

# Hw1 Histogram Equalization

資工四 陳昱瑋 409410118

Date due: 2024/4/11

Date handed in: 2024/4/15

## Technical description

假設 gray levels range 為  $[0, L-1]$ ，其中  $L$  為輸入圖之 gray level 最大值，  
histogram function 為：

$$h(r_k) = n_k$$

其中  $r_k$  為第  $k$  個 gray level， $n_k$  為第  $k$  個 gray level 於整張圖的 pixel 數量。  
如果  $n = N \times M$  為圖的總 pixel 數，則 normalized histogram 為：

$$\Pr(r_k) = \frac{n_k}{n}$$

其中， $\sum_{k=0}^{L-1} \Pr(r_k) = 1$ 。  $\Pr(r_k)$  算出了每個 gray level 於整張圖出現的機率。

假設 gray level 經過 normalized 之後為  $r$ ，其範圍為  $[0, 1]$ ， $r=0$  代表黑色， $r=1$  代表白色。Histogram Equalization 為：

$$s = T(r) \quad 0 \leq r \leq 1$$

$s$  即為 level  $r$  經過 Histogram Equalization 後得到的 level 值。

其中  $T(r)$  具有以下性質：

(1)  $T(r)$  is singular-valued and monotonically increasing in the interval  $0 \leq r \leq 1$

(2)  $0 \leq T(r) \leq 1, 0 \leq r \leq 1$

讓  $\Pr(r)$  和  $Ps(s)$  分別代表  $r$  與  $s$  的 probability density functions(pdf)，也就是

$$Ps(s) = \Pr(r) \left| \frac{dr}{ds} \right|$$

Transformer function 即為：

$$s = T(r) = \int_0^r \Pr(w) dw$$

如為 discrete Transformer function：

$$s_k = T(r_k) = \sum_{j=0}^k \Pr(r_j) = \sum_{j=0}^k \frac{n_j}{n}, \quad 0 \leq k \leq L-1$$

如要還原 normalized pixel，再乘上  $L-1$  即可：

$$S_k = (L-1)T(r_k) = (L-1) \sum_{j=0}^k \Pr(r_j) = (L-1) \sum_{j=0}^k \frac{n_j}{n}, \quad 0 \leq k \leq L-1$$

## Execution process

- **functions**

- 1) `histogram_equal(n_k, n, rows, cols, img, cls)`
  - ⇒ 進行 histogram equalization 的主要工作
  - ⇒ `n_k`: each gray level's number
  - ⇒ `n`: image dimension
- 2) `get_gray_level_num(img, rows, cols)`
  - ⇒ 將輸入圖片計算每個 gray level 的數量，return `n_k`
- 3) `plot(img, s_img, title) & sub_plot(img, title)`
  - ⇒ `plot`: 呈現原圖片之 histogram 與處理後圖片之 histogram 比較
  - ⇒ `sub_plot`: 呈現輸入圖片之 histogram
- 4) `slicing(img, div, interval, cls)`
  - ⇒ 進行 block slicing
  - ⇒ `div`: 每一行切成幾份
  - ⇒ `interval`: 每個 block 的邊長(間距)
- 5) `combine_slice_image(slice_img, div, interval)`
  - ⇒ 將每個 block 合起來
  - ⇒ `slice_img`: 為一個三維的 numpy array，(`kth = k`, `row = i`, `col = j`)

- **global way process**

`get_gray_level_num` => `histogram_equal` => `plot` & `subplot`

- **local way process**

`slicing` => `get_gray_level_num` for each block => `histogram_equal` for each block  
=> `combine_slice_image` => `plot` & `subplot`

