

# Report DIP HW1

B07902144 彭約博 資工三

## Problem 1 Warm up

(a)

▼ approach

將圖片讀入後，用2層for迴圈將3頻道(0,1,2) 每個pixel分別以(299,587,114)的比例相加後，再加上500摒除以1000，並把每個pixel賦值到一個1頻道的圖。

- sample1.jpg
- 1\_result.jpg



(b)

▼ approach

將第2個維度第*i*個pixel和同維度的第559-*i*個pixel對調。

- sample1.jpg
- 2\_result.jpg



---

## Problem 2 Enhancement

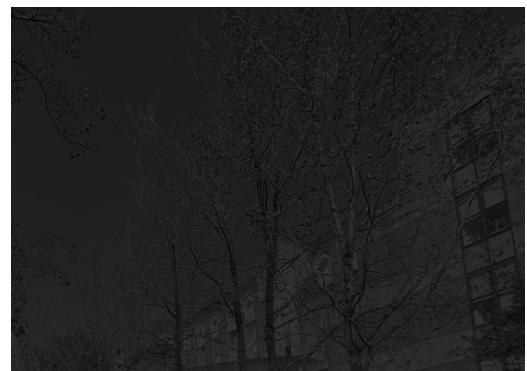
---

(a)

▼ approach

所有pixel乘以5。

- sample2.jpg
- 3\_result.jpg



(b)

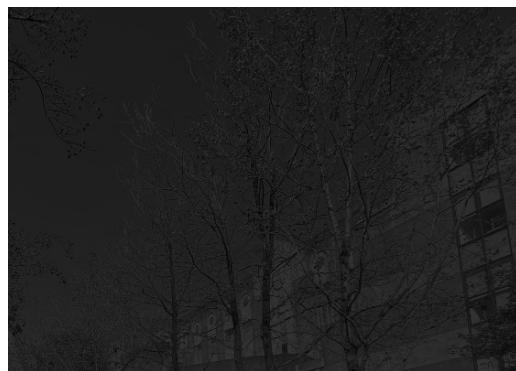
▼ approach

所有pixel除以5。

- Sample2.jpg



- 3\_result.jpg



- 4\_result.jpg



---

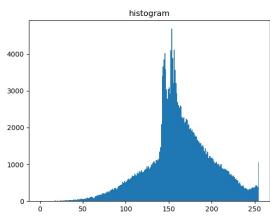
## (c)

### ▼ 討論

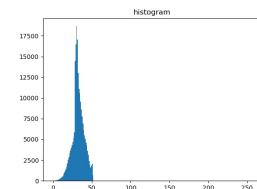
可看出原圖的histogram，在所有pixel除以5時，會有些值變為0而損失，pixel之分部也在intensity較低處，故圖片較暗。

又再將3\_result乘以5時，由於上述因素損失了一些pixel的值(變為0)，故histogram看起來會有間隔，但大致上看起來和原圖不會差太多

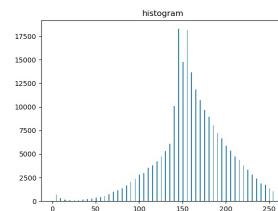
- sample2 histogram



- 3\_result histogram



- 4\_result histogram



**(d)**

▼ approach

先將histogram算出來，並且從最小的intensity (0)累加到最大(255)的 cumulative distribution function(CDF)，然後從然後建立一個轉移函式 $T(i)$ ，值為0~255:

$$T(i) = (CDF(i) - CDF_{min})255 / (CDF_{max} - CDF_{min})$$

並且使用rounding

再將原圖所有值將過 $T(i)$ 轉移到新的值。

- sample3.jpg



- 5\_result.jpg



**(e)**

▼ approach

首先對原圖padding出符合kernel可以對每個pixel卷積的大小，再利用不同的 window size對原圖進行卷積，計算出在window 中，對於所有點，其中央的

點之亮度應該為這些點中的第幾大(rank)，最後將此rank 乘以 這個window中出現的最大intensity，並除以這個window的點的數量，即完成此kernel中央的pixel的更新。

