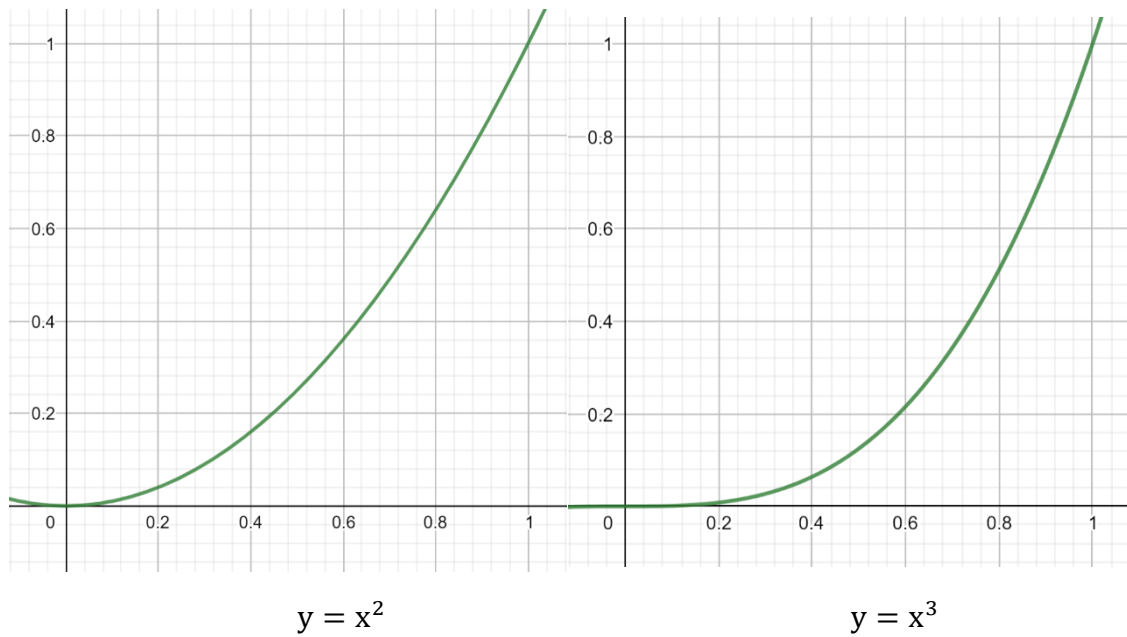


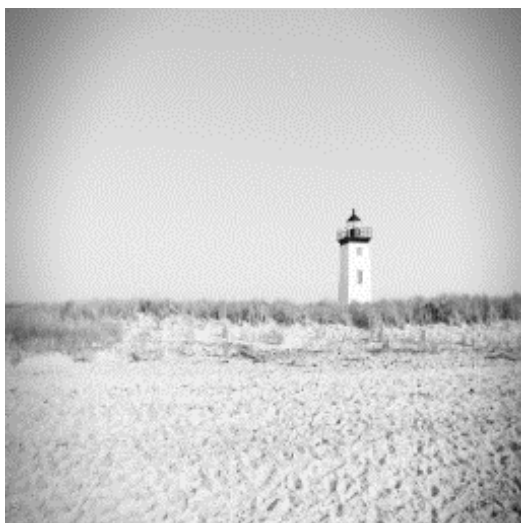
1. Warm-up: simple manipulations

a. Motivation and approach

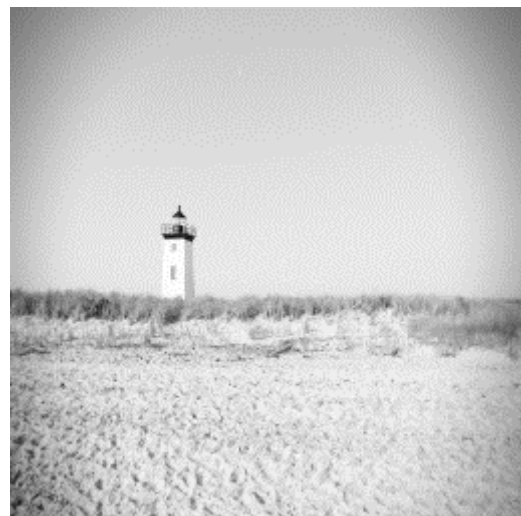
原圖整體偏亮, 因此考慮使用 power-law 轉換時, 指數應為大於 1 的實數
下圖所示, 當指數越大函數在亮點處變動越大, 暗點處則變動越不明顯



