**Hw1\_Histogram Equalization**

Students name : 簡茂芳(409410035)

Data due: 4/17

Data handed in : 4/17

 Technical description

流程概要:

Global approach:

將給定的測試圖片讀出，並將其進行直方圖等化。並且將原圖與處理過後的圖進行比較。

def Global(path):

    img    = cv2.imread(path, cv2.IMREAD\_GRAYSCALE)

    before = cv2.imread(path, cv2.IMREAD\_GRAYSCALE)

    img = ProcessHist(img)

    Display (before, img)

    return

Local approach:

將圖片切成16塊，之後把16塊圖片用陣列傳回，將每一塊做Histogram Equalization 並進行對比，最後，將16 塊被處理後的 block 依照對應部分，重新組成員圖圖，再與原圖比較。

def Local(path):

    subimages = Slice(path)

    origin = Slice(path)

    # if want to know the final result only, you can disable line 111

    for i in range(16):

        afterImg = ProcessHist(subimages[i])

        Display(origin[i], afterImg)

    before = cv2.imread(path, cv2.IMREAD\_GRAYSCALE)

    after = Reconstruct(subimages, before)

    Display(before, after)

    return

具體實作(Histogram Equalization 演算法):

Histogram Equalization:

首先，計算 pmf 以及 cdf 。

再來，將 “T[i] = round(cdf[i] \* 255)”，此時的 T 為函數

(Transformation)。

最後，將每一個像素(pixel)進行轉換即可。

def ProcessHist(img):

    pmf = np.zeros((256,),dtype=np.float16)

    cdf = np.zeros((256,),dtype=np.float16)

    T = np.zeros((256,),dtype=np.uint8)

    m,n = img.shape

    #comput pmf in case loss data ，muliple precent later

    for i in range(m):

        for j in range(n):

            pmf[img[i,j]] += 1

    precent = 1.0/(n\*m)

    # compute cdf

    for i in range(256):

        if (i == 0): cdf[i] = pmf[i] \* precent

        else: cdf[i] =  cdf[i-1] + pmf[i] \* precent

        T[i] = round(cdf[i] \* 255)

    #apply transfrom function

    for i in range(m):

        for j in range(n):

            img[i,j]= T[img[i,j]]

