



DIP Course

Homework Report week #1

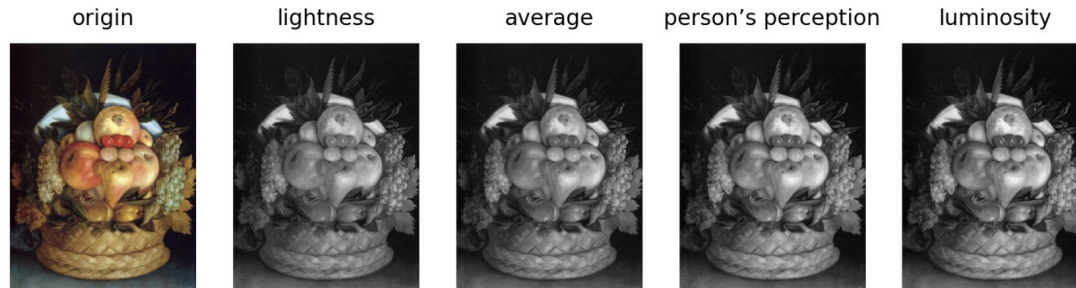
Student: p11922004 任祖頤

Outline

- Grayscale image transformation
- Vertical flipping
- Histogram comparisons
- Image Enhancement
- Noise Removal

Grayscale image transformation

- 4 Grayscale transformation formulas applied
 - lightness: $(\max(R, G, B) + \min(R, G, B)) / 2$.
 - average: $(R + G + B) / 3$
 - person's perception of brightness: $0.2989 * R + 0.5870 * G + 0.1140 * B$
 - luminosity: $0.21 R + 0.72 G + 0.07 B$





Grayscale image transformation

- 差異比較
 - 四種方法可以看出 lightness 的明暗對比最低, luminosity 對比最高
- 討論
 - luminosity 之所以明暗強烈, 是因為它將人眼最敏感的綠色佔比提高到了 0.72 倍
 - person's perception of brightness 是採用人眼對顏色感知受器的比例來設計, 看起來比較貼近彩色原圖, 因此最後採用此種
- 延伸發現
 - cv2.imread 預設的色彩順序為 BGR, 需要使用 cv2.COLOR_BGR2RGB 轉換過比較好進行操作

Vertical flipping

- 延伸發現

- 兩邊數值對調時可以使用單行的 expression 就不用再給一個 temp 變數

```
img[i], img[-i] = np.copy(img[-i]), np.copy(img[i])
```

origin



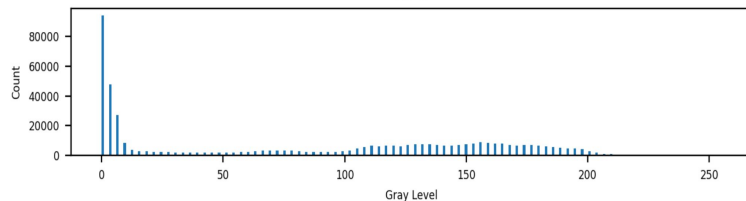
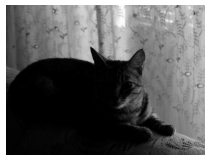
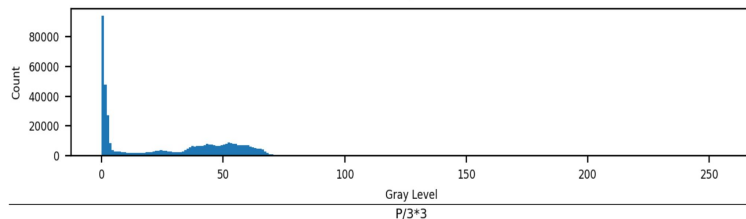
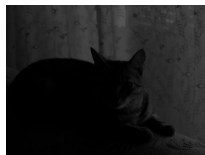
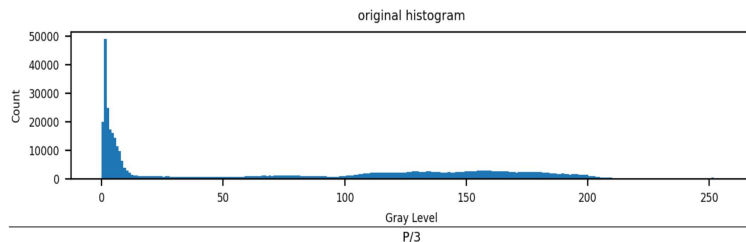
vertical flipping