b. Original image

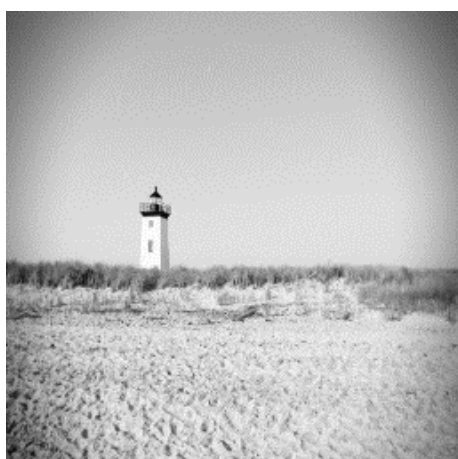


原圖

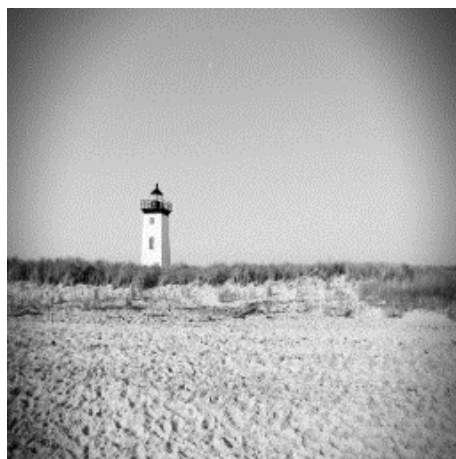


左右翻轉後的 B 圖

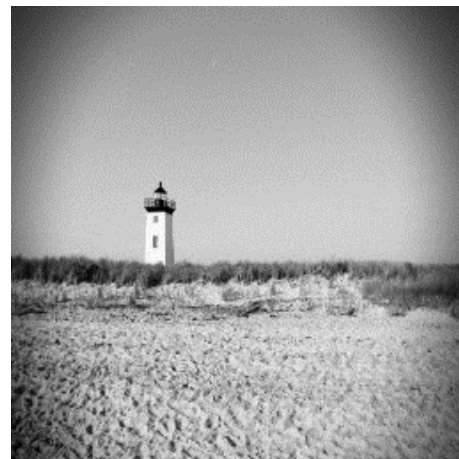
c. Output images



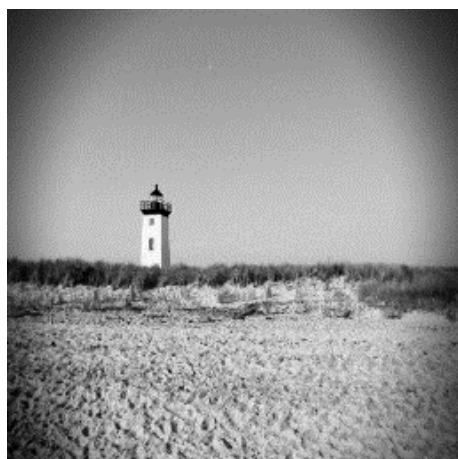
指數 = 1.4



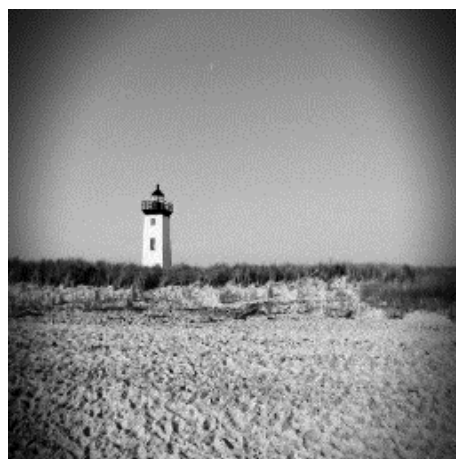
指數 = 1.8



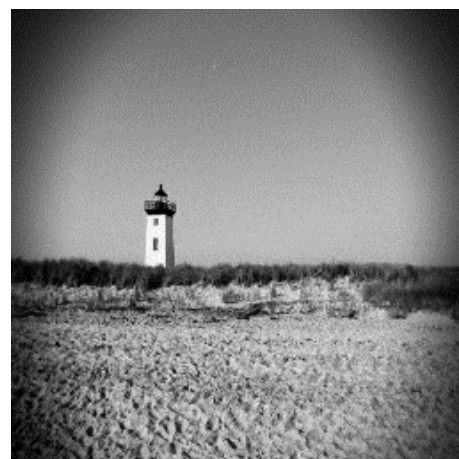
指數 = 2.2



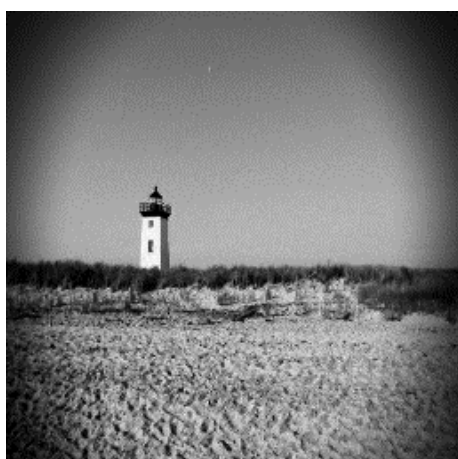
指數 = 2.6



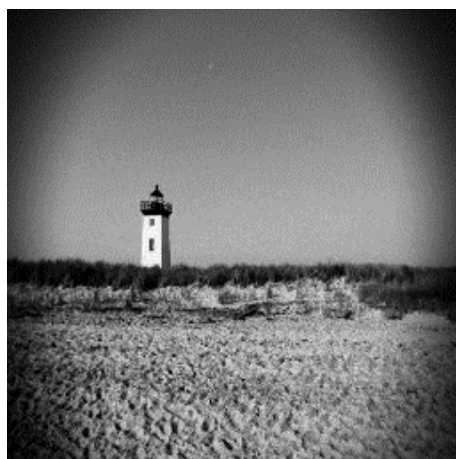
指數 = 3.0



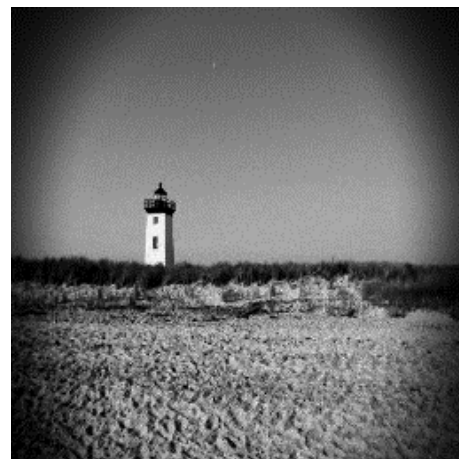
指數 = 3.4



指數 = 3.8



指數 = 4.2



指數 = 4.6

d. Discussion of results