    return img

 Experimental results

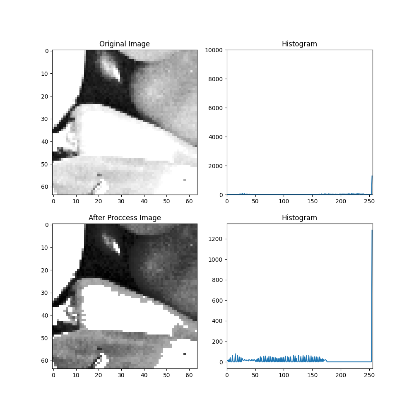
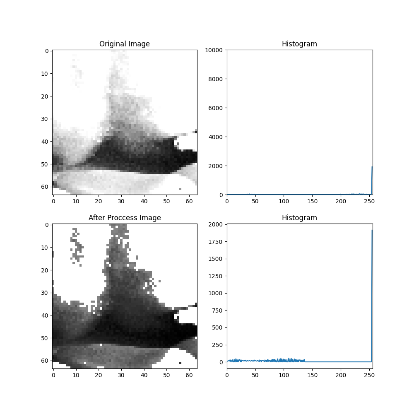
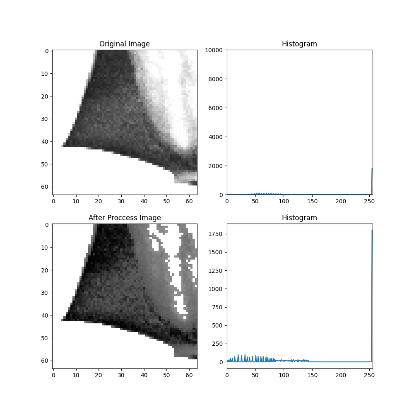
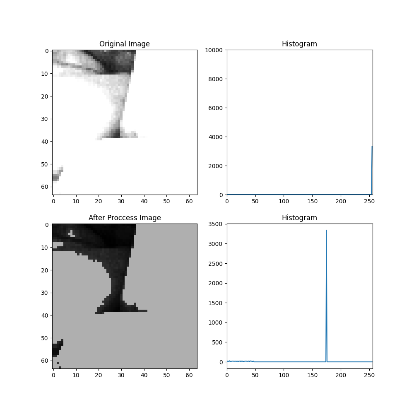
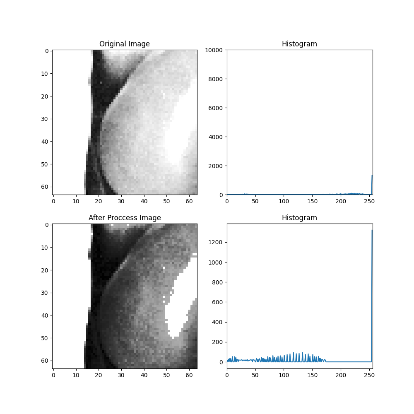
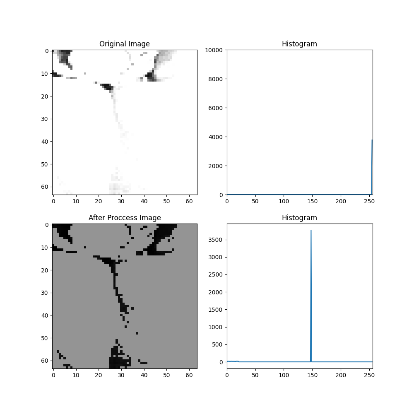
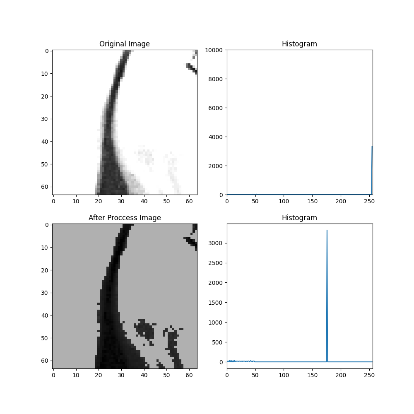
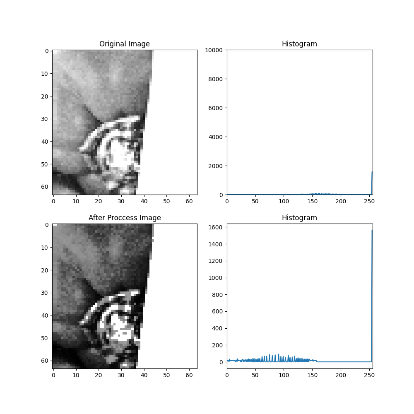
