Homework 2

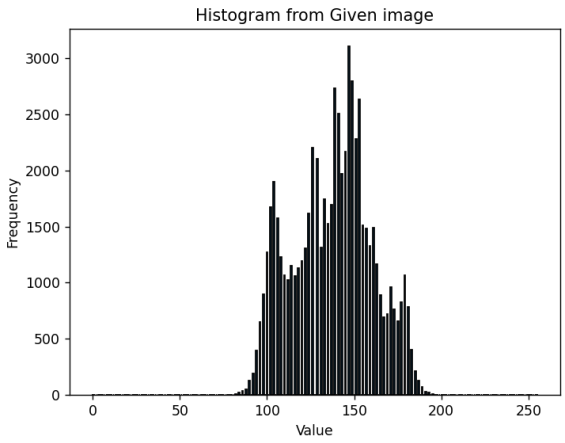
Histogram Equalization

Student： 台科電子碩一 M11302149 趙孟哲

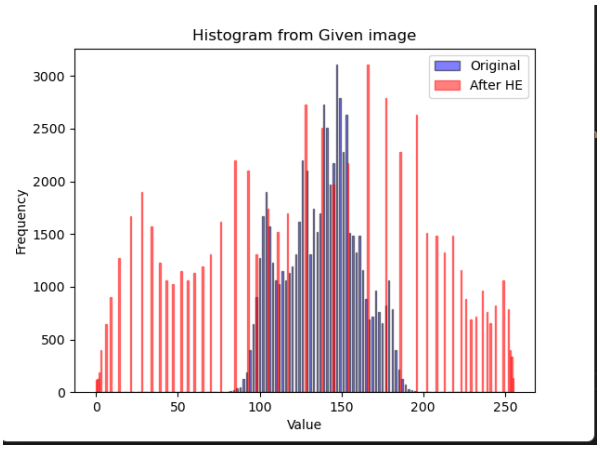
程式碼： github

1.Result

Original:



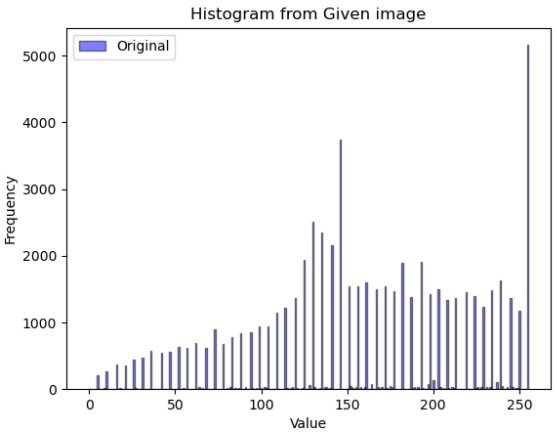
Global HE



PSNR: 28.0291

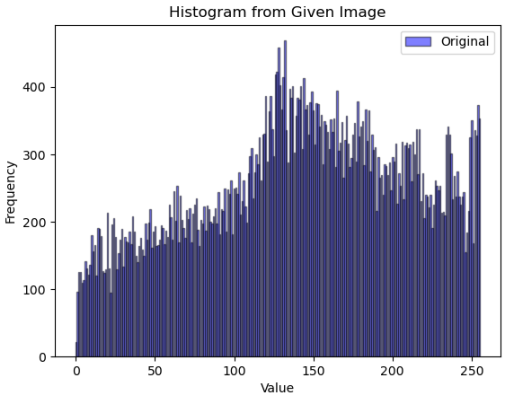
Local HE

- Local kernel 大小: 7



PSNR: 28.0137

- Local kernel 大小: 31



PSNR: 27.8808

Explain

Global Histogram Equalization

```python
def Global_HE(img):
    #TODO:
    # step 1 : Count the number of pixel occurrences (hint : numpy --> unique())
    new_img = np.zeros_like(img,dtype=np.uint8)

    height = img.shape[0]
    width = img.shape[1]

    freq = Generate_Histogram(img)
    cdf= Cumulative_Distribution_Func(freq,height*width)

    # step 2 : Calc                    lization
                    (variable) height: Any
    for i in range(height):
        for j in range(width):
            new_img[i,j] = cdf[img[i,j]]

    # step 3 : Display histogram(comparison before and after equalization)
    new_freq=Generate_Histogram(new_img)
    draw_historgram(freq,new_freq)

    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    cv.imwrite(output_dir+'/'+'output_global.png', new_img)
    print(f"PSNR_global: {calculate_PSNR(img,new_img)}")

    return
```

```python
def Generate_Histogram(img):
    frequency = [0]*256

    for i in range(img.shape[0]):
        for j in range(img.shape[1]):
            frequency[img[i,j]]  +=1

    return frequency
```

```python
def Cumulative_Distribution_Func(frequency,num_pixels,local=False):
    cdf=[0.0]*256
    cdf_normalize=[0]*256

    for i in range(256):
        if(i==0):
            cdf[0] = frequency[0] / num_pixels
        else:
            if(local):
                cdf[i] = min(1,cdf[i-1] + frequency[i] / num_pixels )
            else:
                cdf[i] = cdf[i-1] + frequency[i] / num_pixels

    cdf_min = min([x for x in cdf if x > 0])

    ##normalize
    for i in range(256):
        cdf_normalize[i] = np.round(cdf[i]*255).astype(np.uint8)

    return cdf_normalize
```

