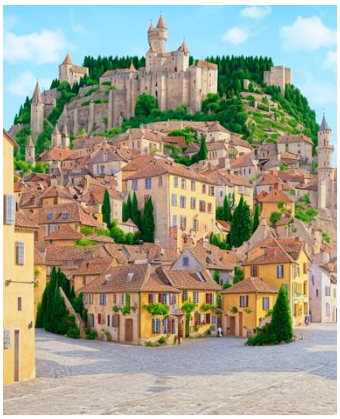
Problem 0



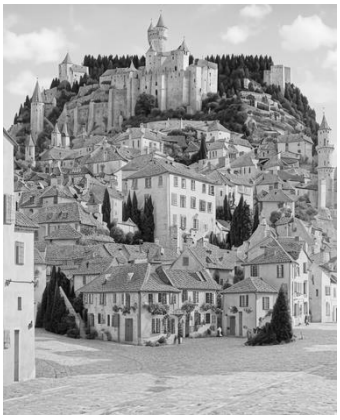
sample1.png

(a) Vertical Flipping

result1.png

把 i-th row 與 (# of rows – 1 – i)-th row 交換即可得到上下顛倒的圖片

(b) Gray Scale



result2.png

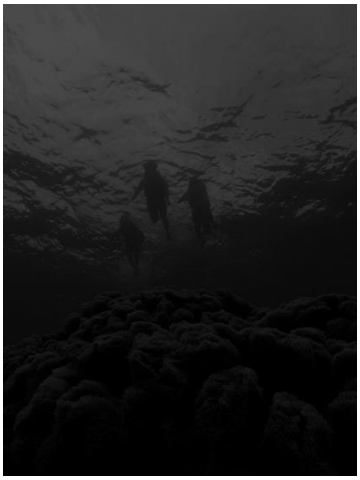
公式: 0.114*Red + 0.587*Green + 0.299*Blue

此公式由 ITU-R 提出的 BT.601 中計算所得到。綠色的權重比其他顏色重因為人眼對綠色較為敏感。

Problem 1

sample2.png

(a) Decrease the brightness s of sample2.png by dividing the intensity values by 3



result3.png

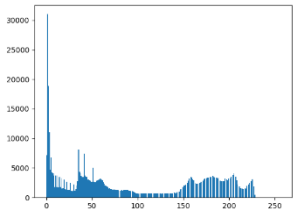
將每個像素的灰階值整數除以三即可得到 result3

(b) Increase the brightness of result3.png by multiplying the intensity values by 3

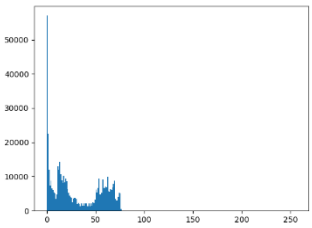
result4.png

由 result3 的灰階值乘以三，result4 的結果灰色漸層比原本 sample2 淡

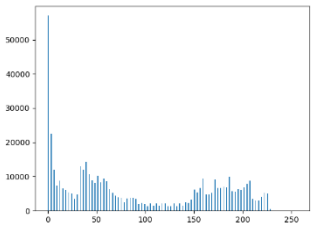
(c) Plot the histograms of sample2.png, result3.png and result4.png



sample2



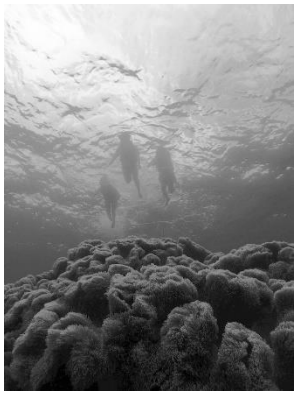
result3

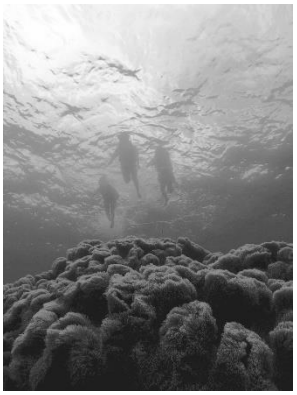


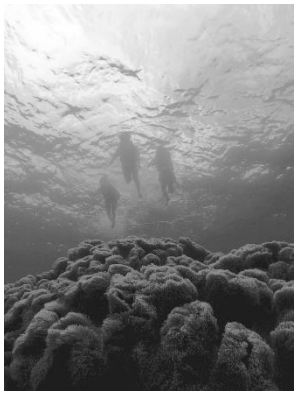
result4

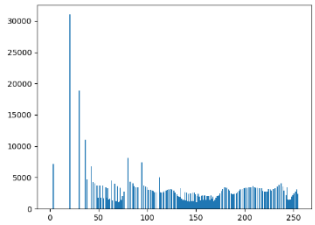
因為 result3 把 sample2 的灰階值除以三所以不會有大於 85 的值，而 result4 又把 result3 乘以三，但因為壓縮的的關係所以像素最多只有 85 個值，像是把原圖的灰階值三個為一組分組。

(d) Perform global histogram equalization on sample2.png, result3.png and result4.png