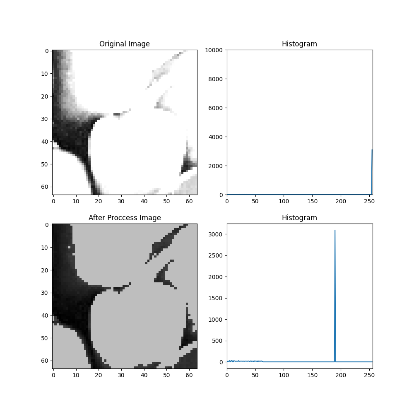
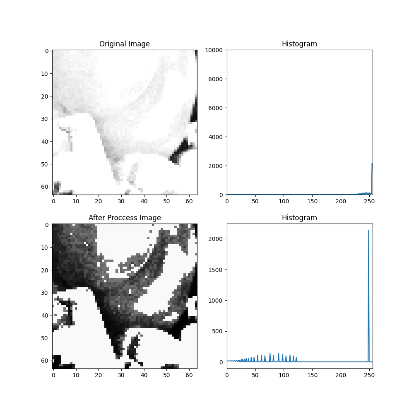
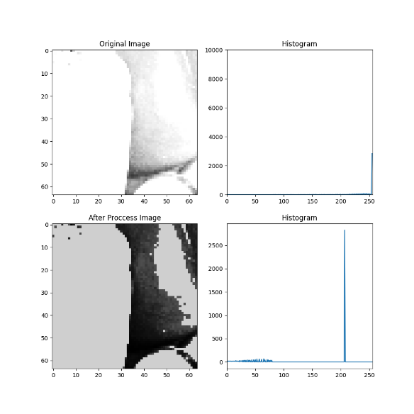
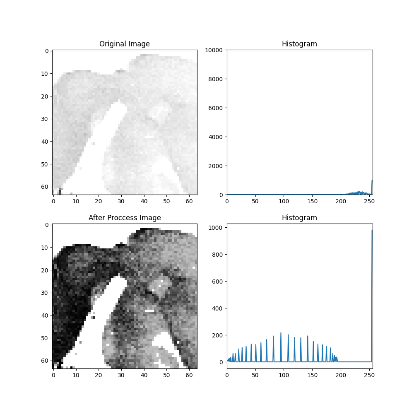
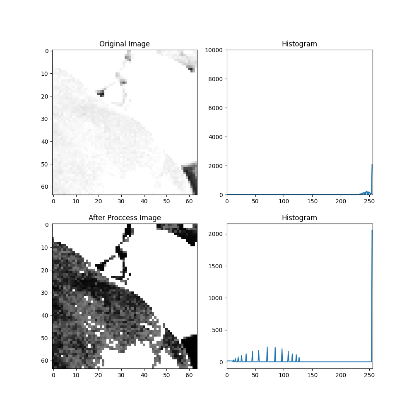
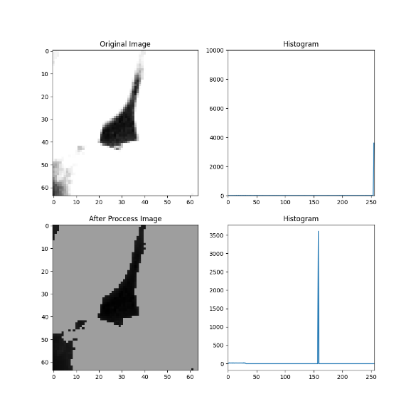
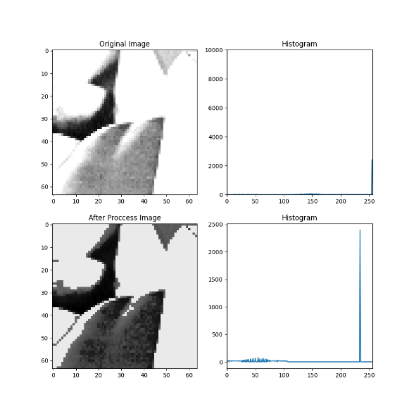
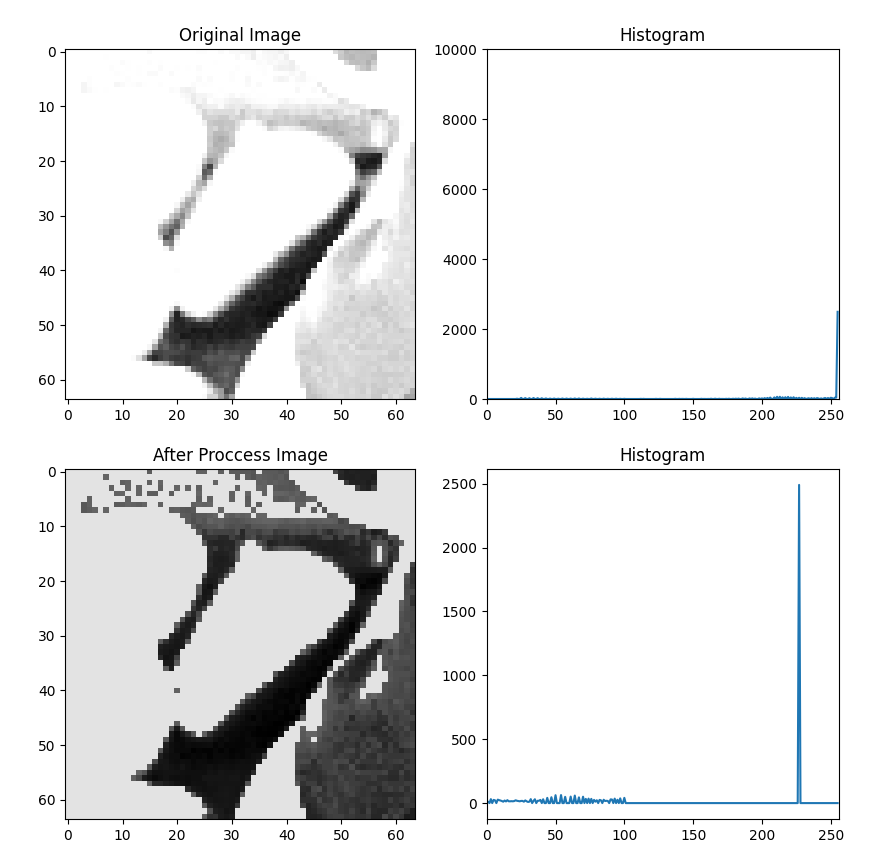
Peppers:

Global:

一張含有 圖表 的圖片

自動產生的描述

Local:



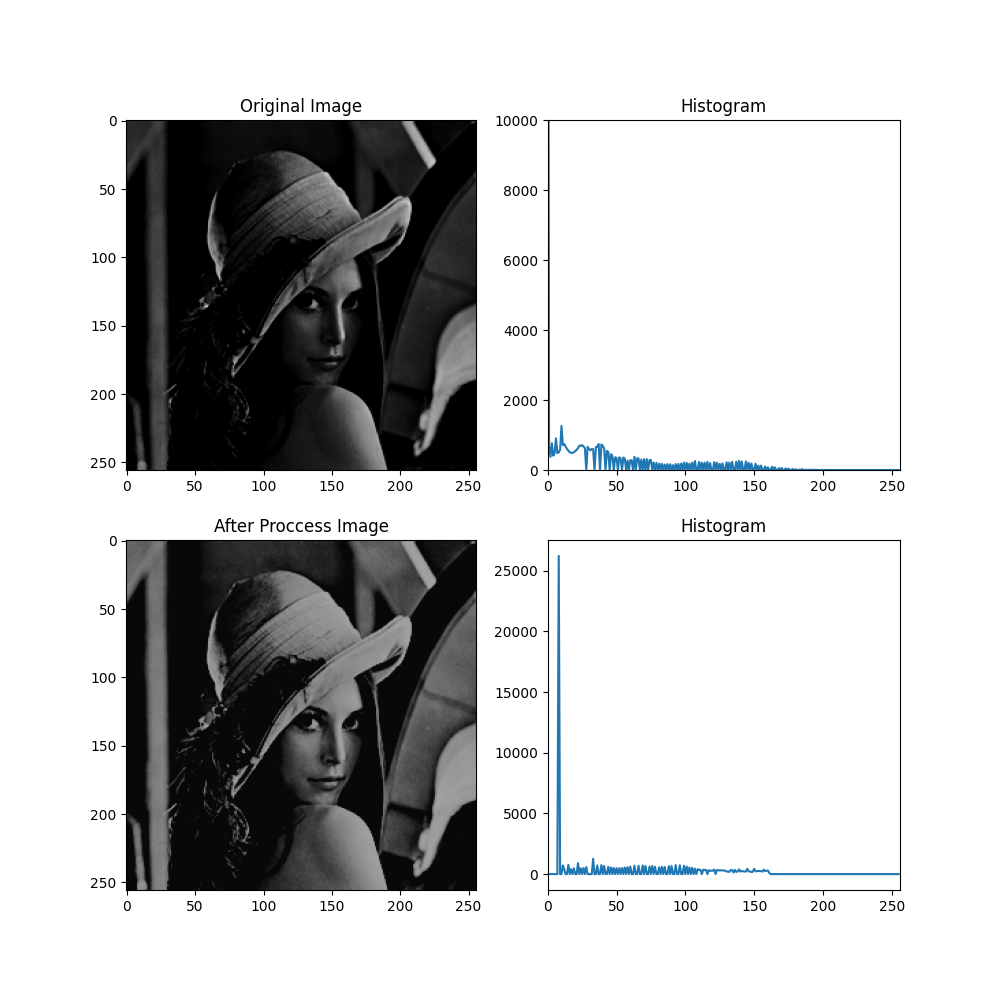
Local 重新組裝後的圖:

一張含有 圖表 的圖片

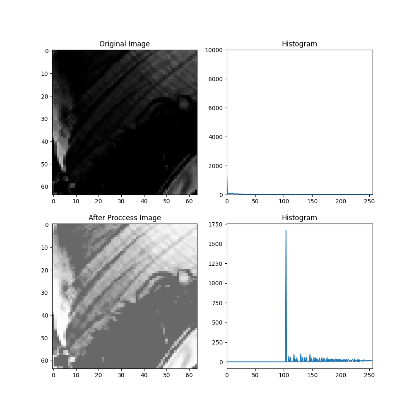
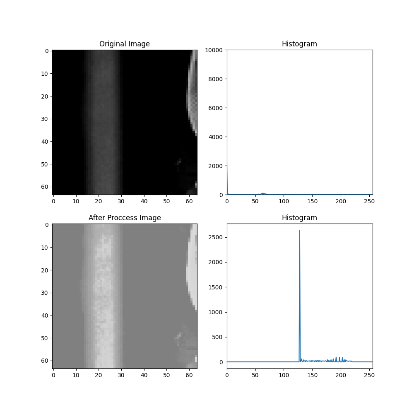
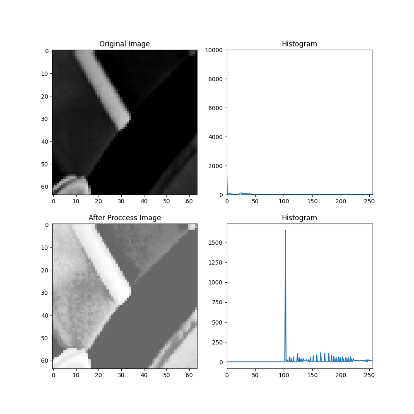
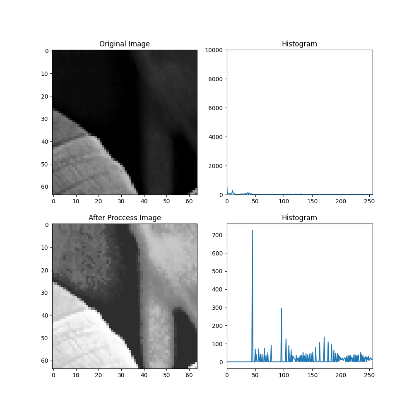
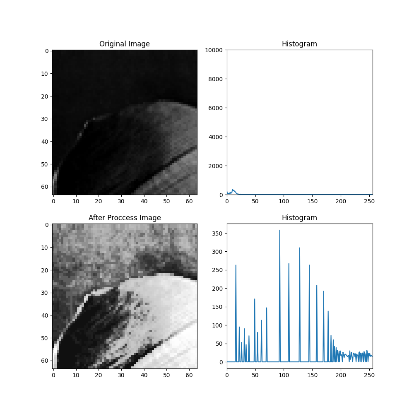
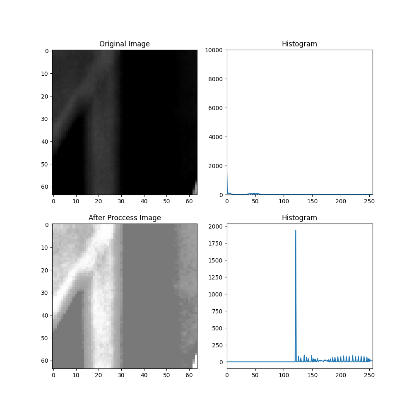
自動產生的描述