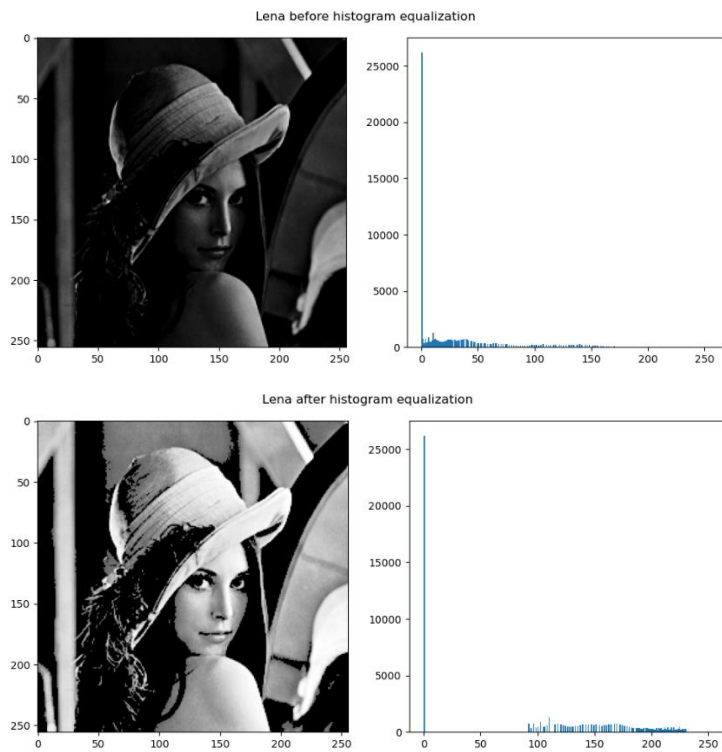
- **Execution**

- Histogram equalization in global way  
`python histogram_global.py`
- Histogram equalization in local way  
`python histogram_local.py`

