DIP HW1　R06944023 吳尚真

WARM-UP: SIMPLE MANIPULATIONS

a) 做水平翻轉，輸出 B.raw 的圖檔

1) Motivation and approach: 因為一張 gray-level 的影像可以想像成是二維的矩陣，每一個 pixel 就是矩陣當中的每一個點。若想要將圖片做水平翻轉，只需要將每一列最左邊的 pixel value 放到最右邊 即可。也就是說，將每一個 row 的第一個 column 換到最後一個 column，第二個 column 換到 倒數第二個 column，以此類推。

2) Original images (optional)

3) Output images

4) Discussion of results: 左邊是原圖，右邊是結果。可以看到，原圖成功被水平翻轉

b) 使用 power transform 來 enhance B.raw

1) Motivation and approach: 根據上課的投影片，我先將原始圖檔的 pixel value normalize 到[0,1]，再做 power，最後再將 pixel value 恢復到[0,255]的區間。

2) Original images (optional) : 同上方左圖

3) Output images (由左至右 exponent = 0.5, 1.5, 2, 2.5, 3, 3.5)





4) Discussion of results: 因為我先將 pixel 值 scale down 到[0,1]區間，我們知道，當 base value 介於[0,1]之間時，exponent value 愈大，power 出來的 pixel 越小。所以再 scale up 回[0,255]後的 pixel value 也會越小，出來的影像會越黑。由上列的各種不同 exponent 的 output images 可以發現到，當 exponent=0.5 時，會太亮，看不清楚一些細節；而若 exponent 取太大，則會太黑。我個人覺得取 2.5 時，看起來最好，因為可以看清楚細節，又不至於讓整張圖片的周圍過黑。

PROBLEM 1: IMAGE ENHANCEMENT

a) 將原圖的 pixel value 除以 2，輸出 D

b) 將原圖的 pixel value 除以 3，輸出 E

1) Your motivation and approach: 因為 a) 與 b) 的差異只在於 pixel value 的除數，所以一起討論。主要的方法就是將每個 pixel value 讀出來，除以指定的被除數，再輸出成 D 及 E。

2) Original images (左)　　3) Output images (D:中)　　(E:右)

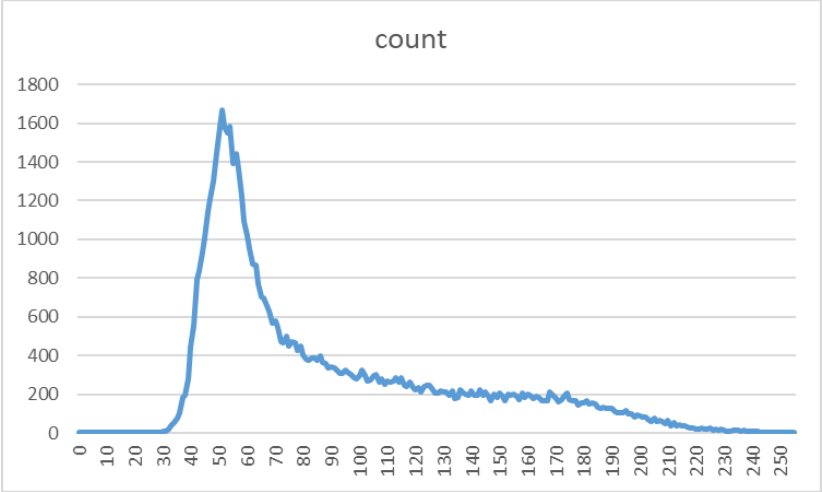


4) Discussion of results: 將 pixel value 除以 2 後，值會變小，值越小 顏色就越黑。所以可以發現，將原圖與 D、E 比較，D 比原圖黑，而 E 又比 D 黑。