(備註:計算量取決於window size的大小)

#### ▼ 討論

使用了許多不同的window做了測試，較小的window size對於細節部分凸顯的部分較少，而較大的window size，較可以凸顯出細節，然而若到了如同 $81 \times 81$ 這樣大的size時，不僅需要花許多時間進行計算，且其在某些細節上的呈現反倒不如 $51 \times 51$ 的多(例如圖左下方的車子)，且亮部會過亮而模糊。

其中我個人最喜歡 $31 \times 31$ ，對於細節掌控的比較剛好，且看起來比較自然。

- sample3.jpg



- 6\_result.jpg ( $31 \times 31$ )



#### 不同window size

- 6\_result.jpg ( $3 \times 3$ )



- 6\_result.jpg ( $7 \times 7$ )



- 6\_result.jpg (11×11)



- 6\_result.jpg (15×15)



- 6\_result.jpg (21×21)



- 6\_result.jpg (41×41)



- 6\_result.jpg (51×51)



- 6\_result.jpg (81×81)



---

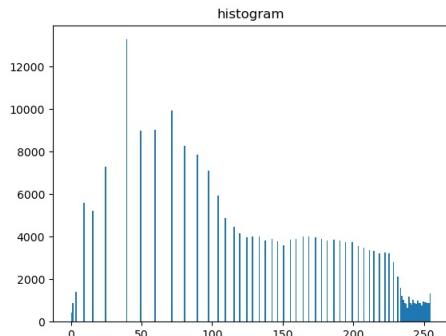
(f)

▼ 討論

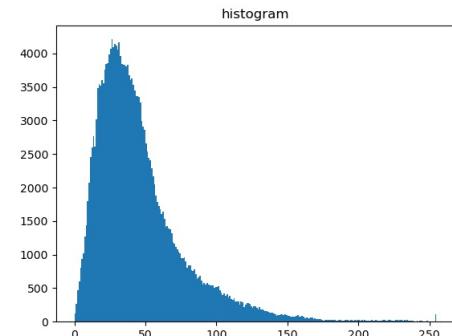
可看出原本經由Global HE的histogram，以整張圖來說都很平均，亮部的點雖然較少，但有較多不同的值，暗部的點較多，但不同的值卻較少，整張圖出現類似過曝的狀態。

對比Local HE的histogram，大部分的值都偏移到暗部，保留較多原圖的色調，但凸顯了某些細節的對比。

- 5\_result histogram



- 6\_result histogram



## (g)

### ▼ approach

主要嘗試了兩種方法

- Power-law(Gamma) transformation
- Log transformation

### ▼ Power-law(Gamma) transformation

數學式:  $s = cr^{Gamma}$

將圖中的pixel代入上式的r，再輸出s到新的圖

(備註: 這裡實作時c=1)

### ▼ Log transformation

數學式:  $s = c \log(1 + r)$

將圖中的pixel代入上式的r，再輸出s到新的圖

### ▼ 討論

使用上述兩種不同的方式進行Enhance，其中我認為Power-law的效果較好，Gamma值較小時，會將較暗的輸入值對應至較寬的輸出範圍，讓其暗部變亮，但不會損失太多亮部的對比，Gamma=0.45時可以清楚看到船身的本體。

Log transformation 將整張圖片的亮度分布在intensity較高的地方，故圖片看起來細節都很清楚，但對比度就不高，覺得不是很好看。

我個人最喜歡Gamma=0.55的Power-law(Gamma) transformation，其有顯現出船的一些細節，又沒有損失太多小島的對比，且海浪的水波紋路較Gamma=0.45清楚。

- sample4.jpg



- 7\_result.jpg (0.55)



### 不同Gamma值

- 0.45



- 0.55



- 0.65

- 0.75



- 0.85



## 不同C值

- 40



- 60



- 80

- 100



---

## Problem 3 Noise removal

(a)