圖片上方其實有顆星星，大約在指數為 3.0 開始變得清楚
另外燈塔的光影變化，在指數漸增後陰影也變得明顯，
草地的紋路也隨著指數漸漸清晰
但是指數增大會伴隨外圍陰影越來越大的副作用，
整體的亮度也越來越暗使得圖片不易辨識
因此最後折衷選擇 3.4 為最佳轉換參數

2. Problem 1: image enhancement

a. Motivation and approach

將原圖所有的 intensity value 除以 2 或 3 產生亮度偏低的 D 和 E
使用 histogram equalization 重新分配亮度值
讓整體圖案的亮暗對比度增大可使圖片細節更易辨識

$$h(v) = \text{round} \left(\frac{cdf(v) - cdf_{min}}{(M \times N) - cdf_{min}} \times (L - 1) \right)$$

取自維基百科: https://en.wikipedia.org/wiki/Histogram_equalization

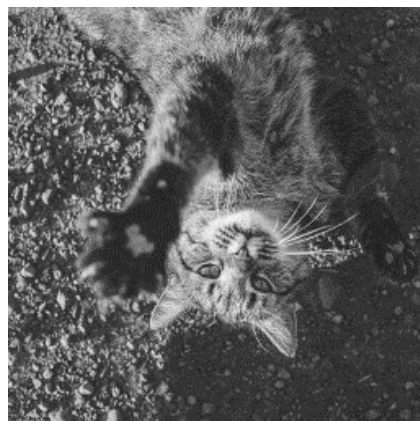
$h(v)$ 是一個將原圖亮度值對應到新亮度值的遞增函數

L 是最大可能亮度(以 8-bit image 為例則是 256)

