

Homework #1

Environment

Matlab_R2018a

WARM-UP

把 RGB 值相加除以三作為灰階值，再在輸出時在 `writeraw` 加上 `transpose` (`writeraw(B.', "B.raw")`)。結果如附圖(Figure 1)



FIGURE 1



FIGURE 2

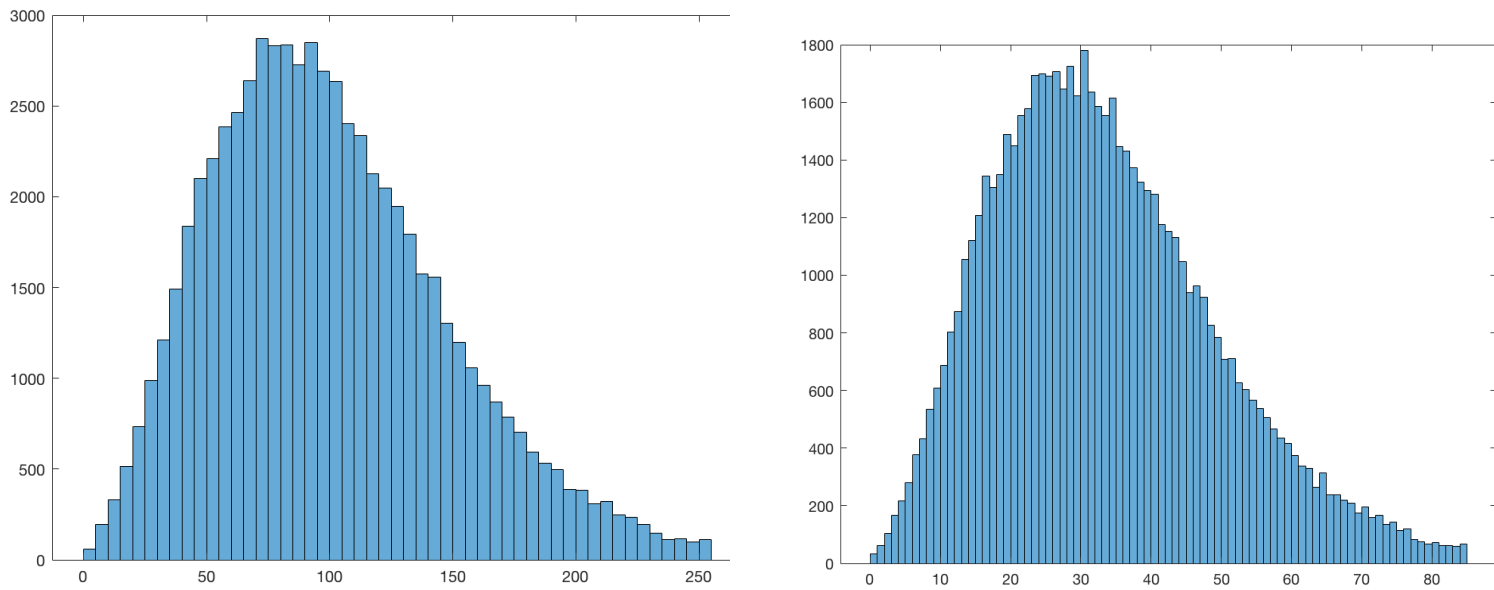
Problem 1

(a)

直接把 D 設為 I_2 即得所求(Figure 2)