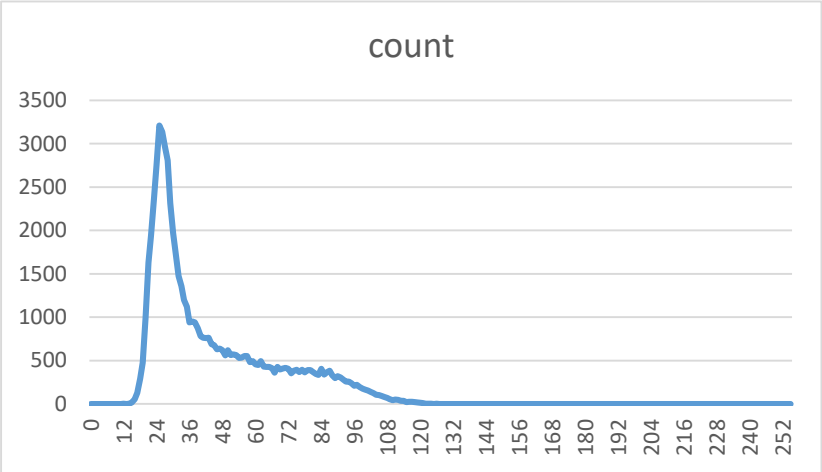
c) 繪出 原圖、D、E 的直方圖

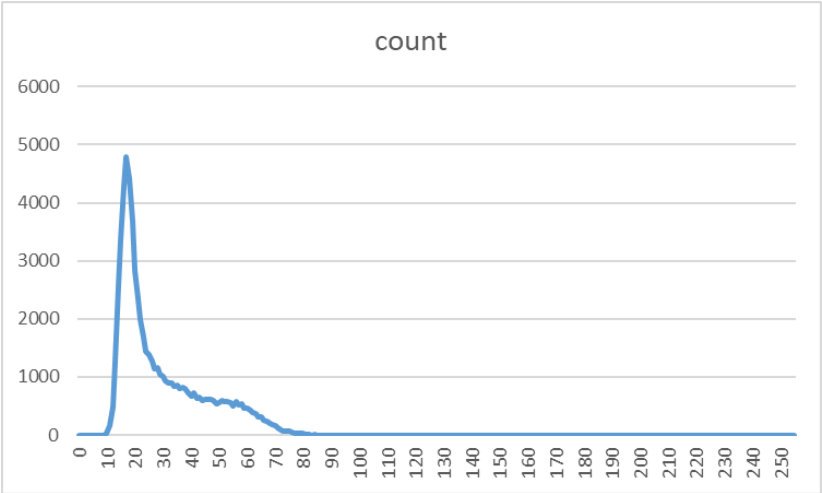
1) Your motivation and approach: 根據老師上課時的提示，可以輸出數據，再用 excel 讀數據、繪圖。我參考[1]，將每個 value 值的 count 輸出成 txt 檔 ，再用 excel 開啟 txt 檔，最後再用 excel 內建的折線圖畫出直方圖。

2) Original images (optional) 原圖 I2 的直方圖



3) Output images (上 D 下 E)