Histogram comparison

- 差異比較

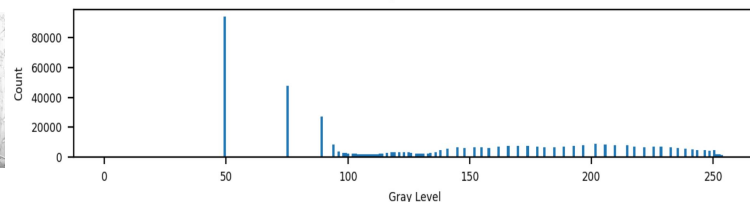
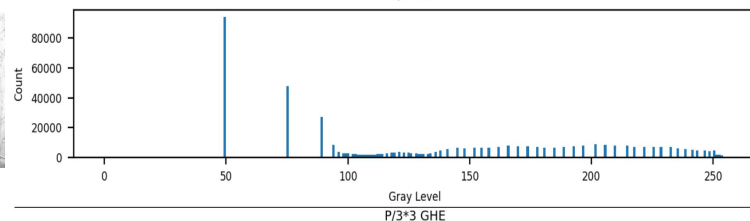
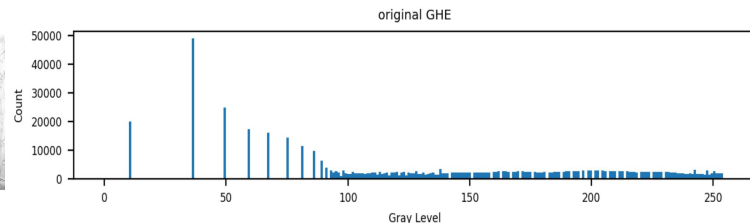
- 原本圖片 P 中, 每個 pixel 有 256 個 grayscale level 來表達一個顏色
- 進行 $P/3$ 之後強制把 256 level 壓縮成 85 個 level, 每 3 個 level 的 pixel 就會被整合在一起
- $P/3*3$ 之後可以看出非 3 的倍數 pixel 值都被整併到最近的 3 倍數 pixel 值之中, 因此畫面最亮跟最暗的數量有稍微提高



Global Histogram Equalization Comparison

- 差異比較

- 原始圖片進行 GHE 處理後保留較多細節
- P/3 與 P/3*3 因為把 grayscale level 非 3 的倍數的 pixels 都整合到 3 的倍數之中，在經過 GHE 處理後相較原始圖片會遺失較多細節(毛髮最為明顯)
- P/3 與 P/3*3 之間因為並沒有改變個別 grayscale level 的 pixel 數量，因此經過 GHE 計算後會得到一模一樣的結果





Global Histogram Equalization Comparison

- 討論

- 原本的做法是先將所有 pixel 看過一次之後建立一個對照表, 再逐一把每個 pixel 依照對照表來改變 grayscale level, 時間複雜度約為 $O(3 \cdot (n^2))$
- 後來參考 [文獻](#) 將內文提到的數學模型套入, 成功將複雜度降低為 $O(2 \cdot (n^2) + n)$

$$p_n = \frac{\text{number of pixels with intensity } n}{\text{total number of pixels}} \quad n = 0, 1, \dots, L-1. \quad g_{i,j} = \text{floor}\left((L-1) \sum_{n=0}^{f_{i,j}} p_n\right),$$

- 可知, 若有做好數學建模就有機會降低運算成本

Local Histogram Equalization Comparison

- 討論

- [LHE 參考方法](#)
- 我找到的 LHE 演算法是每個 pixel 會跟四周圍比較, 並且依照結果進行倍率增加, 但跑完之後數值會很極端 (histogram 都集中在兩側), 猜測可能有更好的 LHE algo 但時間不太夠了就沒有額外嘗試

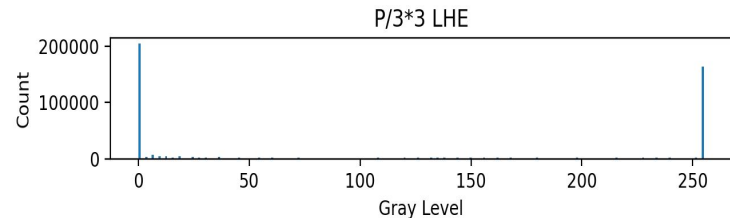
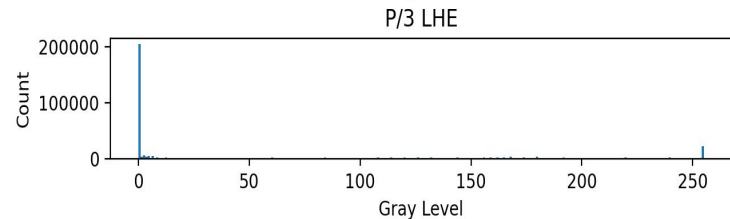
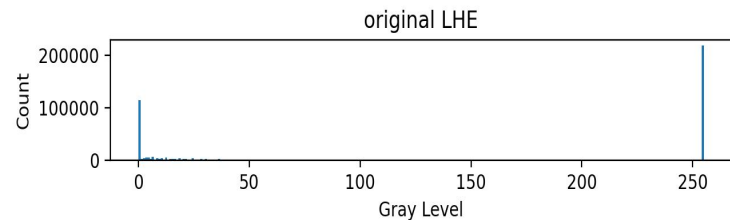