- ▼ approach
  - ▼ 針對sample6使用 low-pass filter

### General form

$$H = \frac{1}{(b+2)^2} \begin{bmatrix} 1 & b & 1 \\ b & b^2 & b \\ 1 & b & 1 \end{bmatrix}$$

使用課堂中之kernel公式，且 $b=2.35$  (經過多次測試PSNR值最高)。

- ▼ 針對sample7使用 median filter
  - 使用課堂中之PMED，將每一個window的9個pixels 展平成為1D的list，並將其以三個為一單位，分別計算MAXMIN、MINMAX，再取其平均並賦值到kernel中央的pixel上。

(兩種方式都會先對原圖進行padding，以讓kernel能夠對每個pixel進行操作)

- ▼ 討論

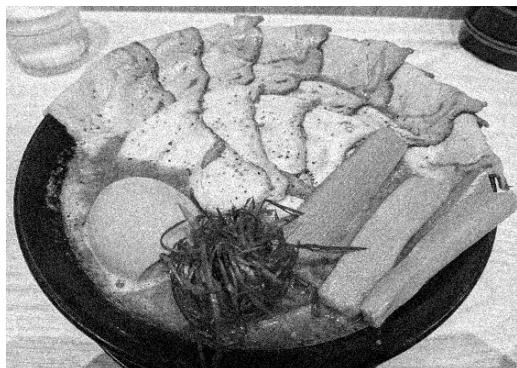
- ▼ Gaussian noise (sample6)

經過low-pass filter後，雖然雜訊看起來有去除一些，但整張圖片也變得比較模糊，其中選擇 $b=2.35$ 是因為反覆測試多次後，發現這樣的setting下2.35的PSNR值最高，故選之。

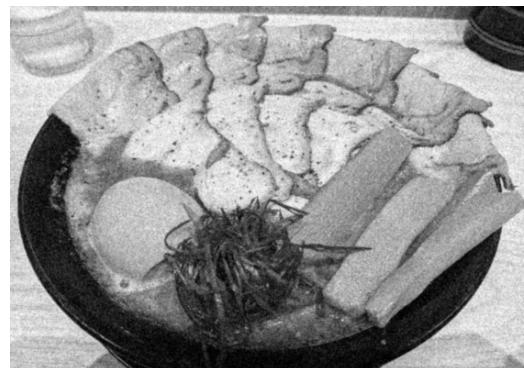
▼ Salt-and-pepper noise (sample7)

PMED的filter對去除此類雜訊的效果很好，然而只處理一次的話，然會有些許點雜訊，故我重複對圖片進行了4次noise removal，才出現下圖比較乾淨的圖，然而缺點是，有些細節也損失了，甚至原本叉燒上的胡椒鹽也一併被去除

- sample6.jpg



- 8\_result.jpg



- sample7.jpg



- 9\_result.jpg



---

(b)

▼ 討論

根據維基百科中對PSNR的描述：

## 數值代表的意義

圖像與影像壓縮中典型的峰值訊噪比值在 30dB 到 50dB 之間，愈高愈好。

- PSNR接近 50dB，代表壓縮後的圖像僅有些許非常小的誤差。
- PSNR大於 30dB，人眼很難查覺壓縮後和原始影像的差異。
- PSNR介於 20dB 到 30dB 之間，人眼就可以察覺出圖像的差異。
- PSNR介於 10dB 到 20dB 之間，人眼還是可以用肉眼看出來這個圖像原始的結構，且直觀上會判斷兩張圖像不存在很大的差異。
- PSNR低於 10dB，人類很難用肉眼去判斷兩個圖像是否為相同，一個圖像是否為另一個圖像的壓縮結果。

本次經過去雜訊的圖結果都介於25~30之間，的確以肉眼就可以看出和原乾淨的圖的差異，不算是效果很好的去雜訊處理，在比較去雜訊之結果時，對於 8\_result的圖片較難以區分不同的b值在去雜訊的效果，故在選擇參數時提供了很好的幫助。

- sample5.jpg



- PSNR = 27.412771

- PSNR = 25.635737