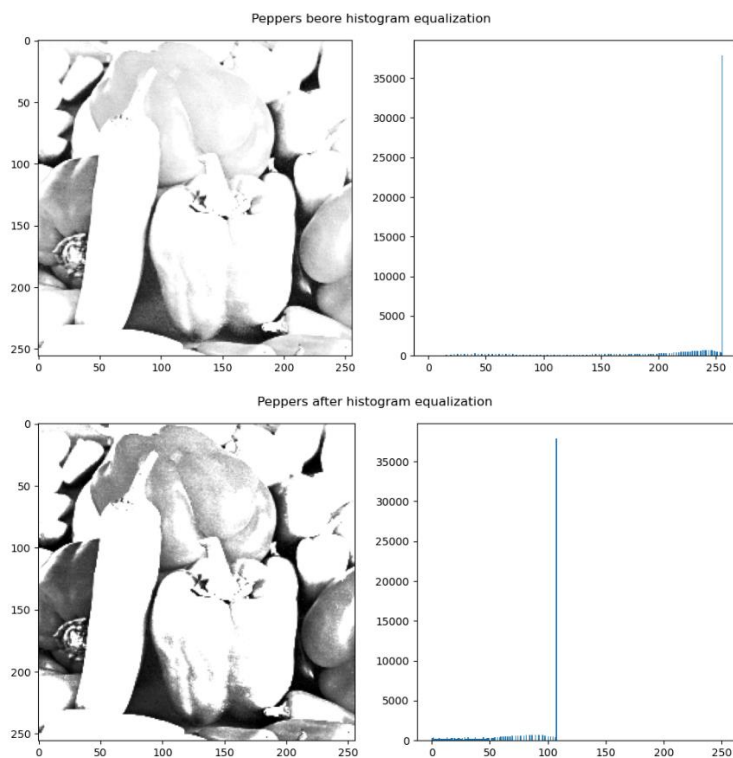
## Experimental result

### 1. Histogram equalization in global way

- Lena

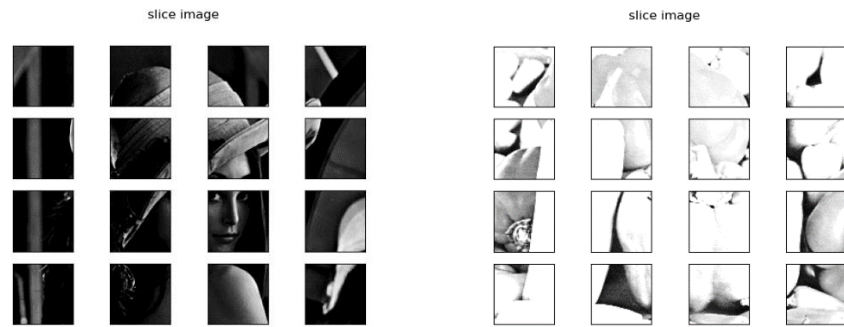


- Peppers

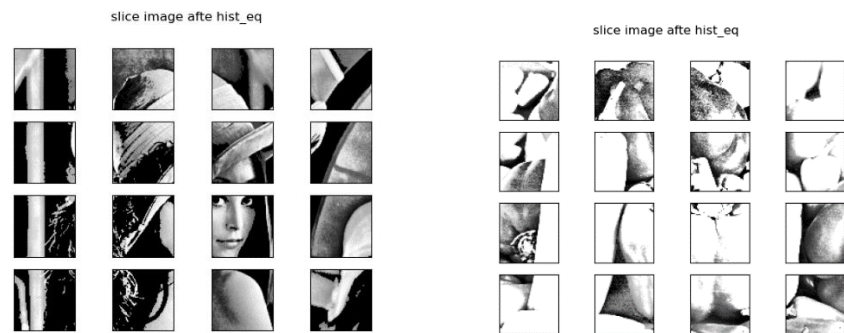


## 2. Histogram equalization in local way

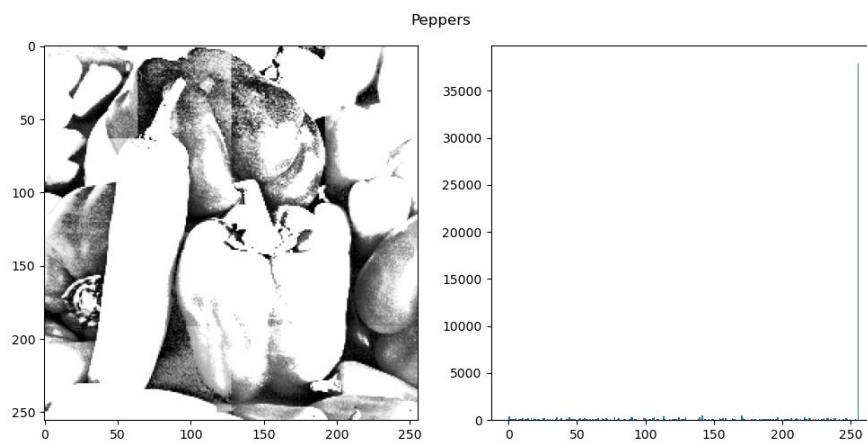
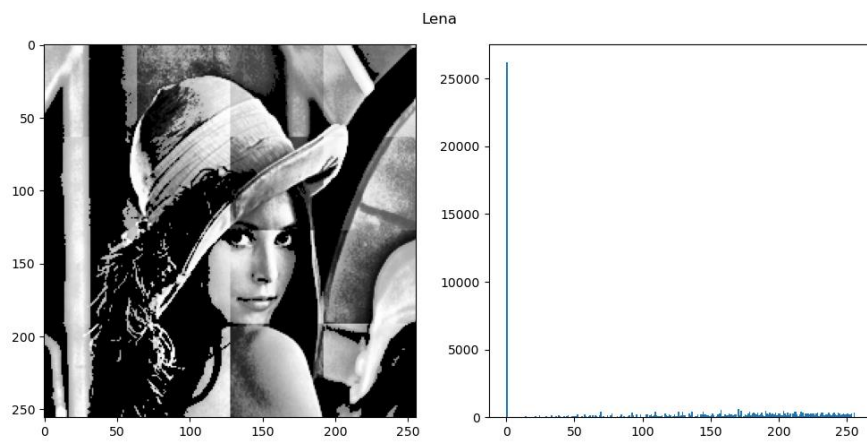
- sliced images: Lena/Peppers