M, N 是 圖的長寬大小

cdf_{min} 是整張圖實際存在的最小亮度值

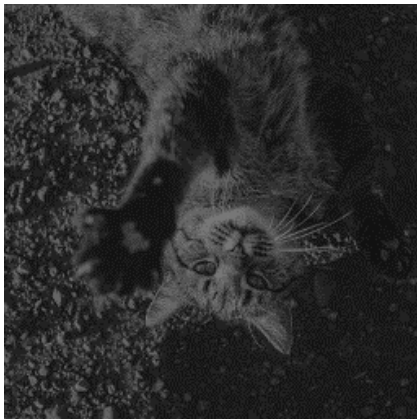
b. Original images



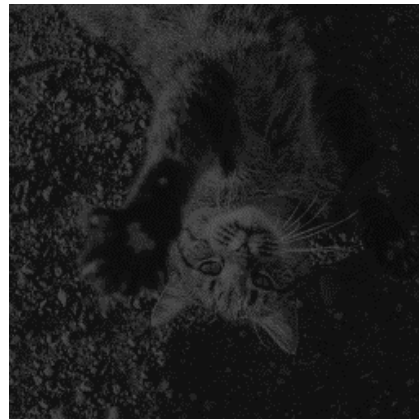
sample2.raw

c. Output images

d.raw



e.raw



globalD.raw



globalE.raw



localD.raw



locale.raw

d. Discussion of results

由最左邊三個 histogram (I2, D, E)可看出若只是將整體 intensity value 除以某值則 histogram 形狀不變只是整體向左壓縮

中間是 global histogram equalization 後的 histogram

可以發現 histogram 形狀依然不變只是較高的地方比較稀疏
而 D 和 E 雖源自同一張圖，global equalization 後卻不同，
應該是因為在除法時會因為亮度值為 digital，
在四捨五入時導致無法復原的破壞，
不過在圖片上差異不大

最右邊是 local histogram equalization 後的 histogram

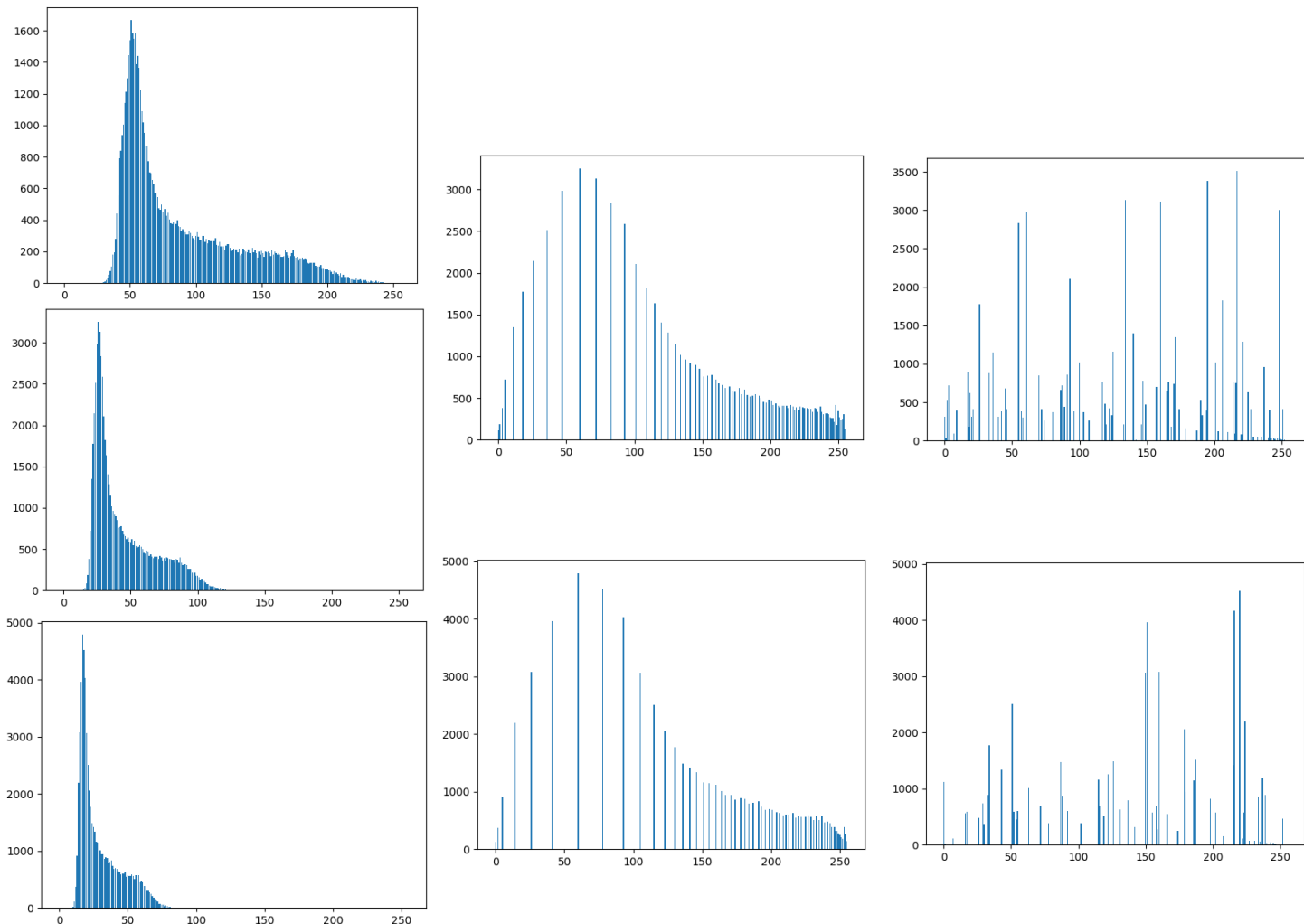
由於是切割成 16 塊等大小圖個別做 histogram equalization