說明：

1. 先統計圖片像素將 0~255 得像素統計出現次數(Generate Histogram 函式)，產生直方圖，並繪製出來。
2. 透過 histogram 產生累計分佈函數 (CDF, Cumulative Distribution Function)，是為每個像素圖的機率。然後做 Normalization，將 CDF 範圍根據機率的大小，映射到 0~255 的範圍。
3. 使用 CDF，將圖片每個像素點對應到相對映的 CDF 值
4. 完成直方圖均值化

Local Histogram Equalization:

```python
def Local_HE(img, size=7):
    new_img = np.zeros_like(img,dtype=np.uint8)
    height=img.shape[0]
    width=img.shape[1]

    half_size = size//2

    for i in range(height):
        for j in range(width):
            area = img[max(0, i - half_size):min(height, i + half_size + 1), max(0, j - half_size):min(width, j + half_size + 1)]
            freq = Generate_Histogram(area)
            cdf = Cumulative_Distribution_Func(freq,area.size,local=True)
            new_img[i, j] = cdf[img[i, j]]

    if not os.path.exists(output_dir):
        os.makedirs(output_dir)

    output_path = f"./HW2Histogram_Equalization/output_images/output_local{size}.png"
    cv.imwrite(output_path, new_img)
    print(f"PSNR_local_{size}: {calculate_PSNR(img, new_img)}")
    freq = Generate_Histogram(new_img)
    draw_historgram(freq)

    return new_img
```

Calculate PSNR

```python
def calculate_PSNR(original, compressed):
    mse = np.mean((original - compressed) ** 2)
    if mse == 0:
        return np.infty
    max_pixel = 255.0
    psnr = 10 * np.log10((max_pixel ** 2) / mse)
    return psnr
```

PSNR 是評估影像處理後的品質的指標，以分貝為單位，如果越大代表圖片經過處理後失真越少，也就是品質越好。

$$PSNR = 10\log_{10}\frac{\psi_{max}^{2}}{\sigma_e^2} \text{ (unit: dB)}$$

3.Discussion

Global HE

全域均值化會讓像素分佈有「離散」的效果，會發現有些向素是沒有點在圖片上的。可以觀察結果圖的 Histogram 得到，這是因為 CDF 正規化的方式採四捨五入，會讓灰度值分開來。這個方法可以很好的將影像變成一張對比度明顯的圖片，但對一些局部對比數不夠的圖片，全域的考量會讓演算法無法處理細節的對比度。

Local HE

- Debug 紀錄：

  使用 local 時，要再迴圈內每次重新計算 Number of pixel 大小。一位在邊界處的 size 會不同。

- 實驗：

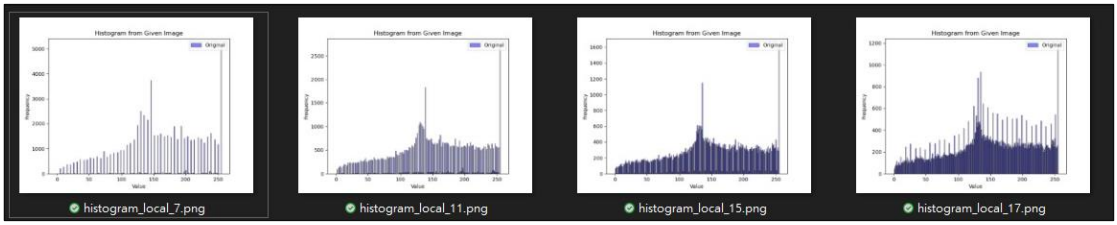
  我將 local histogram equalization 寫成 multi-thread，去嘗試不同的 kernel 大小。分別為 7,11,17,31,41,51,71。

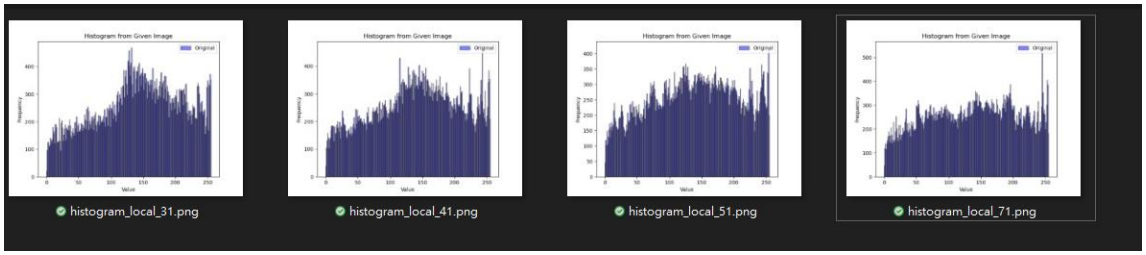
```
sizes = [7,11,15,17,31,41,51,71]

with ThreadPoolExecutor(max_workers=2) as executor:
    for size in sizes:
        executor.submit(Local_HE, img, size)

print("All tasks submitted.")
```

結果如下；





Kernel size	7	11	15	17	31	41	51	71
PSNR	28.014	27.947	27.863	27.866	27.880	27.848	27.832	27.845

觀察 histogram 可以發現整體的灰階分佈隨著 local kernel 的變大-而變為較平均、影像也變得較平滑，接近 global 的效果。但區域較小石能更好的增強局部對比，更適合於細微細節增強的需求。

下圖左為原圖片，右圖為經過 local HE 處理過的圖片，可以看出他也將原圖的區塊瑕疵放大了一個區塊出來。