影像處理作業報告

HW1

Spatial Image Enhancement

授課教授：柳金章

學　　生：楊憲閔

學 　號：613410047

Due date：2024/11/07

Date hand in：2024/11/06

目錄

[Technical description 3](#_Toc181092894)

[Experimental results 10](#_Toc181092895)

[Discussions 19](#_Toc181092896)

[References and Appendix 20](#_Toc181092897)

# Technical description

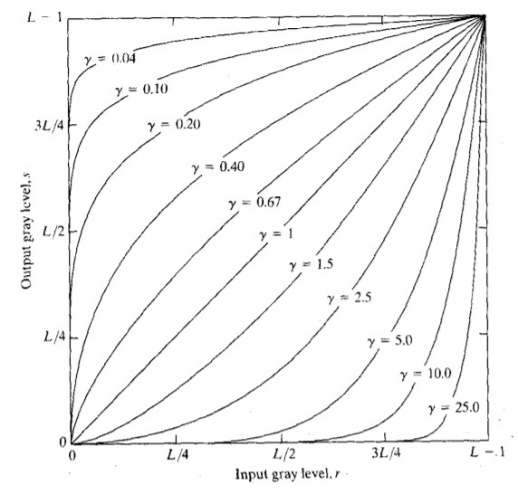
　　影像會受到環境、拍攝工具、拍攝參數等等所影響，導致影像有些時候會偏暗或是偏亮，進而影響影像中的物體輪廓不清或是一些細節顯示較為不佳，因此我們需要對影像進行影像處理(即強化)，來達到將影像變成較適合肉眼觀察的影像，本Homework則是要利用三種方法來達到上述的效果，下面將會介紹三種方法的理論與對應的結果。

1. Power-Law Transformations

又稱gamma transformation。是利用非線性運算的公式對影像進行處理，來自人類的視覺系統具有類似的特色，因此我們可以利用gamma相關公式進行視覺補償，公式如下:

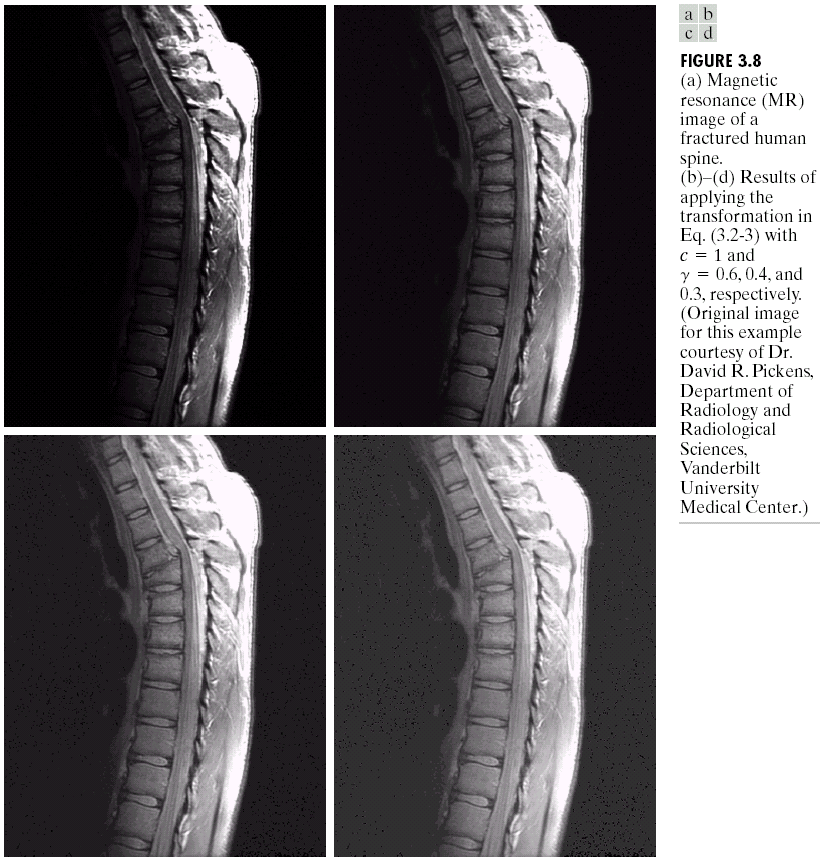
(1)

其中r和s為輸入和輸出影像之pixel亮度，c和γ皆為大於0的常數(c通常設為1)。其轉換曲線如下圖:

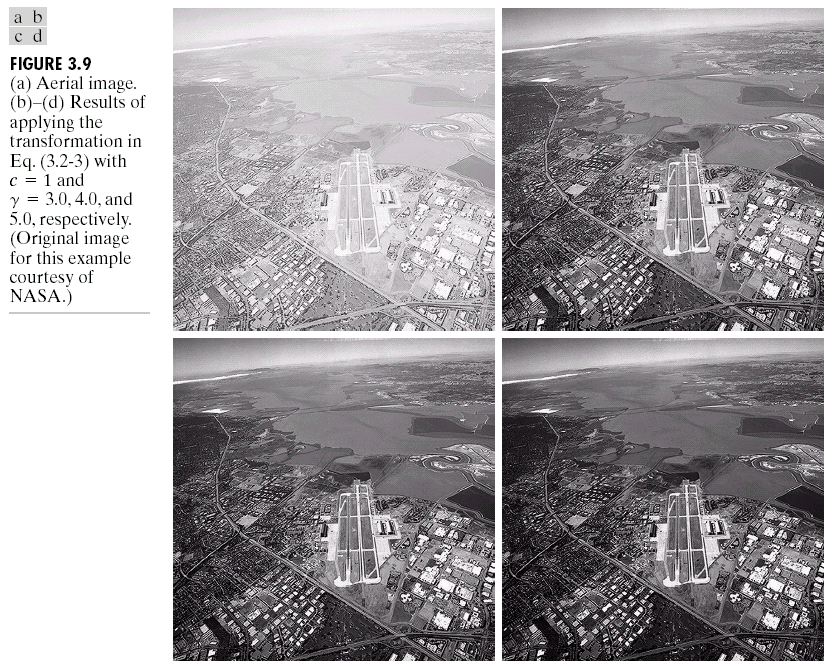


圖(1) 不同γ的轉換曲線。

不難發現圖(1)中，若γ<1時，大部分原本亮度低的會被mapping到亮度高，而原本亮度高的還是亮度高，因此可以理解成將整體的亮度數值提高，也就是讓原本較暗的影像轉成較亮的影像；若γ>1時，原本亮度低的還是亮度低，而大部分原本亮度高的會被mapping到亮度低，因此可以理解成將整體的亮度數值下降，也就是讓原本較亮的影像轉成較暗的影像。



圖(2) 輸入偏暗的影像，可以發現γ越小，轉換而成的影像會越亮，位於暗處的細節也會跟著明顯起來。



圖(3) 輸入偏亮的影像，可以發現γ越大，轉換而成的影像會越暗，位於亮處的細節也會跟著明顯起來。

1. Histogram equalization

色調過於集中於某一區段，對比度不明顯，使用Histogram Equation，將原影像的灰階值平均分布，使其對比度提高，進而成為適合肉眼觀看的影像。

其中我們需要一個直方圖，是一個離散函數，其灰階值範圍為[0, L-1]，L是影像中的灰階總數，通常L為256。

接著我們計算每個灰階值出現的次數，再除以影像的總pixel數，以計算該灰階值出現的機率，公式如下:

(2)

rk是第k個灰階值，pr(rk)是原影像rk這個灰階值的出現機率，nk是有多少個pixel的灰階值為rk，n是影像總pixel數。

再將PDF累加求出累積分佈函數(CDF)，產生一個mapping function，利用CDF的數值讓原本影像中的各個灰階值有一個對應的新灰階值，公式如下:

(3)

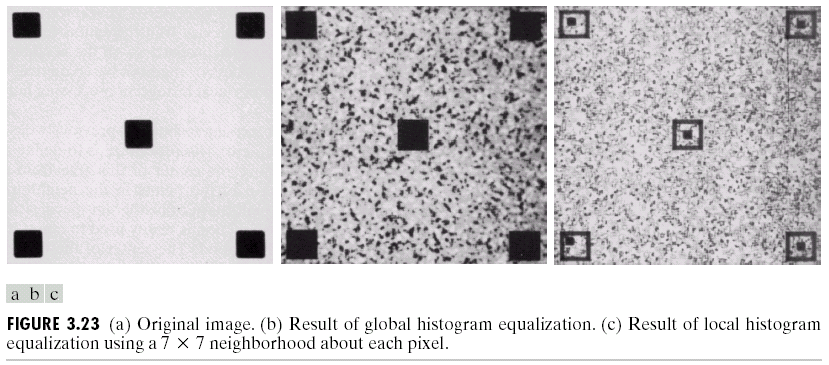
其中T(rk)是上面所說的mapping function，sk是第k個灰階值轉換後所對應之新的灰階值。

最後將原本影像的每個pixel之灰階亮度rk轉換成相對應的灰階亮度sk，即可完成轉換，使得原本數值應過於集中的情況得到改善，達到影像強化的效果。



圖(4) 原圖灰階值大多在[120,200]，經過方法計算進行影像強化後，灰階值直方圖就會顯得比較平均，圖片也較適合肉眼觀察。

此外，這個方法不僅可以使用於整張圖(Global Enhancement)，也可以先對影像進行分割為多個區塊，再對每個區塊做histogram equalization(Local Enhancement)，因為兩者產生的結果大不同，因此我們可以視需求而決定要使用global histogram equalization或local histogram equalization。



圖(5) 原圖中心與四周都有黑色方塊，我們想對他們做消除，可以發現global處理較為不好，那五個方塊還在，但local可以消除方塊的些許部分，看起來效果較佳。

1. Image sharpening using the Laplacian operator

Image sharpening即為對影像中的細節進行強化，其中就是利用數學的一次與二次微分做基礎的，其公式如下。

(4)

(5)

其中一次微分會產生「較厚」的邊緣，對於灰階值的梯度變化反應較明顯；二次微分中則是產生更細緻的邊緣，並對細節(細線條或是孤立的點)反應較為明顯。

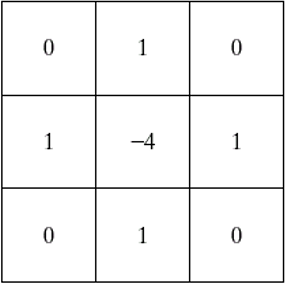
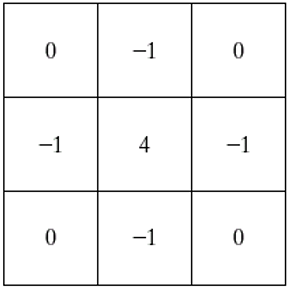
基於二次微分對於細節有更好的反應，有人提出了利用二次微分達到銳利化的方法-「Laplacian operator」，公式如下:

(6)

(7)

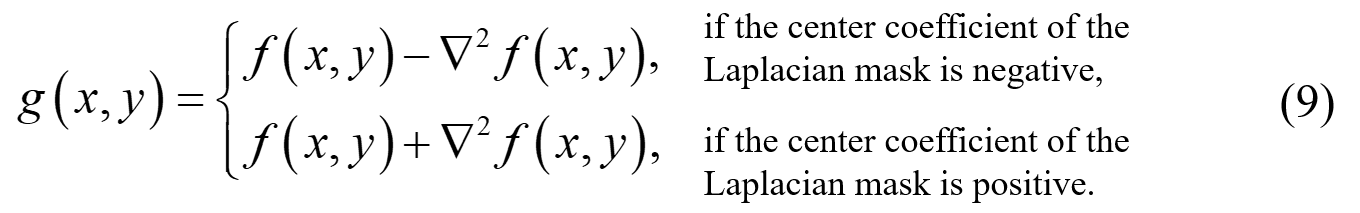
(8)

實作digital Laplacian有很多種，下面為其中一種filter mask。

圖(6) 其中一種filter mask實作Laplacian

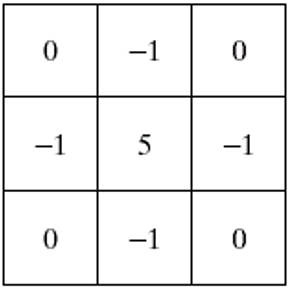
用Laplacian filter mask實現影像強化，其公式如下。



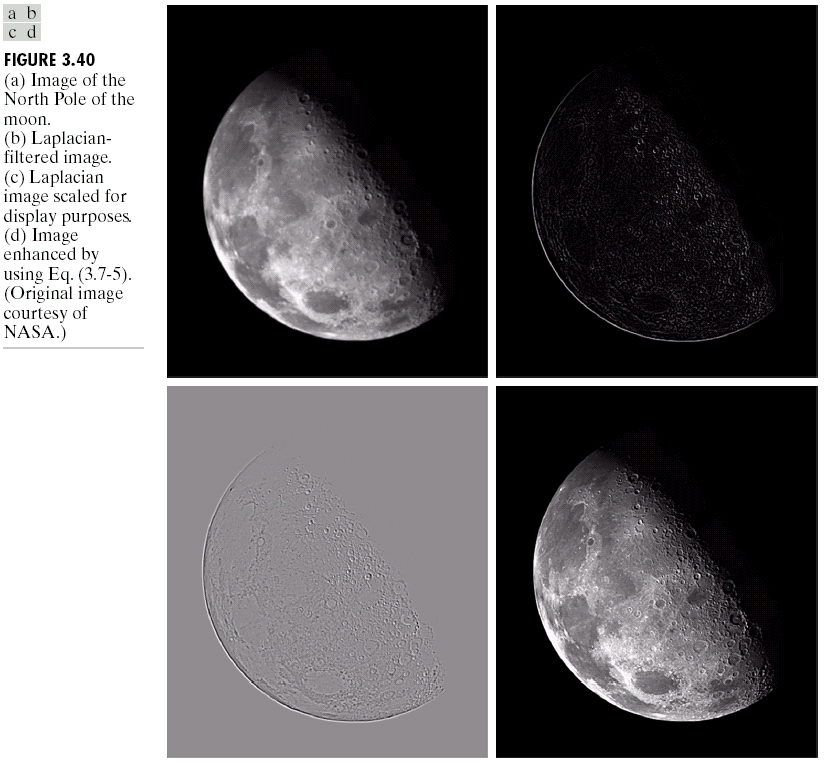
以圖(6)的filter mask為例，可以得到:

(10)

其中g(x,y)是經由Laplacian operator強化過後的影像之(x,y)的灰階值，因此我們可以得到一個Composite Laplacian filter mask，讓我們可以直接用此filter mask與原影像做相乘即可得到我們要的影像。



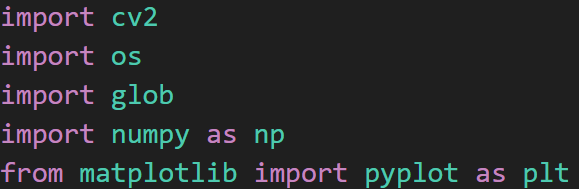
圖(7) Composite Laplacian filter mask



圖(8) Laplacian operator進行影像強化的過程。

# Experimental results

1. 程式執行流程:
2. 確保已安裝相關module，本次作業使用module如下所示:

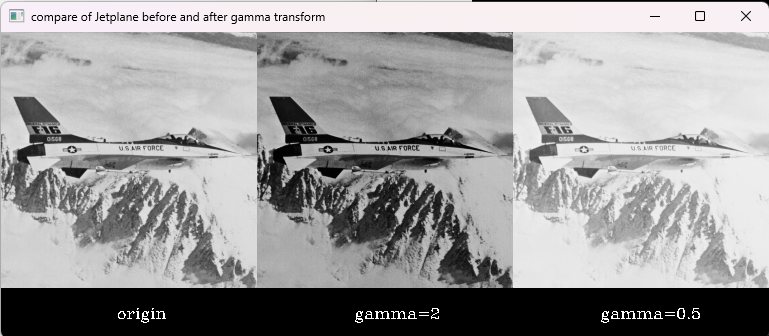


圖(9) 會使用到的module

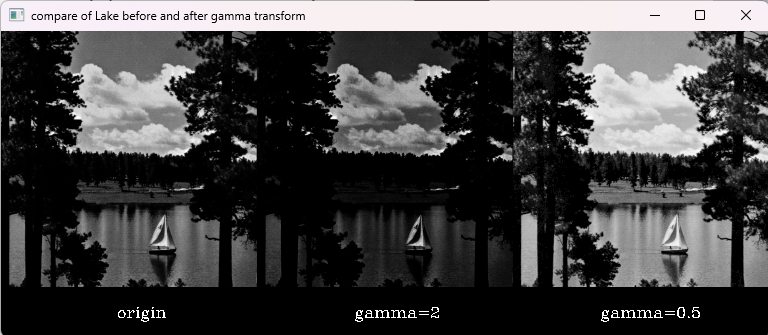
1. 進到作業的目錄底下，會看到一個名為HW1\_test\_image的資料夾，一個main.py，還有這份pdf，點右鍵按在終端中開啟，輸入python main.py，程式即開始執行。
2. 程式會讀取HW1\_test\_image資料夾底下的圖片，並輸出對每個圖片進行transform的結果，即順序會是讀一張圖片，輸出對該圖片進行gamma transform的結果，關掉視窗後會輸出對該圖片進行histogram equalization的結果，並在後面輸出對應的histogram，再關掉視窗後會輸出利用Laplacian operator對該圖片進行Image sharpening的結果，最後關掉視窗則程式繼續讀取下一張圖片，並做一樣的順序，直到所有圖片都被讀取完，即結束程式。
3. 程式執行結果:
4. Gamma transform:



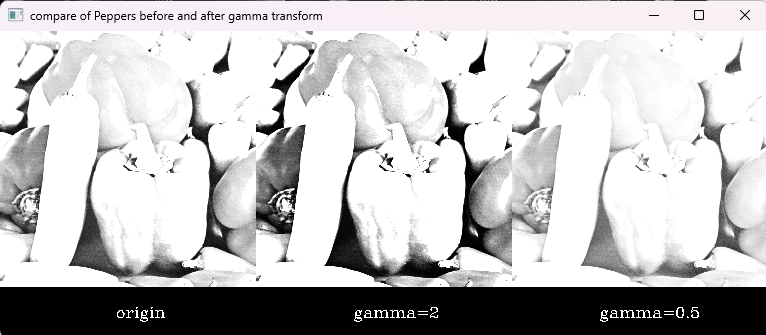
圖(10) Cameraman.bmp原圖與轉換後之影像



圖(11) Jetplane.bmp原圖與轉換後之影像



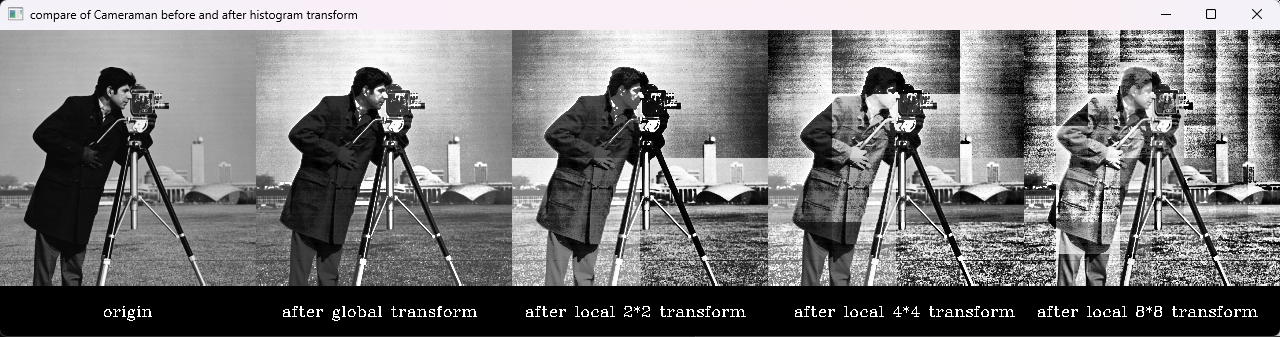
圖(12) Lake.bmp原圖與轉換後之影像



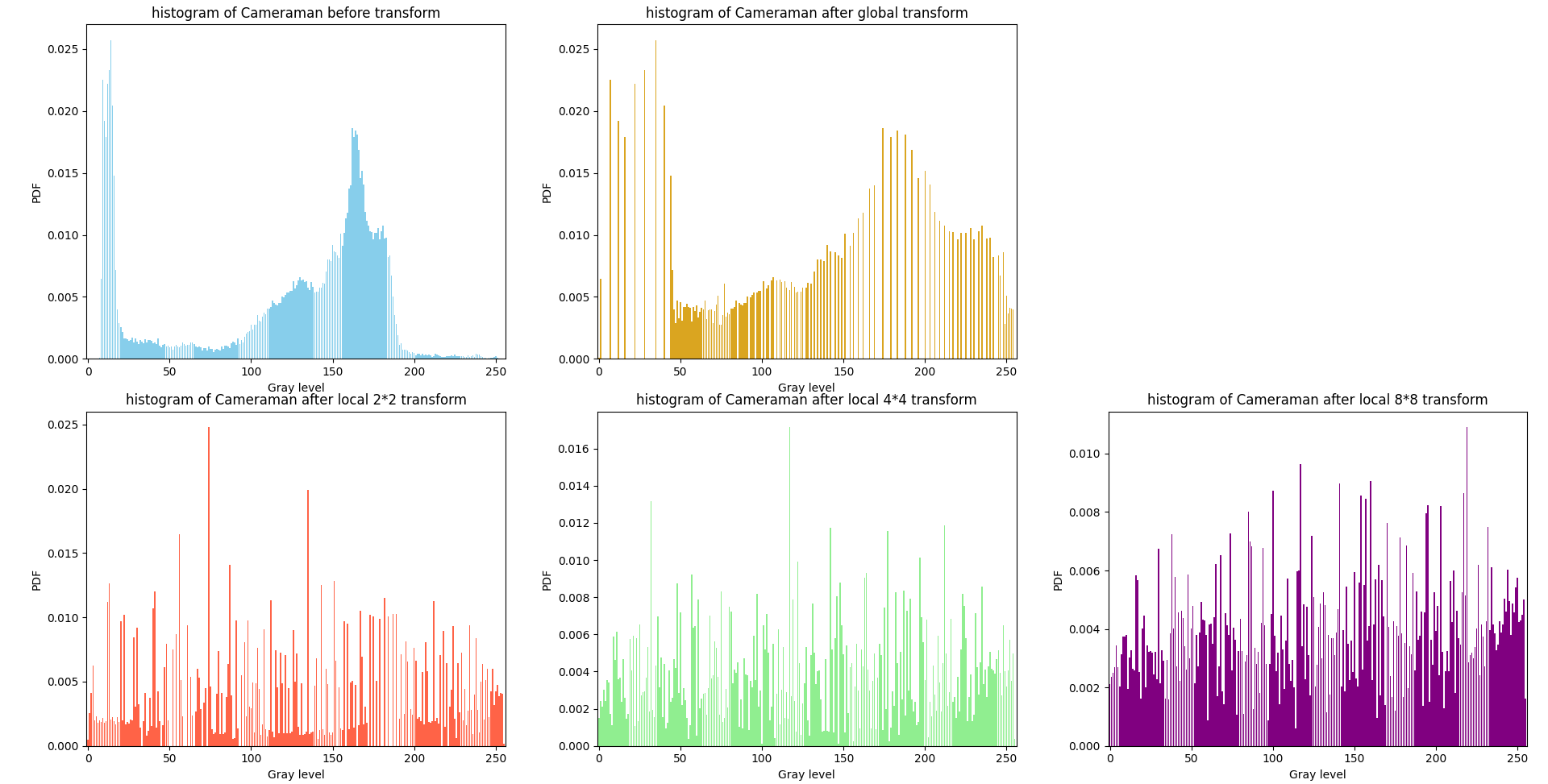
圖(13) Peppers.bmp原圖與轉換後之影像

從這些output中可以驗證上面所說的，γ<1會使整體的亮高，γ>1會使整體的亮度降低。