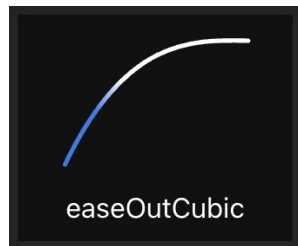
Image Enhancement



Image Enhancement

- 作法

1. 使用 GHE 將暗部的細節凸顯出來
2. 使用 easeOutCubic 將整體明亮度作提升，並且將暗部的對比度提高
3. 使用 easeOutSine 將中間名度的對比度提高，讓畫面看起來更自然一點



Noise Removal - uniform noise

$3 \times 3 - 2^2$



```
[[1 2 1]
 [2 4 2]
 [1 2 1]]
```

PSNR: 27.34

$3 \times 3 - 2^3$



```
[[2 4 2]
 [4 8 4]
 [2 4 2]]
```

PSNR: 27.34

$5 \times 5 - 2^4$



```
[[ 1  2  4  2  1]
 [ 2  4  8  4  2]
 [ 4  8 16  8  4]
 [ 2  4  8  4  2]
 [ 1  2  4  2  1]]
```

PSNR: 27.54

$7 \times 7 - 2^5$



```
[[ 1  1  2  4  2  1  1]
 [ 1  2  4  8  4  2  1]
 [ 2  4  8 16  8  4  2]
 [ 4  8 16 32 16  8  4]
 [ 2  4  8 16  8  4  2]
 [ 1  2  4  8  4  2  1]
 [ 1  1  2  4  2  1  1]]
```

PSNR: 27.03

$5 \times 5 - \text{cross}$



PSNR: 25.18

Noise Removal - uniform noise

$3 \times 3 - 3^3$



```
[[ 3  9  3]
 [ 9 27  9]
 [ 3  9  3]]
```

PSNR: 26.56

$5 \times 5 - 3^3$



```
[[ 1  1  3  1  1]
 [ 1  3  9  3  1]
 [ 3  9 27  9  3]
 [ 1  3  9  3  1]
 [ 1  1  3  1  1]]
```

PSNR: 27.36

$5 \times 5 - 4^3$



```
[[ 1  1  4  1  1]
 [ 1  4 16  4  1]
 [ 4 16 64 16  4]
 [ 1  4 16  4  1]
 [ 1  1  4  1  1]]
```

PSNR: 26.60

$5 \times 5 - 5^3$



```
[[ 1  1  5  1  1]
 [ 1  5 25  5  1]
 [ 5 25 125 25  5]
 [ 1  5 25  5  1]
 [ 1  1  5  1  1]]
```

PSNR: 25.78

Noise Removal - uniform noise

- 在 uniform noise 的問題中，可以看出 3x3 kernel size filter 中心值設定為 4^4 時所獲得的 PSNR 25.74 最高
- 討論
 - 太大片的 filter ($>5 \times 5$), 會使畫面過於模糊, PSNR 無法獲得顯著的提升
 - center value 如果設定太高, 反而會強化雜訊, 獲得的 PSNR 最差 ($5 \times 5 - 5^3$)
 - uniform noise 的分佈平均且密集, 使用 median filter 的方法 PSNR 無法獲得提升 (PSNR: 25.18)

$5 \times 5 - 2^4$



```
[[ 1  2  4  2  1]
 [ 2  4  8  4  2]
 [ 4  8 16  8  4]
 [ 2  4  8  4  2]
 [ 1  2  4  2  1]]
```

PSNR: 27.54

Noise Removal - Salt-and-pepper noise



3x3



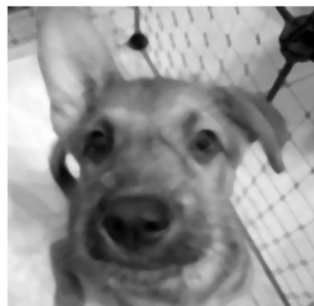
PSNR: 30.09

5x5



PSNR: 27.11

7x7



PSNR: 24.91

5x5-2⁴



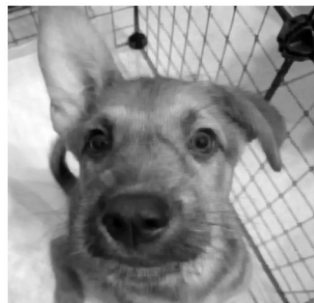
PSNR: 27.82

5x5-cross



PSNR: 31.59

7x7-cross



PSNR: 28.49

9x9-cross



PSNR: 26.56

Noise Removal - Salt-and-pepper noise

- 在 Salt-and-pepper noise 的問題, 5x5-cross 的 median filter 獲得的 PSNR 最佳
- 討論
 - 任意 size kernel filter 中, cross filter 都獲得比 square filter 更好得成績, 判斷可能是因為圖片中有較多毛髮、柵欄等線性物體, 因此 cross filter 更適合處理這類影像
 - median filter kernel 到達 5x5 以上時畫面會變得過於模糊, PSNR 27.11 效果較差
 - 若拿先前 low-pass-filter 的最佳解 $5 \times 5 - 2^4$ 來跑, PSNR 27.82, 效果甚至比 5x5 以上的 median filter 來得更好



PSNR: 31.59



Thank you