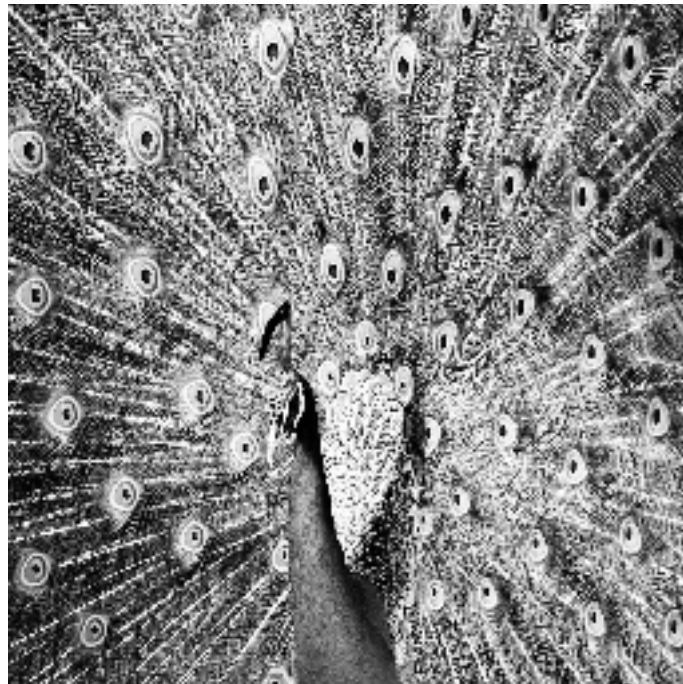
(b)

左邊的 Histogram 為 I_2 右邊的為 D ，從圖中可以看出來 D 更為集中，範圍更小，同時單一區段的次數也因為區段劃分變細而變得不明顯。



(c)

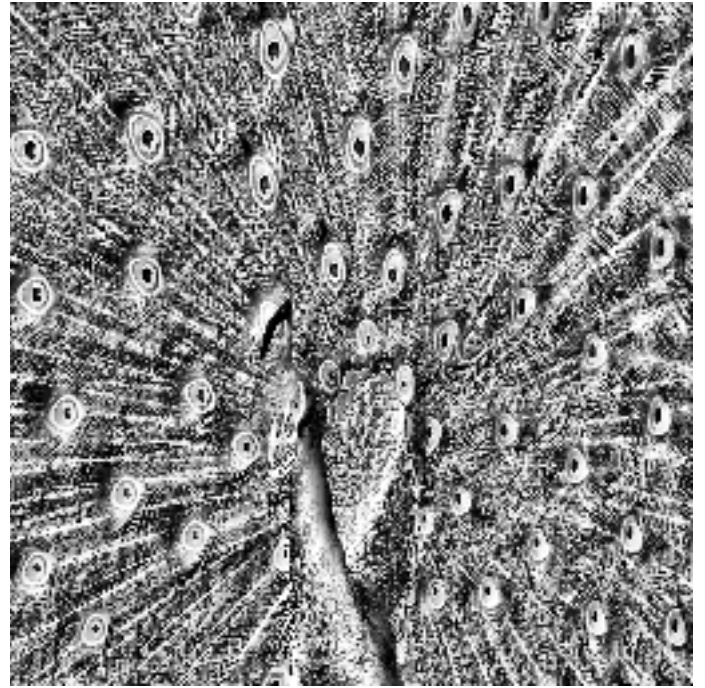
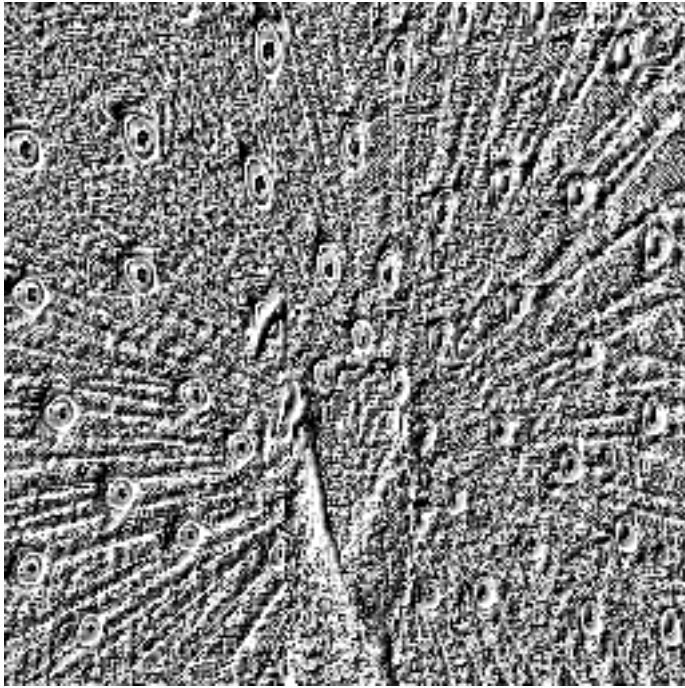
用上課教的方法做 Global Histogram Equalization。先統計各個數值的累計出現次數，在用累計出現次數得到轉換函式已得到 H 。