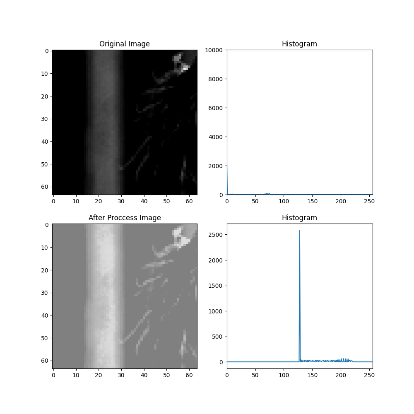
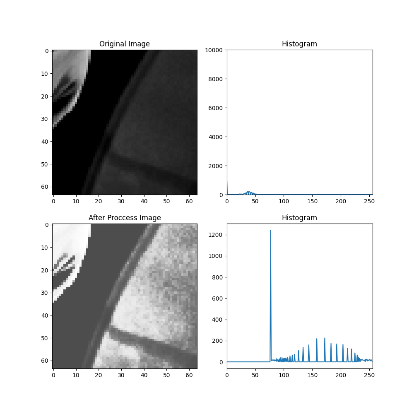
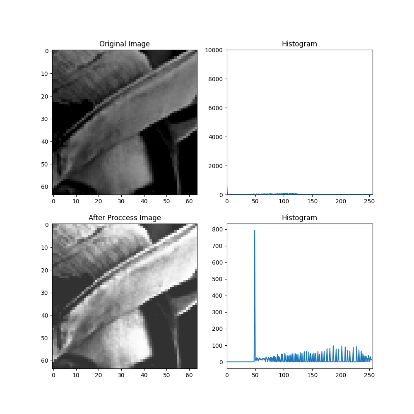
Lena

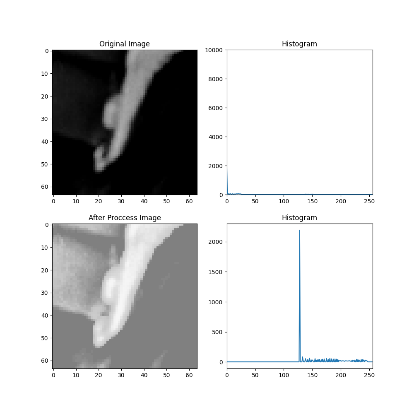
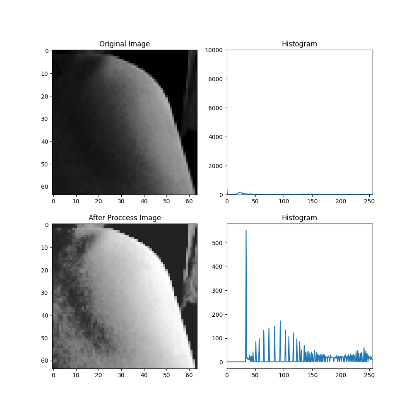
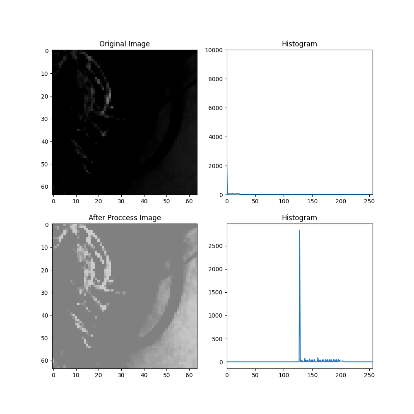
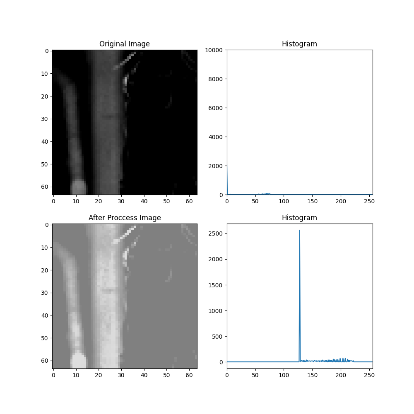
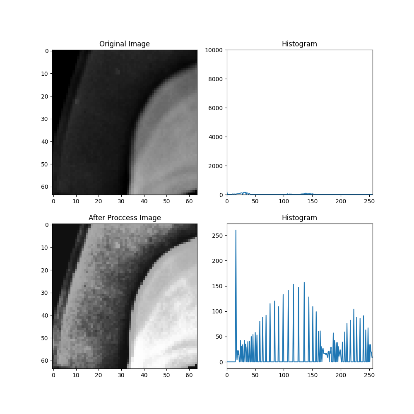
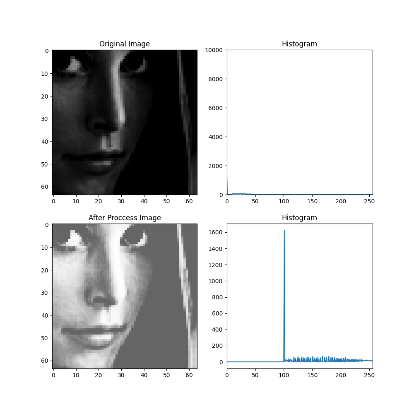
Global:



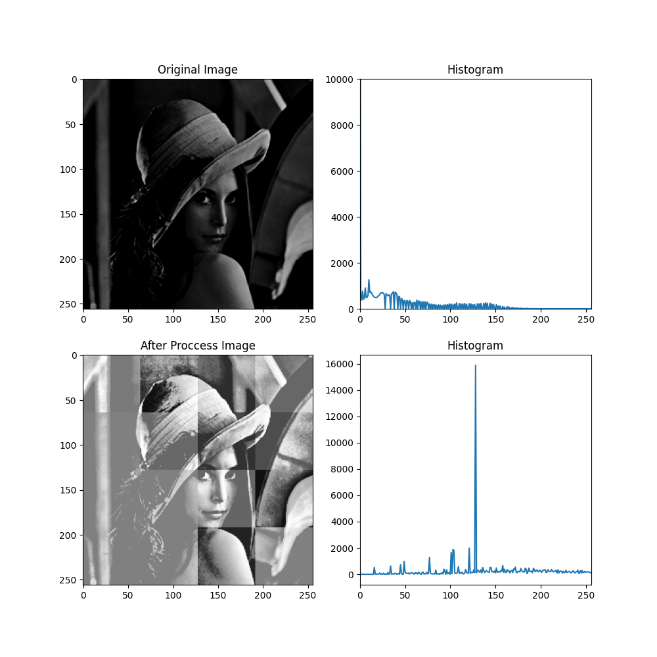
Local:



一張含有 日曆 的圖片

自動產生的描述

Local 重新組裝後的圖:



 Discussions

經過直方圖等化(Histogram Equalization)後，圖片的細節與輪廓變得更加清楚。可以很明顯地感覺到，當圖片進行直方圖等化後的明暗對比效果更好了。而且，因為轉換函數(T)是根據 cdf (累積分布函數cumulative distribution function) 而來，可以確保灰階值較大的像素經轉換，灰階值仍較大。唯一較可惜的是:經過四捨五入，可能會造成有些像素流失(不同灰階值，換算成機率，再四捨五入，可能結果的值一樣)。

 References and Appendix

**數位影像處理\_陳永耀\_單元三 強度轉換與空間濾波\_Part 11 直方圖等化法(下)**

<https://youtu.be/a7WAMC8qvI4?list=PLI6pJZaOCtF2fjFxpVGAqWgENVZw69QD2>