

# Digital Image Processing HW2

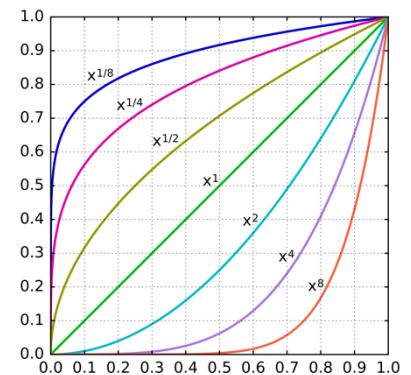
鄧奕辰 311605014

## ● Low-luminosity Enhancement

此任務須將低亮度的圖片轉為高亮度的圖片，且為了讓低亮度的區域提升比例更高，而高亮度提升較低，做了以下操作：

1. 通過將每個像素的紅、綠、藍通道的值除以 255，將通道值正規化為 0 到 1 之間的範圍。
2. 將每個通道的正規化值按照 gamma 值進行次方運算，以下圖方式成長。
3. 將計算後的值反向歸一化（乘以 255），得到新的紅、綠、藍通道值。
4. 由於色彩值的範圍應在 0 到 255 之間，使用 min 函式將新的色彩值限制在這個範圍內，並將其轉換為整數後存儲在目標圖像中。

```
void improveLuminosity(const RGBA** src, RGBA** dest, int srcWidth, int srcHeight, float gamma=0.5){  
    for (int i = 0; i < srcHeight; ++i) {  
        for (int j = 0; j < srcWidth; ++j) {  
  
            double normalizedRed = src[i][j].r / 255.0;  
            double normalizedGreen = src[i][j].g / 255.0;  
            double normalizedBlue = src[i][j].b / 255.0;  
  
            double newRed = 255.0 * pow(normalizedRed, gamma);  
            double newGreen = 255.0 * pow(normalizedGreen, gamma);  
            double newBlue = 255.0 * pow(normalizedBlue, gamma);  
  
            // Ensure that the values do not exceed the maximum (255)  
            dest[i][j].r = min(255, static_cast<int>(newRed));  
            dest[i][j].g = min(255, static_cast<int>(newGreen));  
            dest[i][j].b = min(255, static_cast<int>(newBlue));  
            dest[i][j].a = src[i][j].a;  
        }  
    }  
}
```

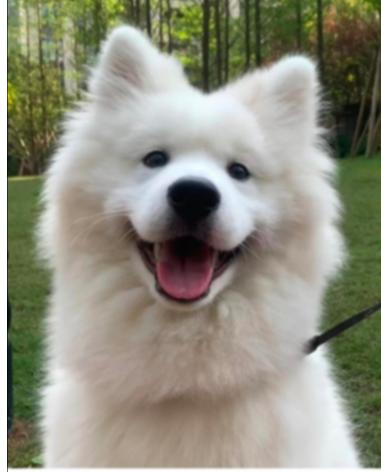


原圖	Gamma = 0.8	Gamma = 0.5
原圖	依照 gamma = 0.8 提升亮度，亮度只有些許提升。	依照 gamma = 0.5 提升亮度，暗部提升比例更多，而亮部並沒有發生過曝的情形。

## ● Sharpness Enhancement

此任務須將模糊的圖片銳化，我使用不同的 filter 來做卷積，每種 filter 所展現的功用略有不同，通常會搭配使用不同的 filter 來做使用，但以下就以兩種 filter 來做討論與分析。

```
void sharpnessEnhance(const RGBA** src, RGBA** dest, int srcWidth, int srcHeight, float factor = 1) {
    // 3x3 拉普拉斯濾波器
    // int laplacianFilter[3][3] = {
    //     {-1, -1, -1},
    //     {-1, 9, -1},
    //     {-1, -1, -1}
    // };
    // You, 1 秒前。Uncommitted changes
    int laplacianFilter[3][3] = {
        {1, -2, 1},
        {-2, 5, -2},
        {1, -2, 1}
    };
    for (int i = 1; i < srcHeight - 1; ++i) {
        for (int j = 1; j < srcWidth - 1; ++j) {
            int sumR = 0, sumG = 0, sumB = 0;
            // 對每個像素應用拉普拉斯濾波器
            for (int k = -1; k <= 1; ++k) {
                for (int l = -1; l <= 1; ++l) {
                    sumR += src[i + k][j + l].r * laplacianFilter[k + 1][l + 1];
                    sumG += src[i + k][j + l].g * laplacianFilter[k + 1][l + 1];
                    sumB += src[i + k][j + l].b * laplacianFilter[k + 1][l + 1];
                }
            }
            // 將結果設為目標像素
            dest[i][j].r = int(min(float(255), max(float(0), float(sumR * factor))));
            dest[i][j].g = int(min(float(255), max(float(0), float(sumG * factor))));
            dest[i][j].b = int(min(float(255), max(float(0), float(sumB * factor))));
            dest[i][j].a = src[i][j].a;
        }
    }
}
```

原圖	Filter 1	Filter 2
		
原圖	使用 filter = {1, -2, 1} {-2, 5, -2} {1, -2, 1} 與原圖在眼睛部分有些許的銳化，但實際效果沒有很明顯。	使用 filter = {-1, -1, -1}, {-1, 9, -1}, {-1, -1, -1} 此 filter 的深淺變化更加敏感，因此會讓深淺變化的地方，有更大幅度的銳化，整體的邊緣也更加明顯。

## ● Denoise

此任務須將圖片中的躁點去除，使用的方法是將該點附近的元素加總取平均，達到柔化的效果，且選取附近周圍的數量會在下方依序做討論。Filter 會以基數方式來變化。

```
void denoise(const RGBA** src, RGBA** dest, int srcWidth, int srcHeight, int factor = 1) {
    int f = factor;
    for (int i = f; i < srcHeight - f; ++i) {
        for (int j = f; j < srcWidth - f; ++j) {
            int sumR = 0, sumG = 0, sumB = 0;

            // Apply a simple averaging filter
            for (int k = -f; k <= f; ++k) {
                for (int l = -f; l <= f; ++l) {
                    sumR += src[i + k][j + l].r;
                    sumG += src[i + k][j + l].g;
                    sumB += src[i + k][j + l].b;
                }
            }

            int pixels = (2 * f + 1) * (2 * f + 1);
            // Calculate the average pixel value
            dest[i][j].r = static_cast<uint8_t>(sumR / pixels);
            dest[i][j].g = static_cast<uint8_t>(sumG / pixels);
            dest[i][j].b = static_cast<uint8_t>(sumB / pixels);
            dest[i][j].a = src[i][j].a;
        }
    }
}
```

原圖	3x3 平均	5x5 平均
		
原圖	使用 3x3 filter 來取平均，降低特別突出點的值，且去躁效果佳。	使用 5x5 filter 來取平均，降低特別突出點的值，且去躁效果更好，但隨著 filter 數越來越大，圖片的解析度也會隨之降低，可以搭配第二小題來做銳化。