(d)

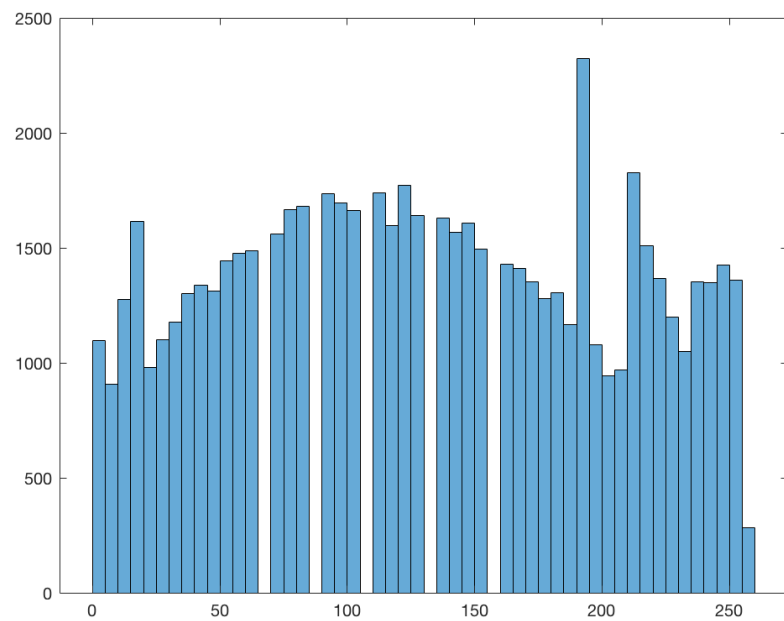
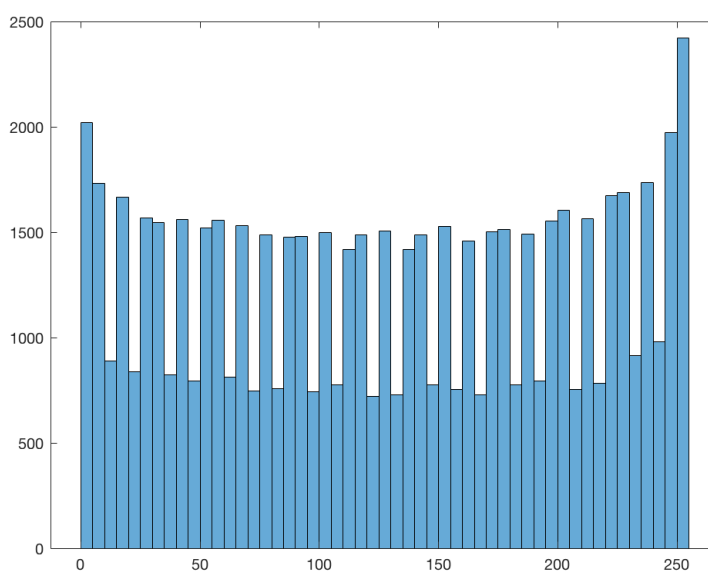
分 L_3 跟 L_9 兩部分分別算以 3×3 跟 9×9 計算的 Local Histogram Equalization，裡面每一個 pixel 的數值都為 $(27/81)$ 剪掉它在附近的排名 $\times 255/81$ ，即如果它是最亮的點，它的值會是 255。