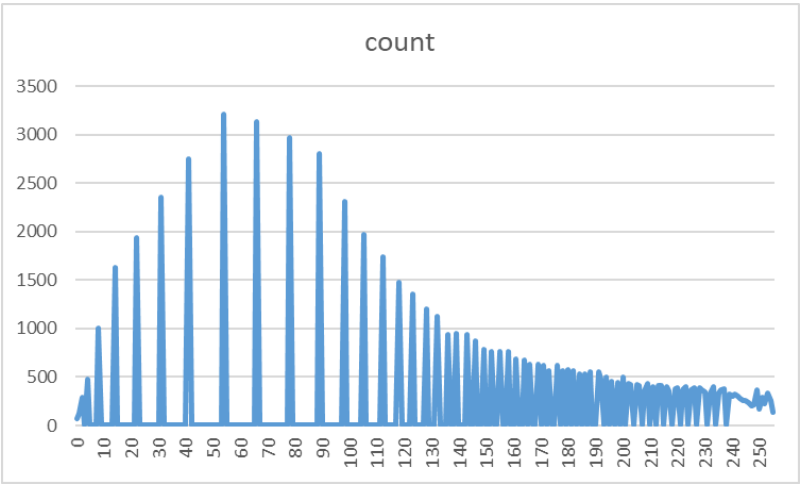


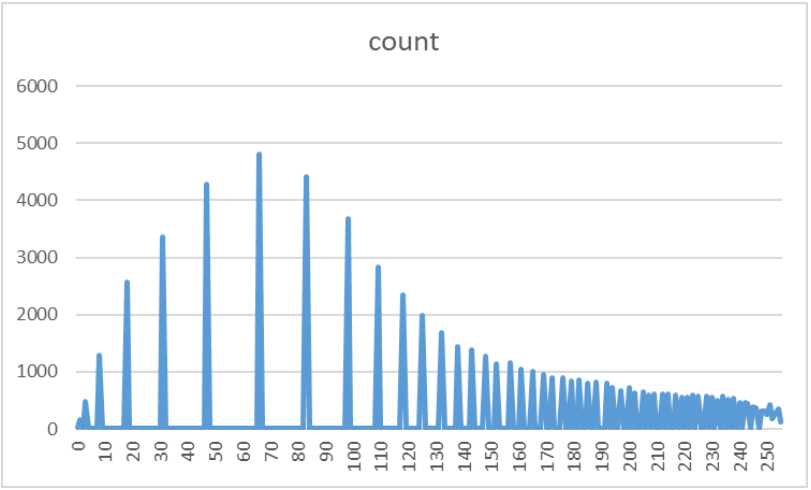
4) Discussion of results: 比較 I2, D, E 的直方圖，可以發現原圖的 pixel value 集中在[30,80]，D 的高峰則往左靠，集中在[15,40] (為原圖/2)，而 E 則更往左，集中在[10,25] (約為原圖/3)。此外，因為 pixel value 被壓縮到左邊，所以 E 的 peak 的 count 數 > D >原圖。

d) 對 D、E 做直方圖均化(global histogram equalization)，並繪出直方圖

1) Your motivation and approach: 根據上課內容，做 global 的直方圖均化，首先要 count 0-255 中 每一個 value 總共有幾個 pixel，接著計算累積分布函數 CDF (將自己以及自己之前的個數累積起來)，因為 CDF 是機率密度函數，機率最大值為 1，所以累積完後要再除以總個數。最後再將原始的 CDF map 到斜率為 1，range 為[0,255]的 normal distribution 的 CDF。

3) Output images 左: Hd 右:He





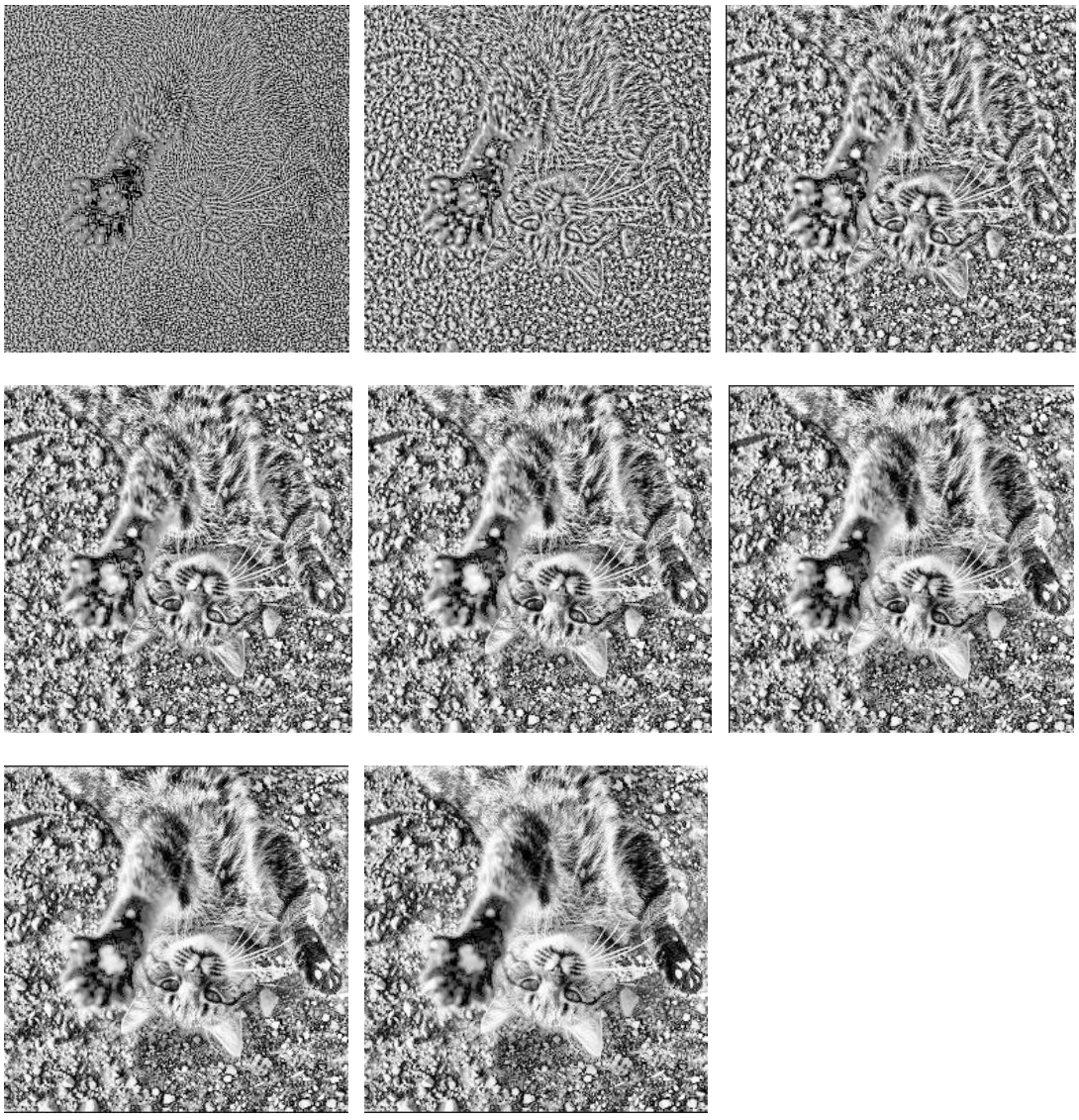
4) Discussion of results: 比較原圖的直方圖 與 Hd, He 的直方圖，可以發現 原始 pixel value 分布情況的 shape 有被保持住，而 He 因為 E 原來的 pixel value 分布比 D 集中，所以均化後，每一個 pixel value 的分布會比 D 遠。(觀察 pixel value [10,110]，He 的每條 slot 的間距都較 Hd 大)。當觀察.raw 檔，D 與 E 的顏色差距很明顯，但再做了 Histogram Equalization 後，Hd 與 He 的圖檔，肉眼上幾乎看不出有甚麼差異，但與 D, E 相比，Hd, He 的明亮對比較明顯。

