * 255 / (masked_cdf.max() - masked_cdf.min())
24     masked_cdf = np.ma.filled(masked_cdf, 0).astype('uint8')
25     histogtam_equalized_img = masked_cdf[img]
26     show_histogram(img)
27     show_histogram(histogtam_equalized_img)
28     plt.subplot(121), plt.imshow(img, cmap='gray'), plt.title('original_img')
29     plt.subplot(122), plt.imshow(histogtam_equalized_img, cmap='gray'), plt.title('histogtam_equalized_img')
30     plt.show()
31     cv2.imwrite('Q1_histogtam_equalized_img.jpg', histogtam_equalized_img)

```

20: 利用 numpy 的函式計算圖片攤平後的長條圖

21: 計算 CDF 累積機率函數

22: 把機率為 0 的部分過濾掉

23: 將 CDF 標準化後縮放到 0~255 的範圍以達到展開效果

24~25: 將沒有值的部分補上 0，再利用 CDF 搭配舊圖索引創建出直方圖均化後的新圖

26~31: 顯示圖片以及 CDF、灰度直方圖等

## B. Histogram Specification

```

33 def histogram_specification(src_img, ref_img):
34     hist_src, bins_src = np.histogram(src_img.flatten(), 256, [0, 256])
35     cdf_src = hist_src.cumsum()
36     # cdf_src_normalized = cdf_src * hist_src.max() / cdf_src.max()
37     cdf_src_normalized = cdf_src / cdf_src.max()
38
39     hist_ref, bins_ref = np.histogram(ref_img.flatten(), 256, [0, 256])
40     cdf_ref = hist_ref.cumsum()
41     # cdf_ref_normalized = cdf_ref * hist_ref.max() / cdf_ref.max()
42     cdf_ref_normalized = cdf_ref / cdf_ref.max()
43
44     Mapping = np.zeros(256)
45     for i in range(256):