左為 L_3 右為 L_9

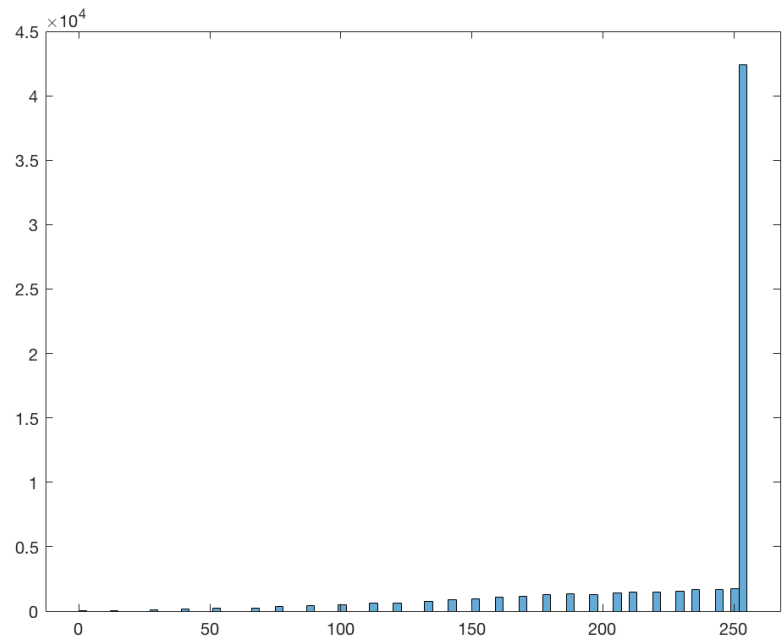
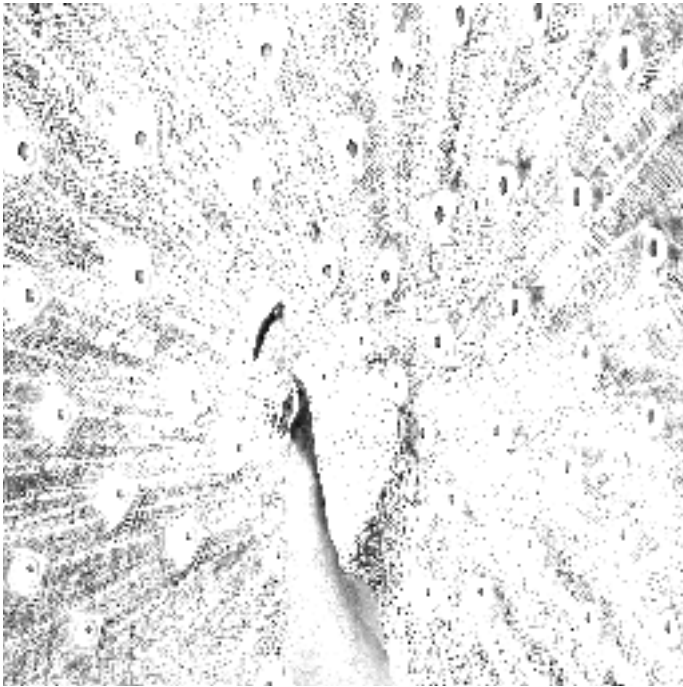


(e)

