

# Homework 2 Report

## 1. Method

- Histogram Equalization

我定義了以下的 function 來實現 Histogram Equalization：

```
def get_hist(img_one_channel):
    hist = np.zeros(256, dtype=np.float32)

    for y in range(img_one_channel.shape[0]):
        for x in range(img_one_channel.shape[1]):
            hist[img_one_channel[y,x]] += 1

    for i in range(1, 256):
        hist[i] += hist[i-1]

    max_val = img_one_channel.shape[0] * img_one_channel.shape[1]

    result = np.zeros(256, dtype=np.uint8)
    for i in range(256):
        result[i] = np.round(255.0 * (hist[i] / max_val)).astype(np.uint8)

    return result

def hist_equalization(img_one_channel):
    result = np.zeros_like(img_one_channel)
    hist = get_hist(img_one_channel)

    for y in range(img_one_channel.shape[0]):
        for x in range(img_one_channel.shape[1]):
            result[y,x] = hist[img_one_channel[y,x]]

    return result

img = cv2.imread('Q1.jpg', cv2.IMREAD_GRAYSCALE)

res_img = hist_equalization(img)
cv2.imwrite('Q1_processed.jpg', res_img)
```

`hist_equalization` 的用途是取得圖片的 histogram，然後再將原圖每個 pixel 的值換成 {histogram 之 CDF 對應的機率} \* 255。至於乘 255 的部分已經在 `get_hist` 內做過了。

- Histogram Specification

這部分也是定義一個 function 來實現：

```
def hist_specification(src_img, ref_img):
    T = get_hist(src_img)
    G = get_hist(ref_img)

    # histogram specification
    H = np.zeros(256, dtype=np.uint8)
    j = 1

    for a in range(256):
        while T[a] > G[j]:
            j += 1

        if not T[a] == G[j]:
            j -= 1

        H[a] = j

    # apply histogram specification
    dst_img = np.zeros_like(src_img)
    for i in range(src_img.shape[0]):
```

```

        for j in range(src_img.shape[1]):
            dst_img[i,j] = H[src_img[i,j]]

    return dst_img

src_img = cv2.imread('Q2_source.jpg', cv2.IMREAD_GRAYSCALE)
ref_img = cv2.imread('Q2_reference.jpg', cv2.IMREAD_GRAYSCALE)

res_img = hist_specification(src_img, ref_img)
cv2.imwrite('Q2_processed.jpg', res_img)

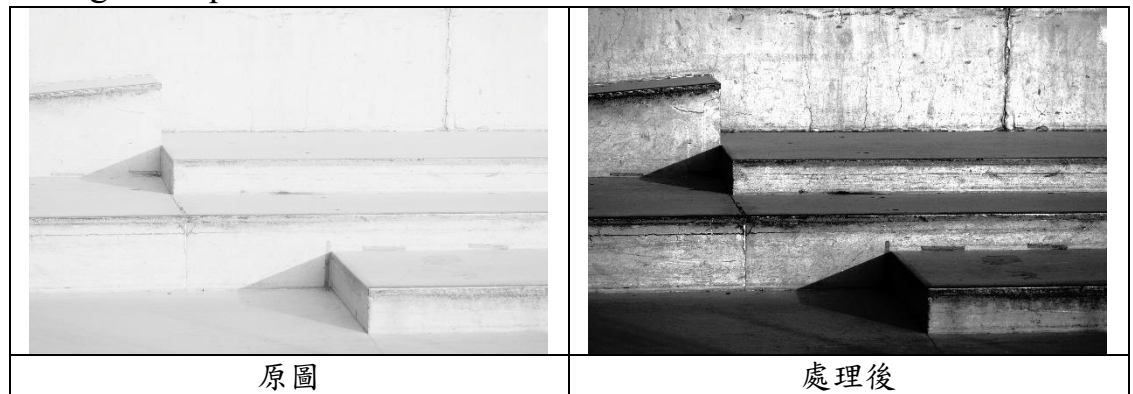
```

這個 function 內做的事也不多，只要將 source 跟 reference 圖片的 histogram 找到之後，將  $z = T(a)$  的值轉換到對應  $G(j)$  的 index，如果在  $G$  裡面沒有找到對應的 index，就找小於且最接近  $z$  的  $G(j)$  來近似；如果  $z$  對應到很多個  $G(j)$  則是取最小的 index。