46         Mapping[i] = np.argmax(np.abs(cdf_src_normalized[i] - cdf_ref_normalized))
47
48     shape = src_img.shape
49     output_img = np.zeros(shape)
50     for(i, j), value in np.ndenumerate(src_img):
51         output_img[i, j] = Mapping[value]
52     cv2.imwrite('Q2_histogram_specification.jpg', output_img)
53     show_histogram(output_img)
54     plt.subplot(131), plt.imshow(src_img, cmap='gray'), plt.title('Source Image')
55     plt.subplot(132), plt.imshow(ref_img, cmap='gray'), plt.title('Reference Image')
56     plt.subplot(133), plt.imshow(output_img, cmap='gray'), plt.title('Output Image')
57     plt.show()

```

34~37: 將來源圖的 CDF 標準化

39~42: 將參考圖的 CDF 標準化

45~46: 將來源圖的每個灰度值與參考圖的灰度值去做相減

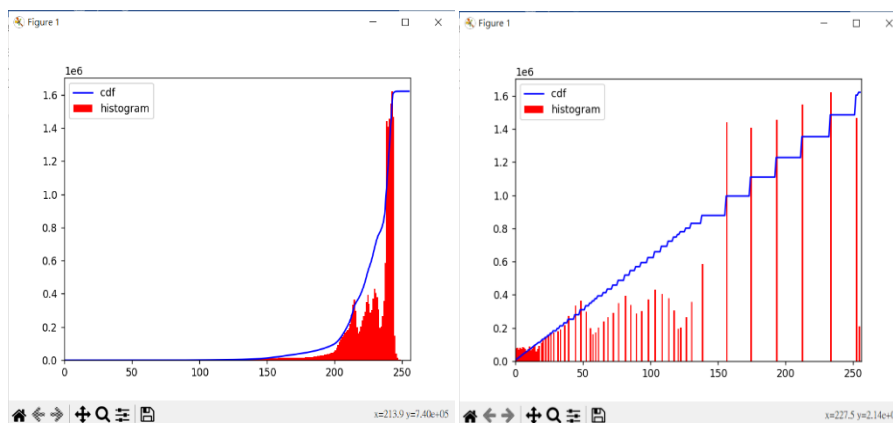
然後找出差距最小的對應索引值，將這個原圖與參考圖的對應關係儲存起來

50~51: 利用原圖套用對應關係製作新圖

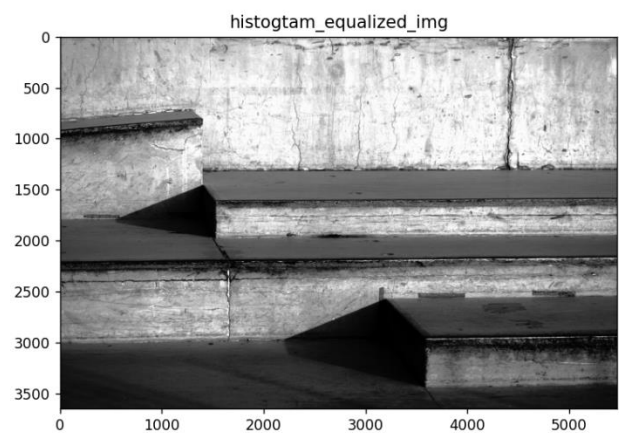
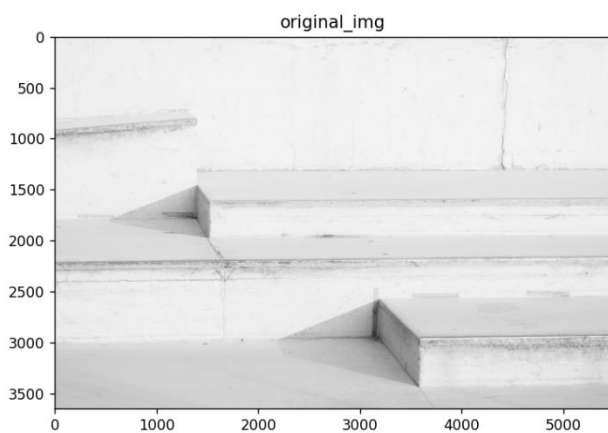
## II. Result

### A. Histogram Equalization

#### 1. 原圖(左)與輸出圖(右)的 CDF 與灰度直方圖

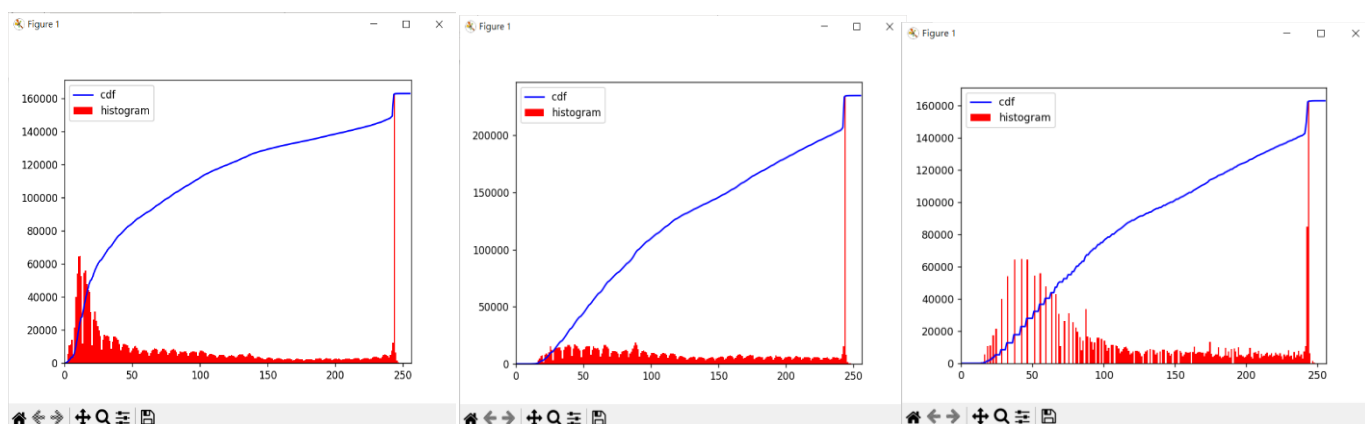


#### 2. 原圖(左)與輸出圖(右)

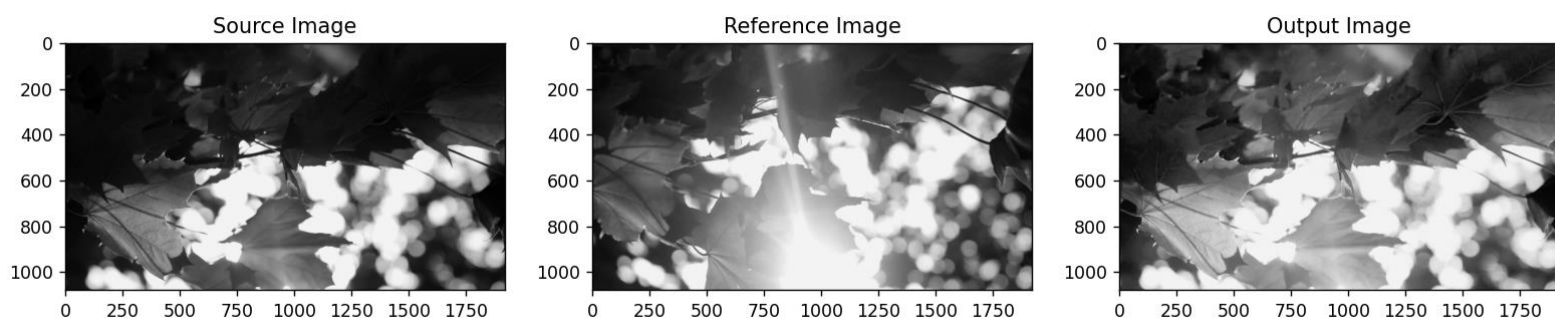


## B. Histogram Specification

### 1. 原圖(左)、參考圖(中)與輸出圖(右)的 CDF 與灰度直方圖



### 2. 原圖(左)、參考圖(中)與輸出圖(右)CDF



III. Feedback: 這次實作了兩種利用累積機率函數的分布去調整圖片的方法，對於如何強化圖片有了更深刻的認識，這也是我第一次輸出圖片的灰度直方圖、累積機率函數，感覺十分新奇。以後遇到機器學習的資料前處理時，會嘗試憑藉該資料領域的特性，套用對應的強化方法，相信會對提升模型的表現有所幫助。