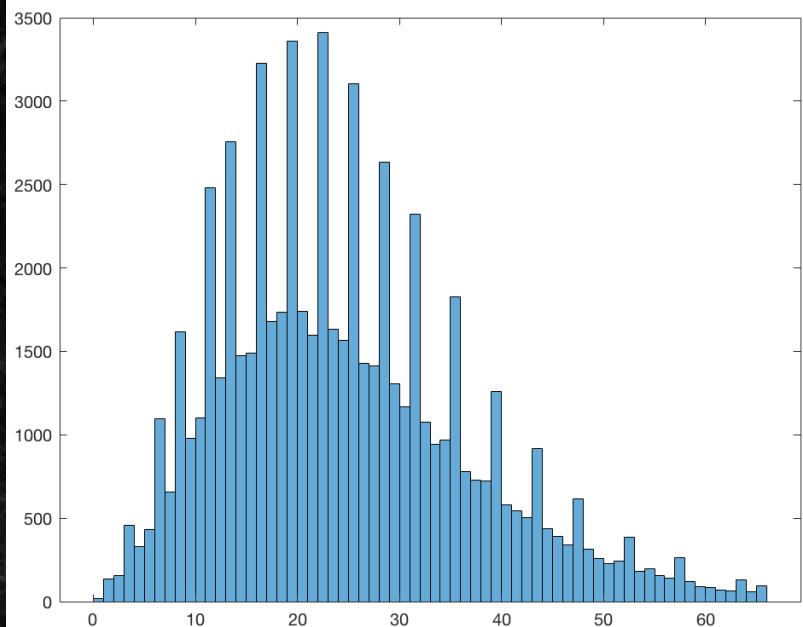
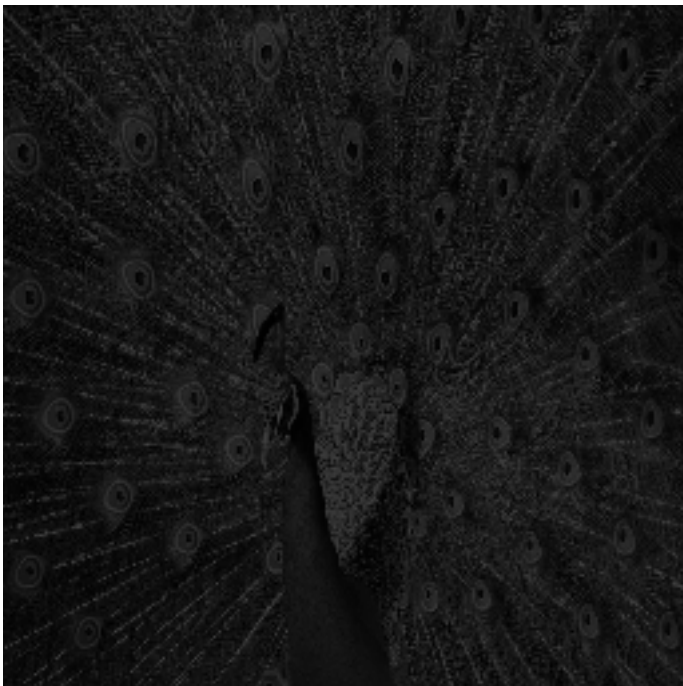
左邊 L_9 右邊 H ，我們可以看到因為算法的特性， H 的圖有著跟原本比較像(形狀上)的分佈，同時基本上除了極高的地方之外都蠻平均的。而在 L_9 中因為我把色彩分配在 $255/81 \cdot n$, $n \leq 81$ 的點上，所以 Histogram 會看起來有一條多一條少的情況($255/81 \approx 3.14$)。



(f)