e) 對 D、E 做區域直方圖均化(local histogram equalization)，並繪出直方圖

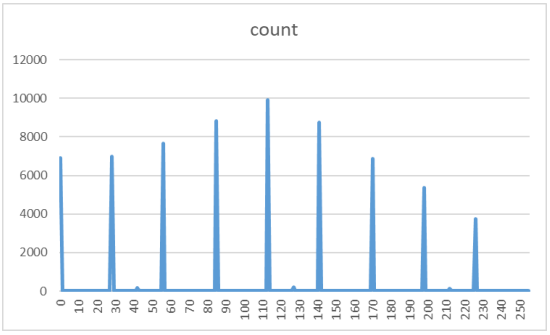
1) Your motivation and approach: 根據我在網路上找到的文件[2]，local HE: 將每一個 pixel 與鄰近 pixel 的灰階值做比較，決定其排序。再依此一排序的正比關係指定一個新的灰階值給這個 pixel。區域的大小(window size)可自己調整。

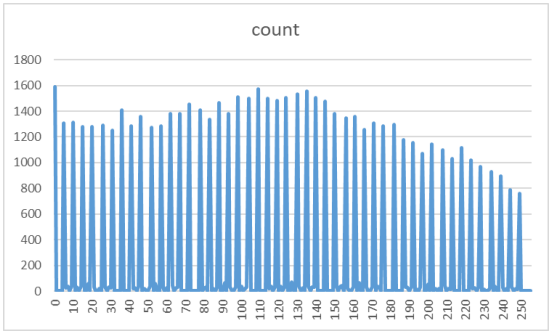
3) Output images

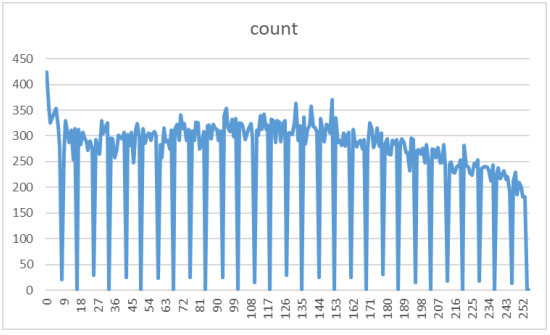
Window size 由左至右，上至下: 3, 7, 15, 21, 31, 45, 51, 71 (先只比較 Hd)

