I. Low-luminosity Enhancement

此處採用的是 Point Operator 的做法，也就是每個點之間是獨立運算的。
首先，我們先將 RGB 的值進行YC_rC_b的轉換，公式如下

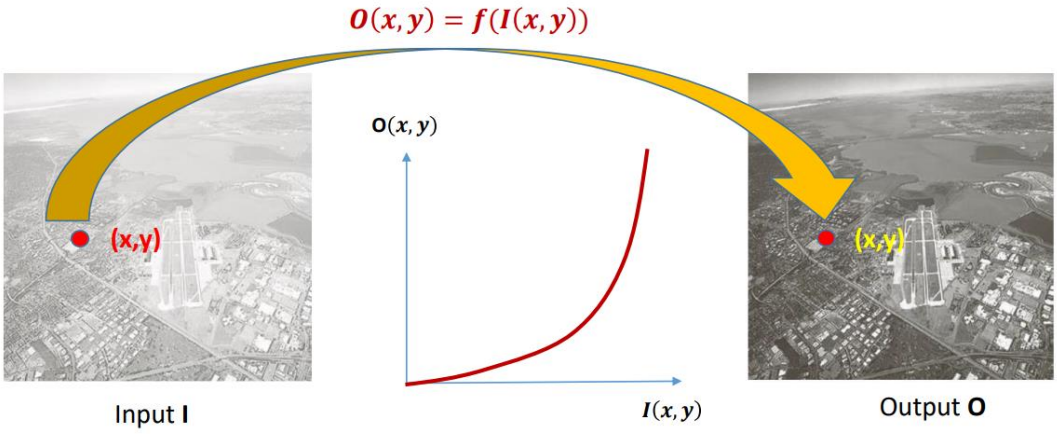
1. Y 亮度分量:
   Y 代表圖像的量度，其計算公式為
   $$Y = 0.299R + 0.587G + 0.114B$$
   這些權重因子（0.299、0.587 和 0.114）是經驗值，用於模擬人眼對不同顏色通道的感知敏感度
2. C_b(藍色差值)分量:
   C_b代表藍色通道與亮度之間的差異，其計算公式如下
   $$C_b = 0.564(B - Y)$$
   這個公式計算藍色通道 B 與亮度 Y 之間的差異，並將其縮放以適應C_b的範圍
3. C_r(紅色差值)分量:
   C_r代表紅色通道與亮度之間的差異，其計算公式如下
   $$C_r = 0.713(R - Y)$$
   這個公式計算藍色通道 B 與亮度 Y 之間的差異，並將其縮放以適應C_r的範圍

得到 Y 值(量度分量)後，我們就能對其進行 Log Transformation，變換公式如下，

$$s = c \times \log(1 + r)$$

s 是輸出的像素值，c 是供我們調整的參數，而 r 則是輸入的像素值。
此處的 r 即代表量度分量 Y。

透過 Log Transformation，我們可以將暗的地方拉開，如圖



Log Transformation 之後，再將YC_rC_b轉換回 RGB 值並輸出到 bmp 檔中
以下為經不同 C 轉換後的兩張圖片。


II. Sharpness Enhancement

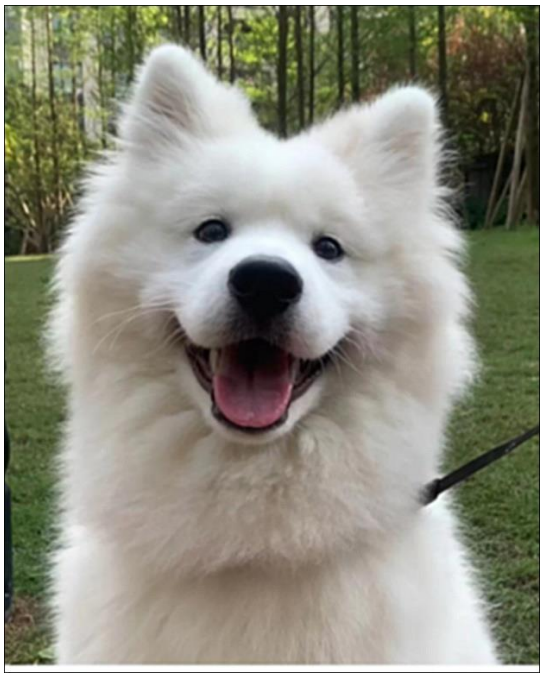
此處採用的是 Neighborhood Operations 中的 Sharpness Enhancement 的技巧，每個像素透過自身和鄰近的 RGB 值去計算出一組新的 RGB 值，也就是 convolution 的方式。