最後合起來的 histogram 已與原本完全不同

圖案上也可以看得出來雖然個別小圖中的圖案更能清晰辨識
(例如貓臉不會過白)

但是小圖間有很明顯拼湊痕跡

這樣不連續的結果顯示 local histogram equalization 並不是個好的方法



3. Problem 2: noise removal

a. Motivation and approach

sample4.raw 應該是 sample3.raw 加上 uniform noise 的結果

sample5.raw 則應該是 加上 impulse noise 的結果

對上述兩張圖片都嘗試三種 noise removing 方法:

1. low-pass filtering: 先延展邊界再用 filter 做 convolution 得到新亮度值

$$\frac{1}{10} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad \frac{1}{84} \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 2 & 3 & 3 & 3 & 2 & 1 \\ 1 & 2 & 3 & 4 & 3 & 2 & 1 \\ 1 & 2 & 3 & 3 & 3 & 2 & 1 \\ 1 & 2 & 2 & 2 & 2 & 2 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

3 X 3 7 X 7

2. median filtering: 先延展邊界後再在特定大小內找中位數做新亮度值
分別做 3 X 3 與 7 X 7

3. outlier detection:

ϵ 挑選 20, 80, 160 三種

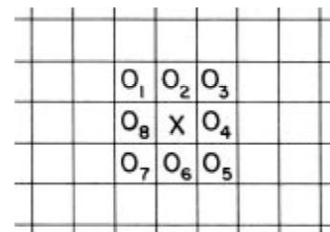
b. Original images

sample3.raw



(2.67)

sample4.raw

















$$\text{if } \left| x - \frac{1}{8} \sum_{i=1}^8 O_i \right| > \epsilon \quad \text{then } x = \frac{1}{8} \sum_{i=1}^8 O_i$$

(2.65)

sample5.raw



c. Output images				
	sample4		sample5	
lowpass	(2.74) 3 X 3	(2.79) 7 X 7	(2.92) 3 X 3	(3.00) 7 X 7
				
median	(2.79) 3 X 3	(2.79) 7 X 7	(3.00) 3 X 3	(3.00) 7 X 7
				
outlier	20 (2.68)		20 (2.91)	
				
	80 (2.67)	160 (2.67)	80 (2.60)	160 (2.54)

d. Discussion of results

low-pass 和 median 在 filter size 增大後新圖比較少雜訊但也比較模糊
low-pass 在 impulse noise 中效果非常差
因為 impulse noise 個數不多值卻非常極端，在平均時仍對整體影響過大
outlier 則可以在 impulse noise 中顯示非常好的效果
但是在 uniform noise 時則不太有效果
另外當選擇越小的閾值新圖越接近 low-pass filtering
但當閾值過大則會造成過多 noise 被保留

還有一個有趣的現象

就是閾值 80 的 outlier 在 impulse noise 上雖然肉眼上看起來比 low-pass 好
但是 PSNR 上卻顯示 low-pass 的比較高
顯見 PSNR 只是參考標準 與人眼的相似度還是有差異

最後選擇 low-pass 7 X 7 的結果作為最終 output N1

outlier 80 作為 N2

N1 4_low7.raw



N2 5_out80.raw