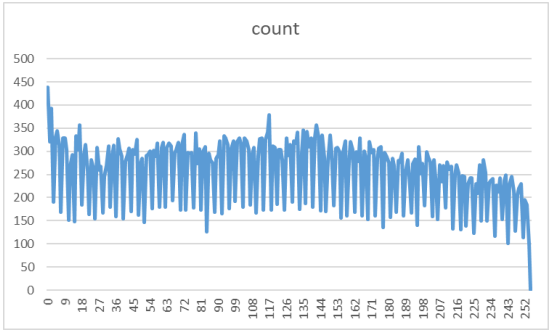


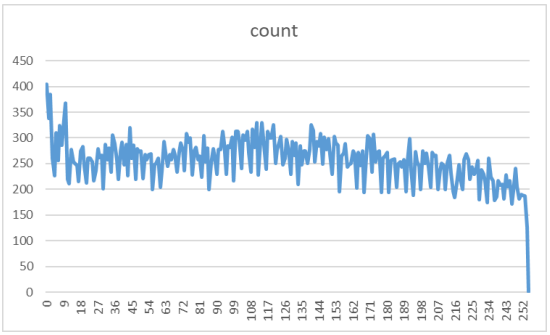
Window size 由左至右，上至下: 3, 7, 15, 21, 31, 45, 51, 71 (先只比較 Hd)

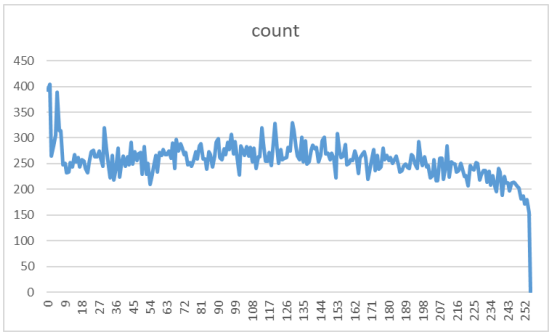


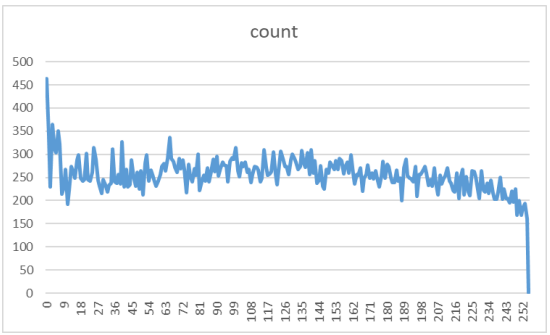


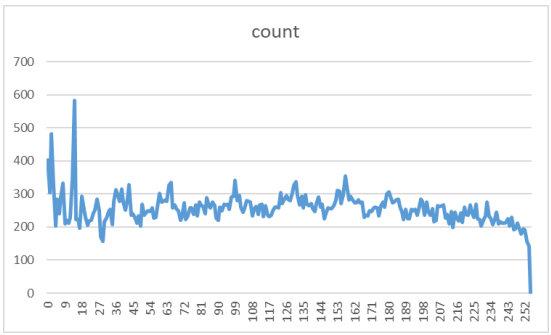




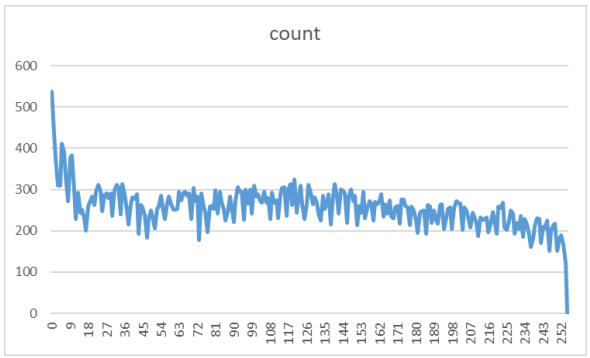








Le window size = 31



4) Discussion of results: 使用不同的 window size 會讓結果有所不同。比較不同

window size 的直方圖可以發現，當 window size 越大，每一種 pixel 的個數會越相近，看起來越 uniform。(當然，程式執行的時間也會越久)。若只比較.raw 檔顯示的圖，可以發現當 window size 過小時，出來的結果並不好，只有手掌的地方看得比較清楚，但是當 window size 大於 21 之後，產生出來的貓咪都頗清楚(我用肉眼很難看出分別)。若同時考量成像結果與運算的 complexity，我選擇 window size = 31。