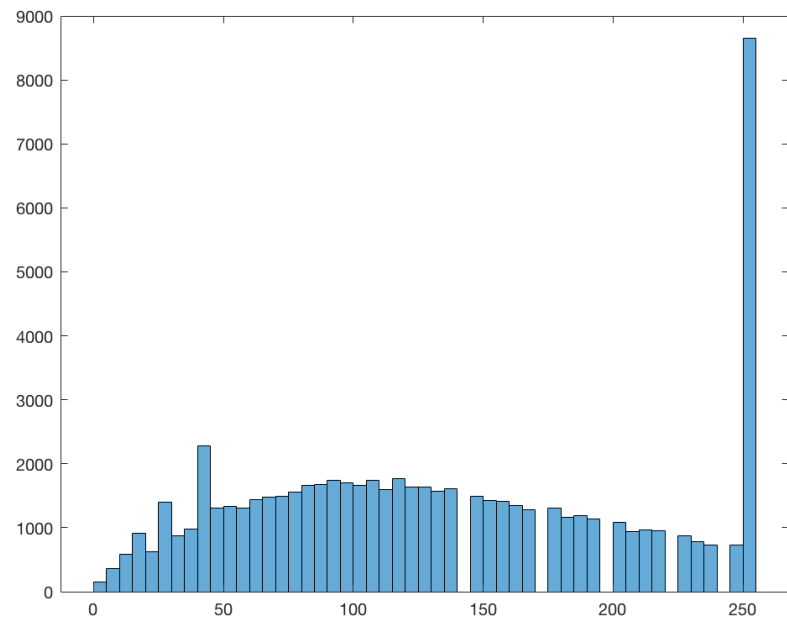
Log Transform

Log Transmission 的目的是要把顏色深的部分變淺，把深的小區塊經過 transmission 後變得大塊。在試過幾個常數 a 之後，我最後用的是 $a = 10$ 來保證不會過淡或因為太深看不出細微的暗部細節。

Inverse Log Transform

Inverse Log 會反過來加暗亮部，但因為 $\text{Input}(D)$ 的範圍只有原本的三分之一，故不是很適合 Inverse log transform。

Power-law Transform



在這裡取 $G = D$ 的每個元素的 1.4 次方，在經過 power-law transform 處理後，原本被壓成三分之一的範圍會被拉大，所以會變得比較容易分辨，試過幾個次方後，最後選的是 1.4 次方。

Problem 2

(I)

(a) 左邊的 σ 是 10，右邊的是 20，右邊的噪點比左邊的明顯很多。



(b)左邊的是隨機 256x256 中大於等於 253 為 pepper，小於等於 3 為 salt，右邊的是隨機 256x256 中大於等於 250 為 pepper，小於等於 5 為 salt。



(c)

Gaussian: 以下分別測試講義中提到的三種濾鏡及 odd 和 even 兩種處理方式，效果最好的是

$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix}$ ，而 odd 和 even 看起來並沒有差別，做法：在原圖中從初始點開始取 3x3

並把 3x3 的小 matrix 和 low-pass filtering 單項相乘的的值和作為取的那點的新值，如此便的所求。



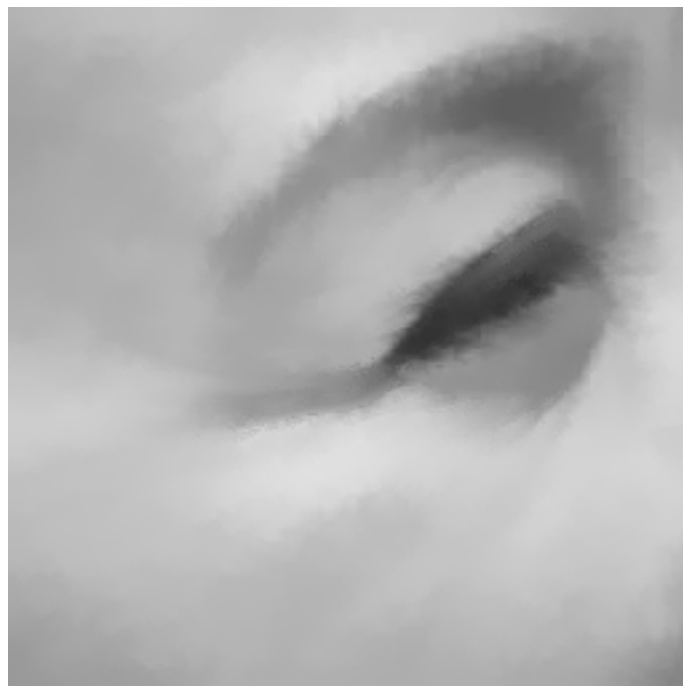
Salt-and-Pepper: 為了消除 Salt-and-Pepper noise，首先我在歷遍每一個點時把他為中心的 3×3 ，把 3×3 中除了中心以外的值放進 array 中，然後算他們的平均和標準差，若中央點減平均大於標準差，那麼就把那點的值設為平均值。如此一來就完成了 Salt-and-Pepper 的 removal。