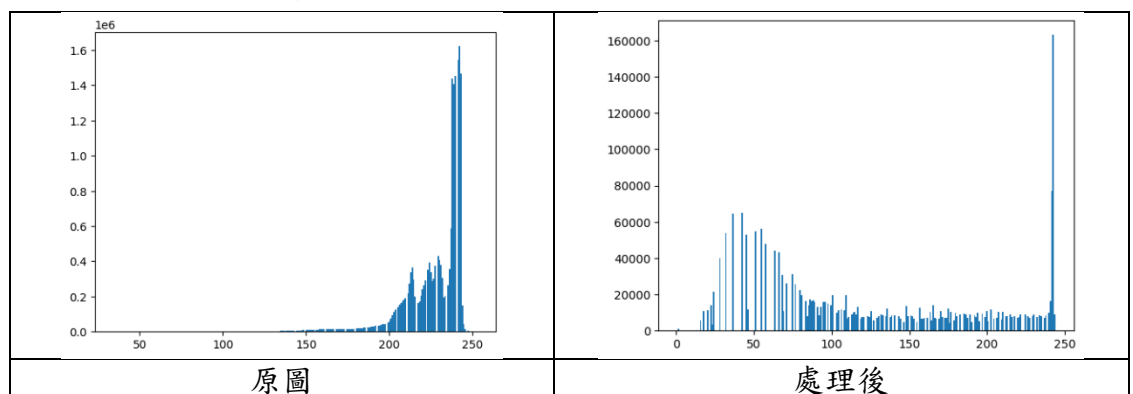
## 2. Result

執行完程式之後可以看到結果如下：

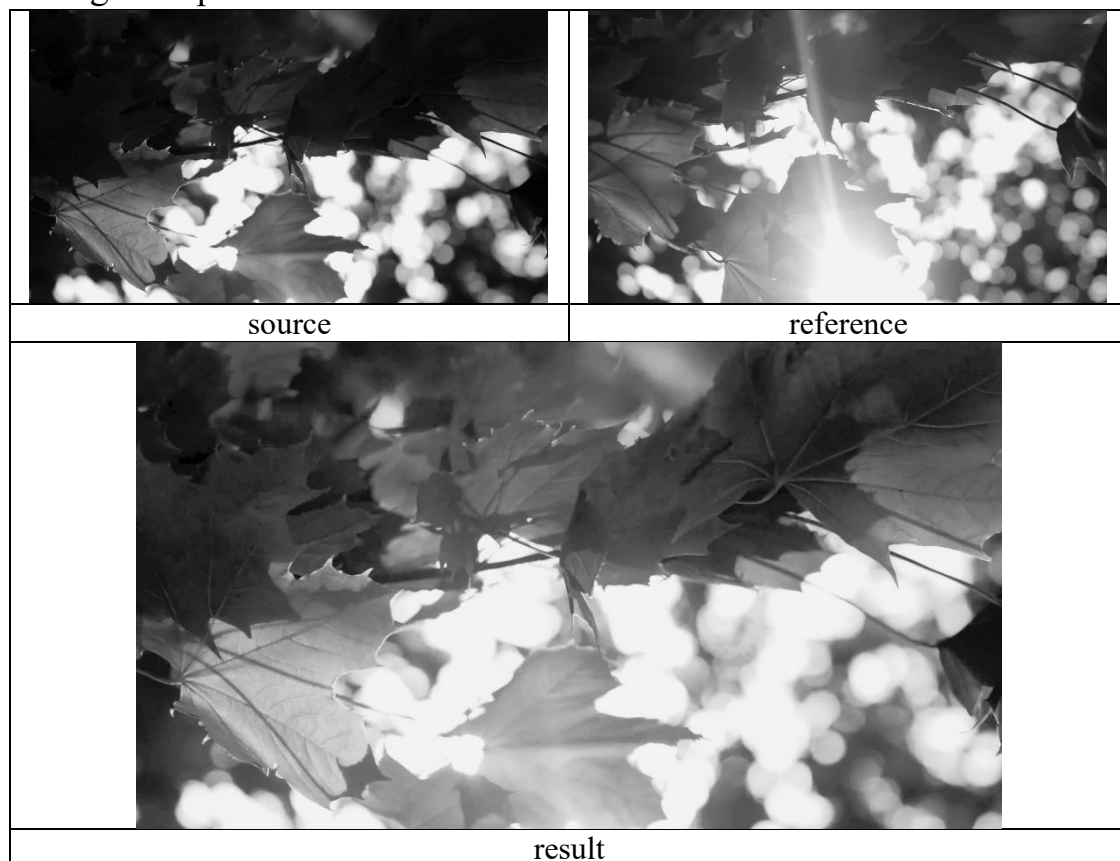
- Histogram Equalization



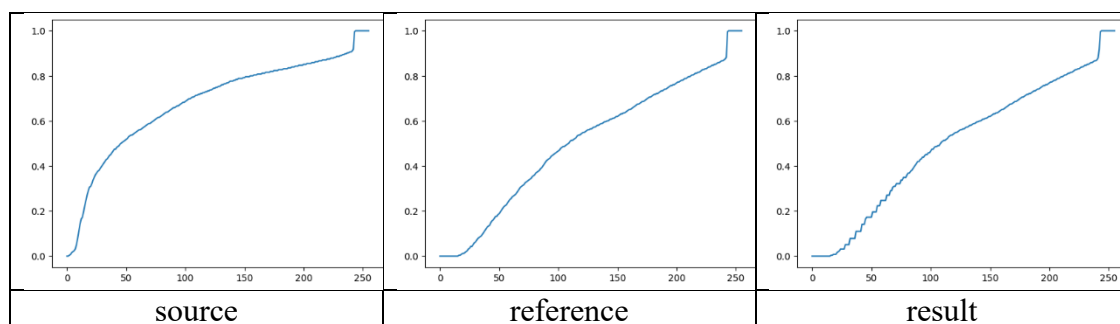
原圖會偏白是因為 pixel 的值分佈都在較高的地方，把處理前後的 histogram 畫出來看就可以知道原因了。處理後的值會分佈得比較散，但因為原圖的分佈就已經非常偏了，所以經過處理的圖片多少還是會有一樣的情形。



- Histogram Specification



經過處理的圖片色調會更接近 reference，就是經過 histogram specification 得到的結果。



把對應的 CDF 畫出來，可以得知 histogram specification 能讓原圖上 pixel 的機率分佈轉換到在 reference 上的分佈。

### 3. Feedback

這次作業將課本上的公式轉為程式，並實際跑過一遍得到結果，讓我對使用 histogram 來處理圖片的幾種方式更為熟悉，也對這些影像處理方式更有概念。