f) 比較 global histogram equalization 和 local histogram equalization: global HE 是針對整張圖做，所以不需要指定 window size，而 local HE，是針對區域做，所以需要給 window size。就直方圖的結果來看，global HE 會大致保持原來 pixel value 的分布情況，但 local HE 則可以做到非常 uniform。

PROBLEM 2: NOISE REMOVAL

a) 為 3b, 3c 影像設計適合的 noise removal filter，輸出 N1, N2

1) Your motivation and approach: 根據老師上課所教的，3b 看起來像 Gaussian noise (uniform noise)，3c 看起來像 salt and pepper noise (impulse noise)，所以對 3b 使用 low-pass filter，對 3c 使用 non-linear filter (median filter)。

Low pass filter 的實踐步驟: 使用 3x3 大小的 filter size，傳入指定的 weight (b)，根據老師講義上的 general form: $\frac{1}{(b+2)^2}\begin{matrix}1 & b & 1\\ b & b^2 & b\\ 1 & b & 1\end{matrix}$ 產生對應的 mask，再根據 convolution 的步驟，將 mask 裡面的 value 分別乘以原圖的 neighbor pixel value，再相加，再除以$\frac{1}{(b+2)^2}$，最後把值 assign 給當前位置。

Median filter 的實踐步驟: 傳入指定的 neighbor 大小(filter size)，將每一個位置的 neighbor 放到 vector 進行排序，再將中位數 assign 給此位置。

2) Original images (optional) 左: 3b 右: 3c

3) Output images

對 3b 使用 low pass filter, weight = 1, 2, 3, 4, 5, 6 (由左至右，由上至下)

對 3c 使用 median filter, filter size = 3, 7, 9 (由左至右)

4) Discussion of results: 從上面的結果圖，對 3b 使用 low pass filter 後，其實會發現，不論 weight 調多少，肉眼看起來的結果好像差不多，但是當計算 PSNR 後可發現，原圖 3a 與 3b 的 PSNR=23.4909，原圖 3a 與 weight = 1, 2, 3, 4, 5, 6 計算的 PSNR 分別為: 25.6475, 26.9133, 27.3347, 27.3413, 27.1637, 26.9282。可以發現當 weight 變大時，PSNR 會上升，但上升到一定程度後，weight 再增加，PSNR 卻會下降，所以我取 PSNR 最高的 weight=4 輸出。 (PSNR 越大越好)

相對的，對 3c 使用 median filter 後，其實可以明顯的發現，filter size=3 時效果最好，當 filter size 再繼續成長時，會變得越來越模糊。比較原圖 3a 與 3c 的 PSNR= 14.707，比較原圖 3a 與 filter size = 3, 7, 9 的 PSNR 分別為: 30.6107, 27.5165, 26.1671，可以發現 filter size=3 最好，與肉眼觀察的結果相同。

b) 計算 N1, N2 的 PSNR

1) Your motivation and approach: PSNR 的算法: 根據老師講義上的公式，先求原始圖與欲量測的圖的 mean square error (差值->再平方->再除以圖形大小 w*h)，再根據公式$\mathrm{PSNR} = 10\log(\frac{255^2}{MSE})$ 求出 PSNR。

3) 輸出結果:

3a 與 3b => PSNR=23.4909 dB

3a 與 N1 (weight =4) => PSNR=27.3413 dB

3a 與 3b => PSNR= 14.707 dB

3a 與 N2 (filter size =3) => PSNR= 30.6107 dB

4) Discussion of results: 比較 PSNR 後可以發現，對 salt and pepper noise 使用 median noise 的噪音移除效果非常好，PSNR 成長了很多；相較之下，對 uniform noise 使用 low pass filter，雖然 PSNR 也增加了，但效果沒那麼明顯。

Reference:

[1] http://chu246.blogspot.com/2016/09/octave-excel-histogram.html

[2] http://ccy.dd.ncu.edu.tw/~chen/course/vision/ch2/ans02.pdf