(d) $PSNR_S = 30.2230$, $PSNR_G = 28.2788$

$PSNR_S$ 跟原圖比 $PSNR_G$ 更接近，因為 $PSNR$ 比較大，這跟看起來的結果相符

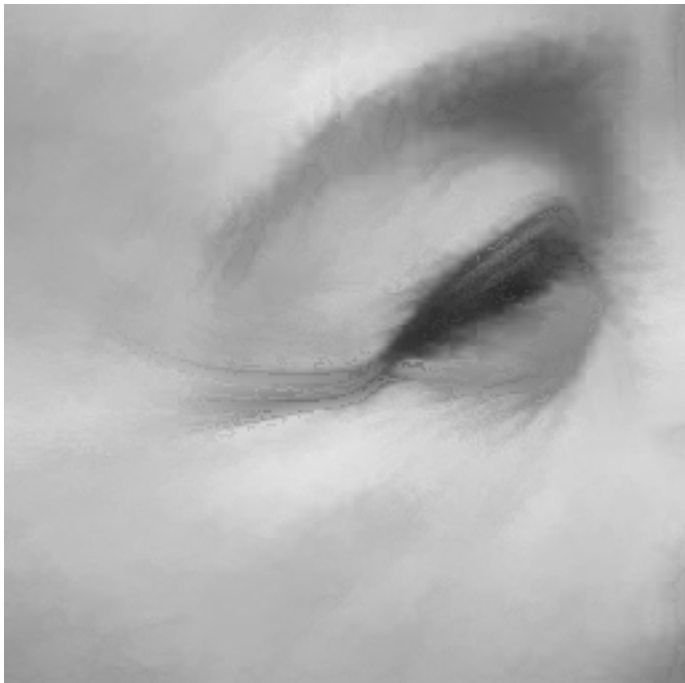
(II)



處理方式：用大範圍的 Salt-and-Pepper(Impulse Noise Removal) 來處理，首先先跑一次 5x5 的來消除一些比較小的皺紋。



接著跑一次更大的 13x13 來把比較大的皺紋清掉，接著還要跑一次 5x5 去把一些剛剛造成的噪點去除

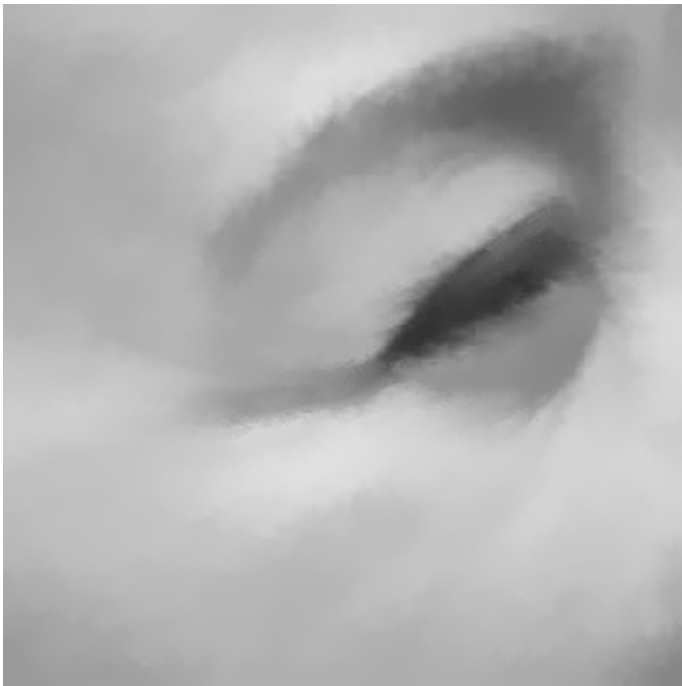


5x5 + 13x13



5x5 + 13x13 + 5x5

接著再試著跑一層 13x13 跟 5x5 試著看看會不會更好，在魚尾紋上方的皺紋看起來有更不明顯一點，後面再加就沒什麼差別了。



5x5 + 13x13 + 5x5 + 13x13



5x5 + 13x13 + 5x5 + 13x13 + 5x5