1. Histogram equalization:

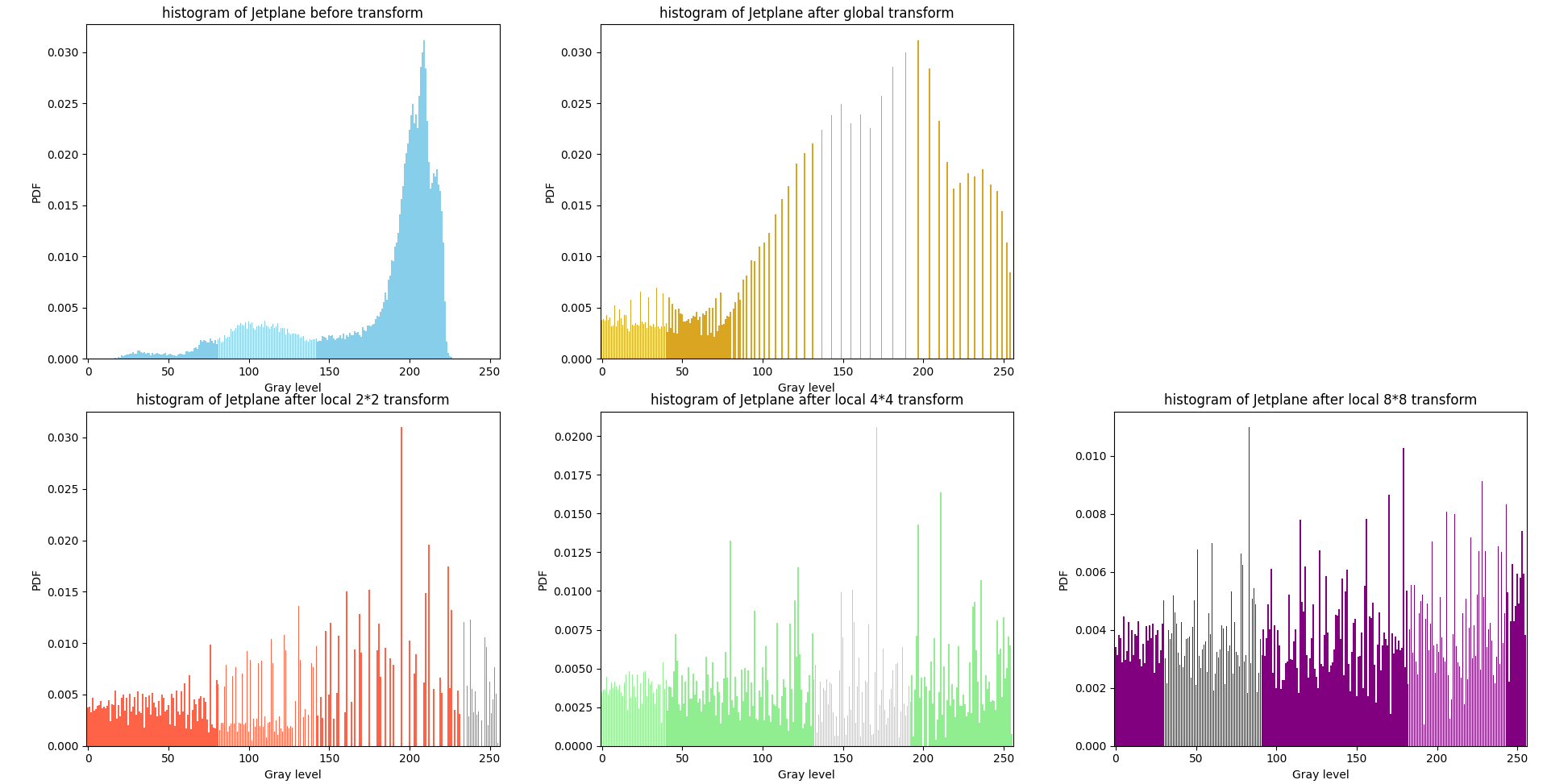


圖(14) Cameraman.bmp原圖與global histogram equalization 與local histogram equalization後之影像



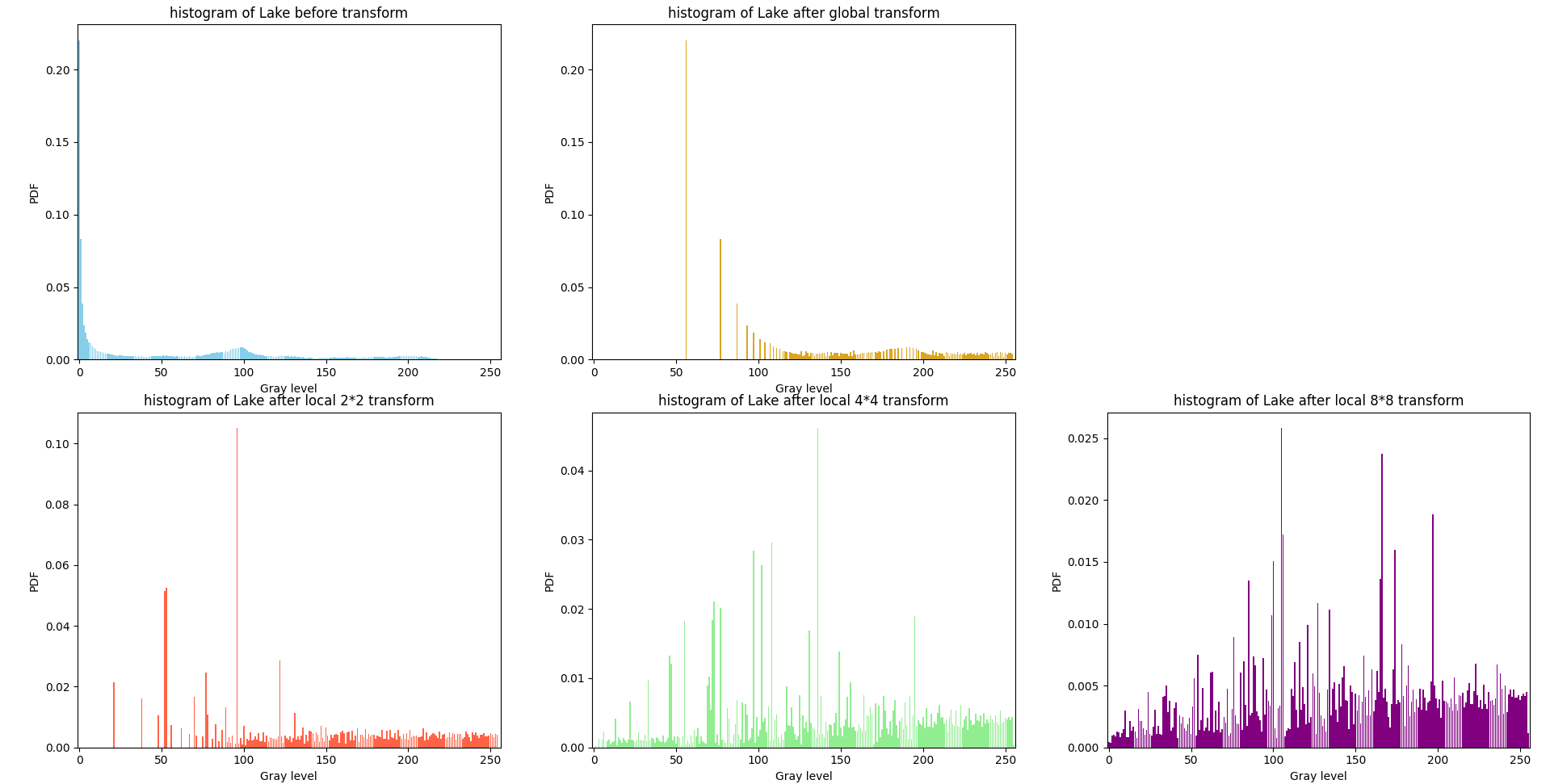
圖(15) 原圖與global histogram equalization與local histogram equalization後之histogram

