

# HW1 Histogram Equalization

408410094 資工三 葉丞勳

Data due: 4/16

Data handed in: 4/9

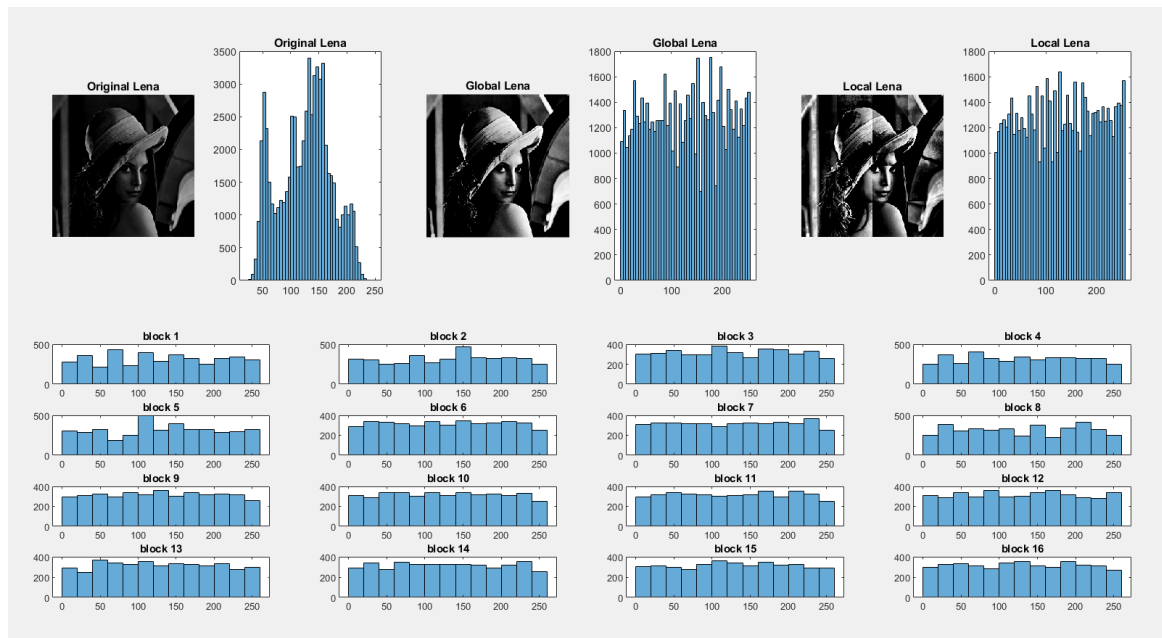
# 1. Technical description

(1)Global :首先將所提供的圖檔使用 `imread()` 讀入，然後利用 `size` 將此圖片的行數跟列數，而此圖片總共的像素個數就是行數乘以列數。接下來準備四個空的 `256*1` 的矩陣，分別是 `frequency`, `pmf`, `cumulative`, `result` 用來統計此圖片的像素分佈。一開始用兩個 `for` 迴圈將圖片中每個像素的值取出，並利用 `frequency(i)` 來統計每個 `gray level` 出現的次數，接著將每個 `gray level` 出現的次數除以像素的總個數，將所得到的結果存入 `pmf(i)`，表示此 `gray level` 在這張圖中出現的機率。接著宣告變數 `sum=0`，然後用一個 `for` 迴圈從 1 跑到 256，每跑一次迴圈，就把 `frequency(i)` 加到 `sum`，表示 `gray level` 從 1~i 的出現次數，然後把 `sum` 除以像素總個數，存進 `cumulative(i)`，表示 `gray level` 從 1~i 在圖中的出現機率，接著將 `cumulative(i)` 四捨五入後乘上 255 再利用 `uint8()` 轉乘 `uint8` 型態後存入 `result(i)`，建立每個 `gray level` 的對照表。最後同樣先用兩個 `for` 迴圈走遍圖片中每個點，將 `result(img_Lena(i, j)+1)` 存入新的矩陣 `glo_Lena(i, j)`，表示將本來在 `(i, j)` 這點的像素值加 1 後再將它作為索引去剛剛建立好的對照表中查表，查完後將結果存入新矩陣的 `(i, j)` 這點。