- sliced images after histogram equalization: Lena/Peppers



- images after histogram equalization: Lena/Peppers



## Discussions

- **Exploration about global histogram equalization**

以一般的做法使用 global way，由公式  $\Pr(r_k) = \frac{n_k}{n}$  可以得知 histogram

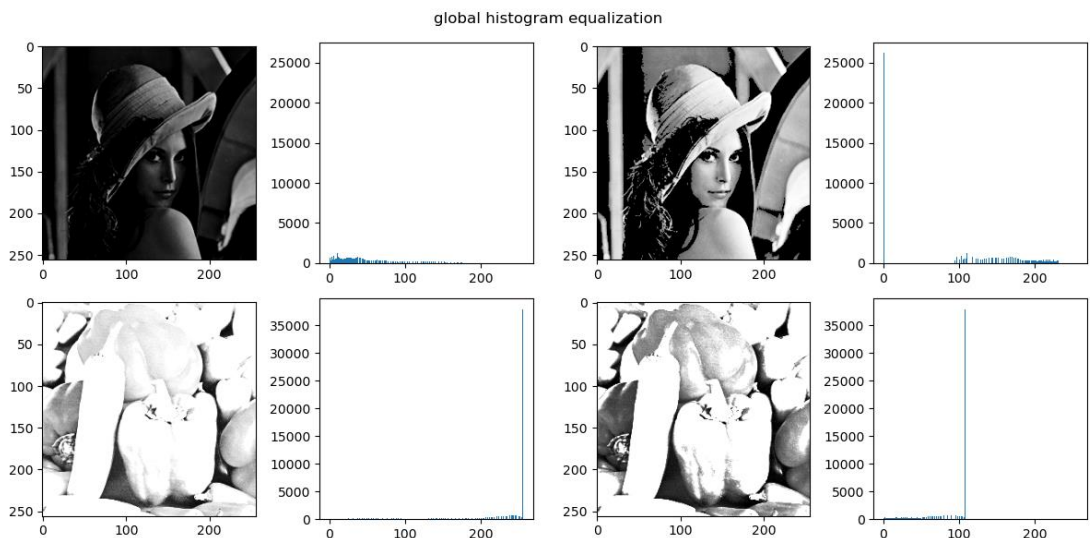
normalization 是根據整張圖為基礎，去得出 "global" pixel number for each gray level。以 Lena 為例，由於原圖整張圖偏暗，histogram 在 gray level 0 的附近產生峰值 (peak value)，經過 transformer function:  $s_k = T(r_k) =$

$$\sum_{j=0}^k \Pr(r_j) = \sum_{j=0}^k \frac{n_j}{n}, \quad 0 \leq k \leq L-1$$

，將 gray level 0 的數值分散在剩餘的 gray level，因此得到 gray level 較平均之圖像，也可以說明成「對比較強」之圖像（黑的地方就是黑，而該亮的地方也會因此凸顯出來，因為其他 gray level value 是加成在 gray level 0 數量的基礎上）。

Lena（曝光不足）的反例就是 Peppers（曝光過度），可以看到整張圖呈現過鋪的情況，gray level 255 附近產生峰值 (peak value)，透過如 Lena 的方法，以 gray level 0 為基底，分散於剩餘 gray level。因為大部分的值都集中在 255，經過 transformer function，峰值被分散到較低的 gray level range。

經過實驗，可以發現 L 值可以影響整張圖 gray level number 的分布。如果將 L 設為 256，會使經過 histogram equalization 後的圖的對比更強烈，由於  $s_k$  為 normalized 後的值， $S_k = (L-1)s_k$ ，當 L 越大，則  $S_k, 0 \leq k \leq L-1$  之間的差距越大。