為了運算方便，我採用一個3×3大小的卷積核(filter)，名為$g(x,y)$。假設原本的 RGB 值叫做$f(x,y)$，最後的輸出則為$h(x,y) = f(x,y) * g(x,y)$
換句話說，公式如下

$$\begin{aligned} h(x,y) = {} & g(-1,-1) \times f(x-1,y-1) + g(-1,0) \times f(x-1,y) \\ & + g(-1,1) \times f(x-1,y+1) + g(0,-1) \times f(x,y-1) \\ & + g(0,0) \times f(x,y) + g(0,1) \times f(x,y+1) \\ & + g(1,-1) \times f(x+1,y-1) + g(1,0) \times f(x+1,y) \\ & + g(1,1) \times f(x+1,y+1) \end{aligned}$$

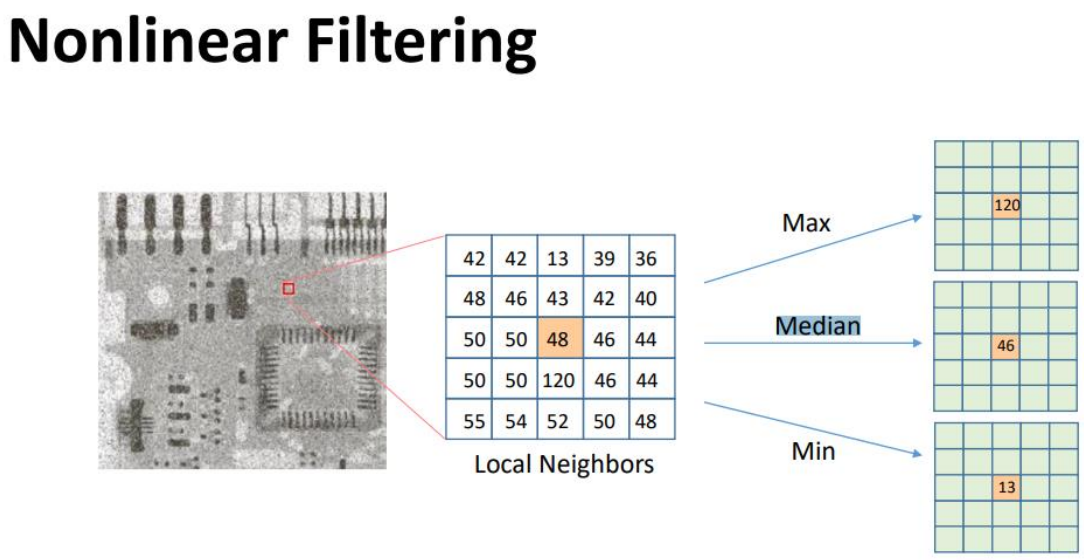
而在此份 code 中，我使用$h(x,y) = \begin{matrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{matrix}$ 和 $h(x,y) = \begin{matrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{matrix}$

以下是不同輸出的比較




III. Denoise

在此份 code 中，我使用了 median filter 進行運算。Median filter 的運算原理很簡單，就是利用 noise 的平均為 0 的特性，去取特定範圍內的中間值。如圖。



而在此次作業中，我分別將 median filter 的大小設為5×5以及3×3去做 median filter。通常，較大的窗口（例如 5x5）能夠更好地去除較大的 noise，但可能會導致一些細節的損失，而較小的窗口（例如 3x3）能夠保留更多的細節，但可

能不夠有效地去除較大的 noise。

我們可以比較下面兩張圖的結果，上圖為5×5而下圖為3×3。