 圖(16) Jetplane.bmp原圖與global histogram equalization 與local histogram equalization後之影像



圖(17) 原圖與global histogram equalization與local histogram equalization後之histogram

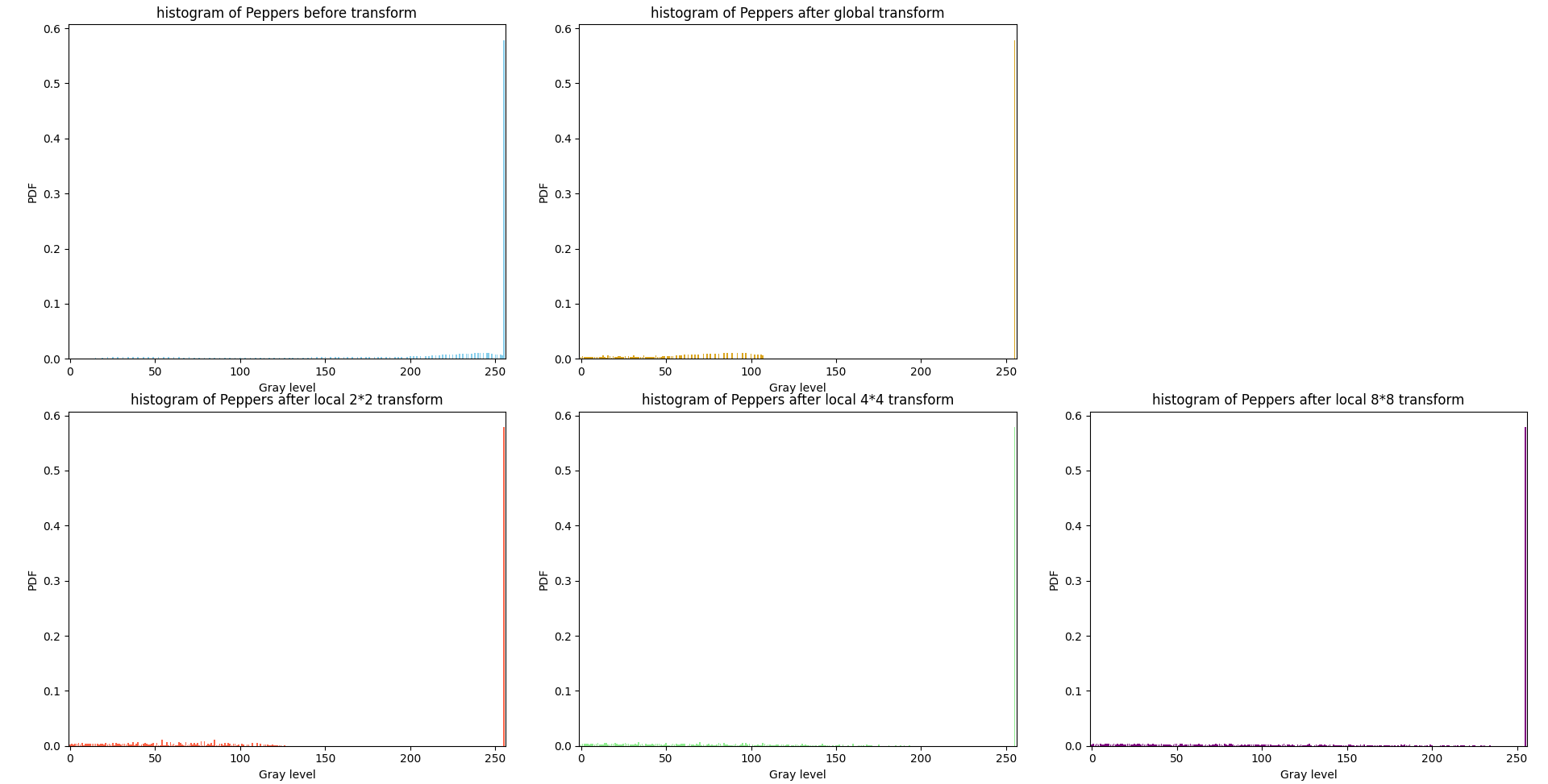
圖(18) Lake.bmp原圖與global histogram equalization 與local histogram equalization後之影像



圖(19) 原圖與global histogram equalization與local histogram equalization後之histogram



圖(20) Peppers.bmp原圖與global histogram equalization 與local histogram equalization後之影像



圖(21) 原圖與global histogram equalization與local histogram equalization後之histogram

可以發現在經過處理後，集中在某處的話會被分散掉，使之較為平均，如圖(15)、圖(17)、圖(19)與圖(21)中紅框處可以清楚看到這個現象。

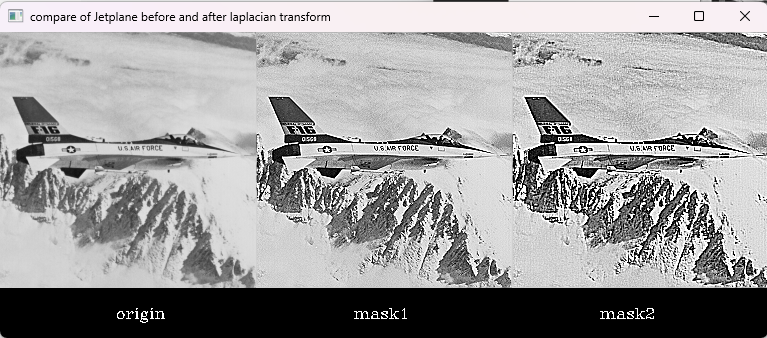
其中Global整張圖處理與Local分段處理的概念不同，但原理是一樣的，不過分成部分來剖析的話，我個人覺得Local在單一小格內來看灰階的漸層感會比Global 還要優秀一些，但很明顯地，從肉眼的理解來看，圖的邊界看起來很不連續且非常礙眼，反而更難辨識了。