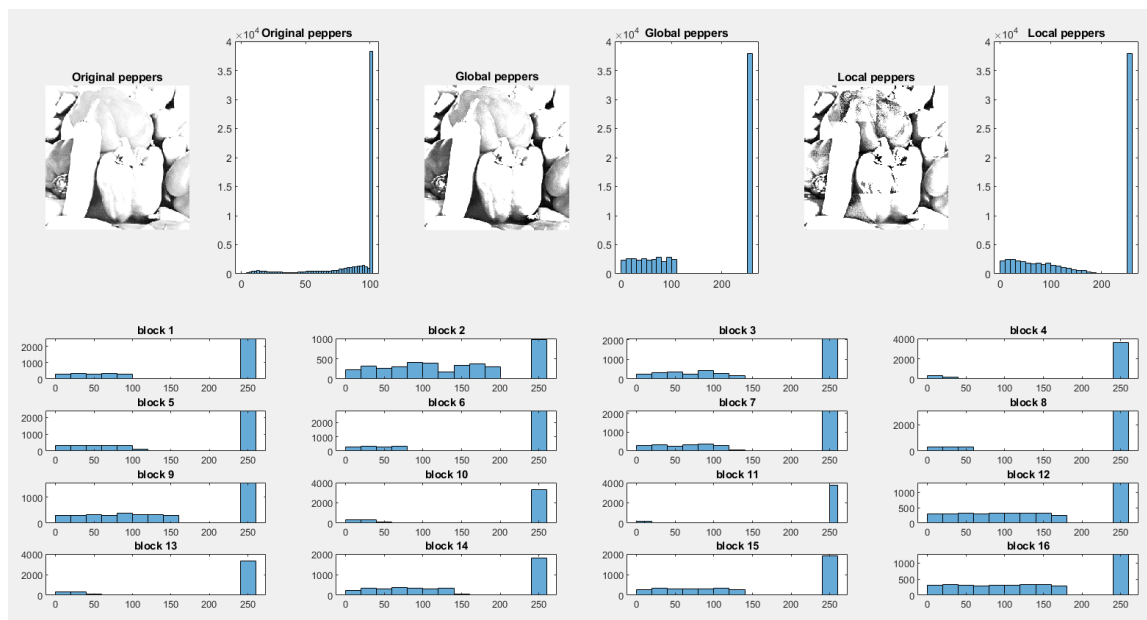
(2)Local :大致上與 `global` 的方法類似，差別在於它是個別的對切割好的區域進行 `histogram equalization`，其他的步驟都一樣。

## 2.Experimental results

### (1) Lena



### (2)Pepper



第一列從左到右分別為原始圖片、global histogram equalization 後得到的圖

片、local histogram equalization 後得到的圖片。下面的 16 張圖表示在做

local 的 histogram equalization 時切出來的 block 的 histogram。

## 4.Discussion

在 3 的實驗結果中，可以發現 global equalization 會比 local equalization 的效果還要好，在 local equalization 後的圖片會明顯看圖片中有一格一格的現象，我認為這是因為在做 local 的時候會分別對每個 block 做 equalization，而由於每個 block 做完 equalization 後彼此像素的值並不會連續，因此會產生上述所說的現象。還有在對 Lena 做 equalization 時，可以觀察到不論是 global 或是 local 的 histogram 都分佈個蠻均勻的，但對 peppers 做的時候，他的 histogram 總會有一部份一直處於 gray level 高的地方，其他部分就分佈的很均勻，我認為這是由於原始圖檔的分佈很不平均的關係，才會造成以上的現象。

## 5.References and Appendix

<https://jason-chen-1992.weebly.com/home/-histogram-equalization>