- **Comparison between histogram equalization in global way and local way**

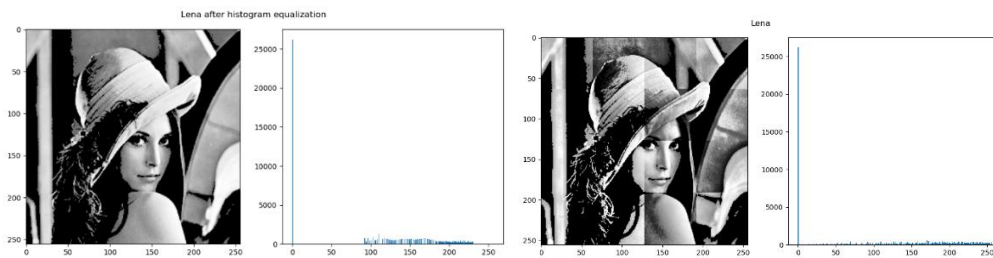
local way 即將原圖切割為  $x \times x$  的圖，在實驗中我們將原圖切割成 16 個

blocks，即  $x \times x = \frac{256 \times 256}{16} = 64 \times 64$ 。將 16 個 blocks 分別做 histogram

equalization 最後將 16 個 blocks 合成，對合成的圖做 histogram 觀察。在這裡對 global 與 local 做差別上的討論。

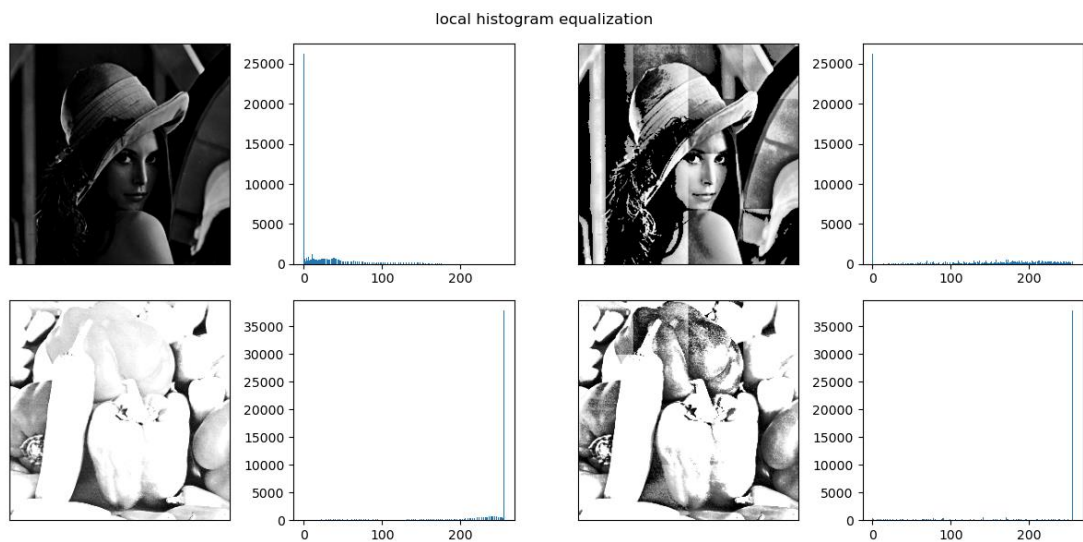
可以觀察到若使用 local way，在每個 block 都會有明顯的對比，由於每個 block 的資訊(gray level)相較於整張圖都較少，因此會造成最後的合成圖整體對比度都較為強烈；反觀 global way，考慮到了整張圖 grayscale 於各個 level 數量的分布，最終結果呈現較為滑順，且灰階分布、對比也較為合理。

以 Lena 為例，以下為 global 與 local 的 histogram：



可以明顯看出來 global 的 histogram 不向 local 那麼的分散，大部分集中於 100~240 的範圍，而 local 的範圍則為 50-250。大範圍的 gray level 也就導致最後的圖片呈現高對比、強烈的效果。

注意在實作 block concatenation 須對每個 block 做 normalized，由於每個 block 的 max, min{gray level} 不同，需透過 normalized 將每個 block 的 gray level range 調整為 [0, 255]，在 concatenate blocks 才不會造成明暗度落差。



## **References and Appendix**

[Matplotlib documentation — Matplotlib 3.8.4 documentation](#)

[Matplotlib Subplot \(w3schools.com\)](#)

[NumPy documentation — NumPy v1.26 Manual](#)

[OpenCV: Histograms - 2: Histogram Equalization](#)

[直方圖均化\(histogram equalization\) @ 小花的學習日誌 :: 痞客邦 :: \(pixnet.net\)](#)