比較特別的是Peppers.bmp這張，這張很明顯大部分 都是亮的，在經過處理後看到圖(21)只有最高亮度以外的地方mapping到亮度低的位置，最高亮度依然占比很大，強化出來的影像和原始影像相比，亮度低的地方有所增加，但整體亮度依然很亮。這顯示了當原始影像當中有若有某個值相較於其他值高出很多很多時，經過histogram equalization後，所得到的強化效果是有限的。

1. Image sharpening using the Laplacian operator:

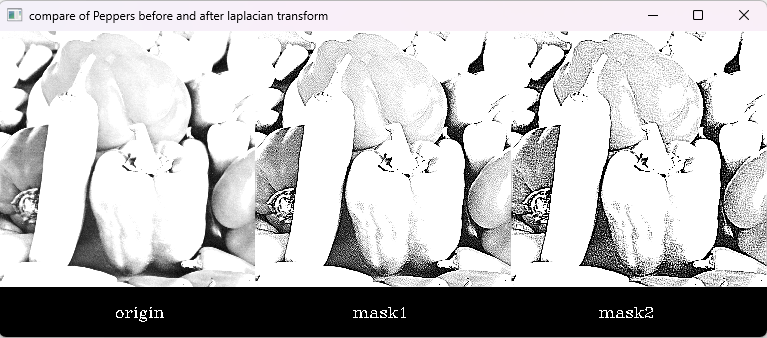


圖(22) Cameraman.bmp原圖與Image sharpening using the Laplacian operator後之影像



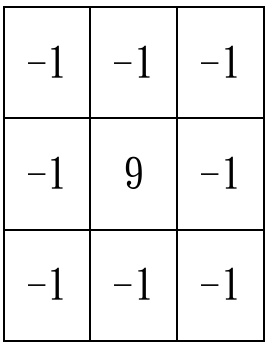
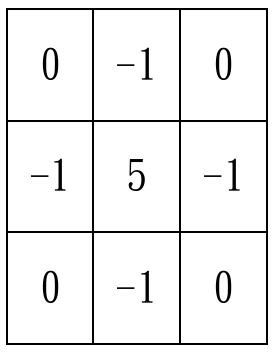
圖(23) Jetplane.bmp原圖與Image sharpening using the Laplacian operator後之影像

圖(24) Lake.bmp原圖與Image sharpening using the Laplacian operator後之影像



圖(25) Peppers.bmp原圖與Image sharpening using the Laplacian operator後之影像

其中mask1與mask2為:



圖(26) 左邊為mask1，右邊為mask2

這裡可以看出使用的是合成型Laplacian operator，直接對影像做銳利化，並且因為未padding，故不會對影像最外圍的邊界做Laplacian operator。從上面的output可以看出，如同上面的論述，在經過Laplacian operator進行銳利化後，原始影像當中本來沒那麼明顯的細節將會顯示出來。

# Discussions

大致總結一下這三種強化方法:

* 1. gamma transform主要是把影像做整體亮度調整，簡單但無法做更進一步的強化。
  2. Histogram equalization是對整體進行平均化，避免較集中於某些值，如此可以讓大致的輪廓顯現，但遇到像Peppers.bmp這種某個值的出現頻率過於壓制其他值時，histogram的效果仍會不佳。
  3. Sharpening using the Laplacian operator是針對影像當中細節去做銳利化使之顯現出來，但跟前兩種方法相比，它不會顯著地影響影像整體的亮度和輪廓。

每個方法都有它的特色，有優點也有缺點，因此不是所有的影像都適合某某影像增強的方法，找出適合的方法並應用之，才是最好的!

# References and Appendix

histogram equalization

<https://zh.wikipedia.org/zh-tw/%E7%9B%B4%E6%96%B9%E5%9B%BE%E5%9D%87%E8%A1%A1%E5%8C%96>

histogram equalization

<https://jason-chen-1992.weebly.com/home/-histogram-equalization>

gamma correction

<https://jason-chen-1992.weebly.com/home/-gamma-correction>

Laplacian Operator

<https://medium.com/%E9%9B%BB%E8%85%A6%E8%A6%96%E8%A6%BA/%E9%82%8A%E7%B7%A3%E5%81%B5%E6%B8%AC-%E6%8B%89%E6%99%AE%E6%8B%89%E6%96%AF%E7%AE%97%E5%AD%90-laplacian-operator-ea877f1945a0>

Laplacian Operator

<https://docs.opencv.org/3.4/d5/db5/tutorial_laplace_operator.html>

Laplacian Operator

<https://homepages.inf.ed.ac.uk/rbf/HIPR2/log.htm>

subplot

<https://steam.oxxostudio.tw/category/python/example/matplotlib-subplot.html>

puttext()

<https://steam.oxxostudio.tw/category/python/ai/opencv-text.html>

puttext()

<https://blog.csdn.net/weixin_41010198/article/details/89155899>