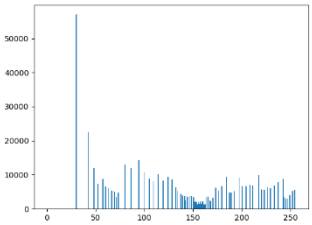




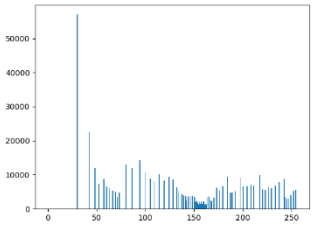




result5.png



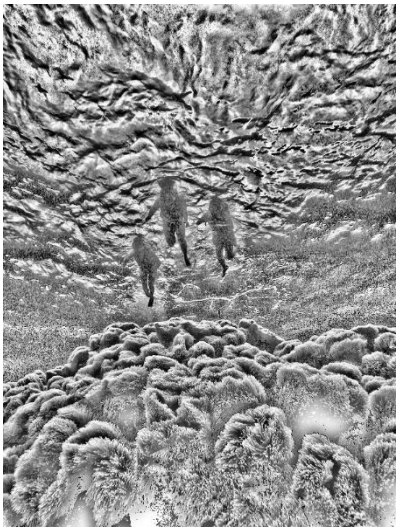
result6.png

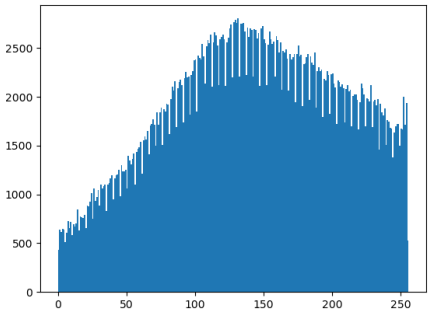


result7.png

Equalize 之後三張圖的亮度差不多，由於 result3 壓縮過原本的灰階，result6 與 result7 不存在某些 result5 的灰階值。雖然 result6 與 result7 少了某些值，但結構與 result5 差不多，可以看出某些區域較密，某些較疏。

(e) Perform local histogram equalization on sample2.png

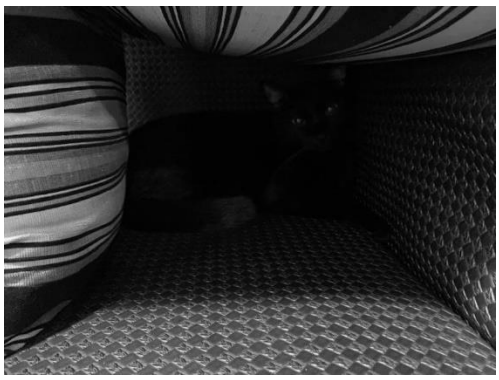




result8.png

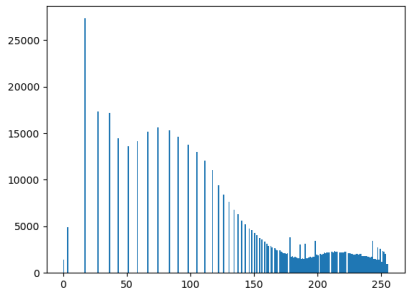
Result8 為使用 size=31*31 之 kernel 的 local equalization，由於用 local equalization，result8 的對比度比 sample2 大得多，histogram 也並非像 global equalization 一樣平坦。使用 size=31*31 的原因是在大 kernel 會使用較久的時間但成效沒有比較好。

(f) Design a transfer function to enhance sample3.png



sample3.png

result9.png

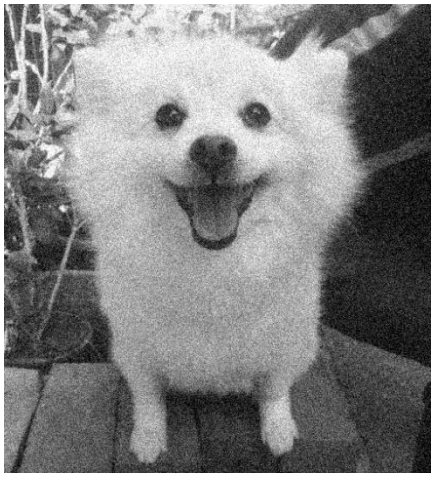


Design: 把圖像亮度*2，然後做 global equalization。

Motivation: 先提高亮度只能看到部分的輪廓，在利用 equalization 強化灰階的變化。

Problem 2

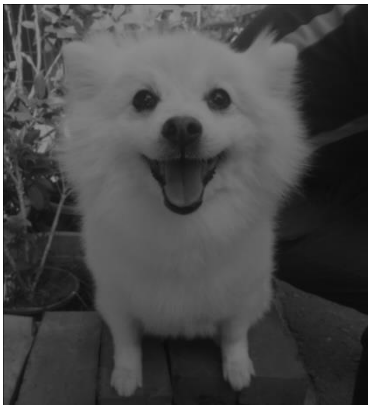
sample4.png



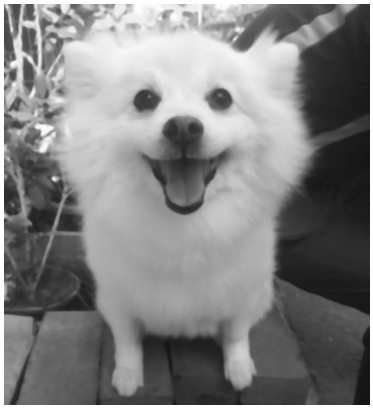
sample5.png

sample6.png

(a) Design different filters to remove the noise in sample5.png and sample6.png



result8.png



result9.png

Sample5 中的雜訊為 Gaussian Noise，適合用 Low-pass filter 平滑範圍內的雜訊，使用的 kernel 為[[1,2,1],[2,4,2],[1,2,1]]，因為取平均的關係，圖片會比較暗。
Sample6 中的雜訊為 Salt and Pepper Noise，適合用 Median filter 去除極值的雜訊，使用的 kernel size 為 5*5。

(b) Compute PSNR values of result10.png and result11.png

	Result10	Result11
PSNR	27.33920	37.62469

由於 Result10 用 Gaussian filter，每個 pixel 與原圖都均有差異，其 PSNR 會較低。而 Result11 用 Median filter，每個與原圖相同的 pixel 均有保留，修改的只有極值的雜訊，PSNR 較高。