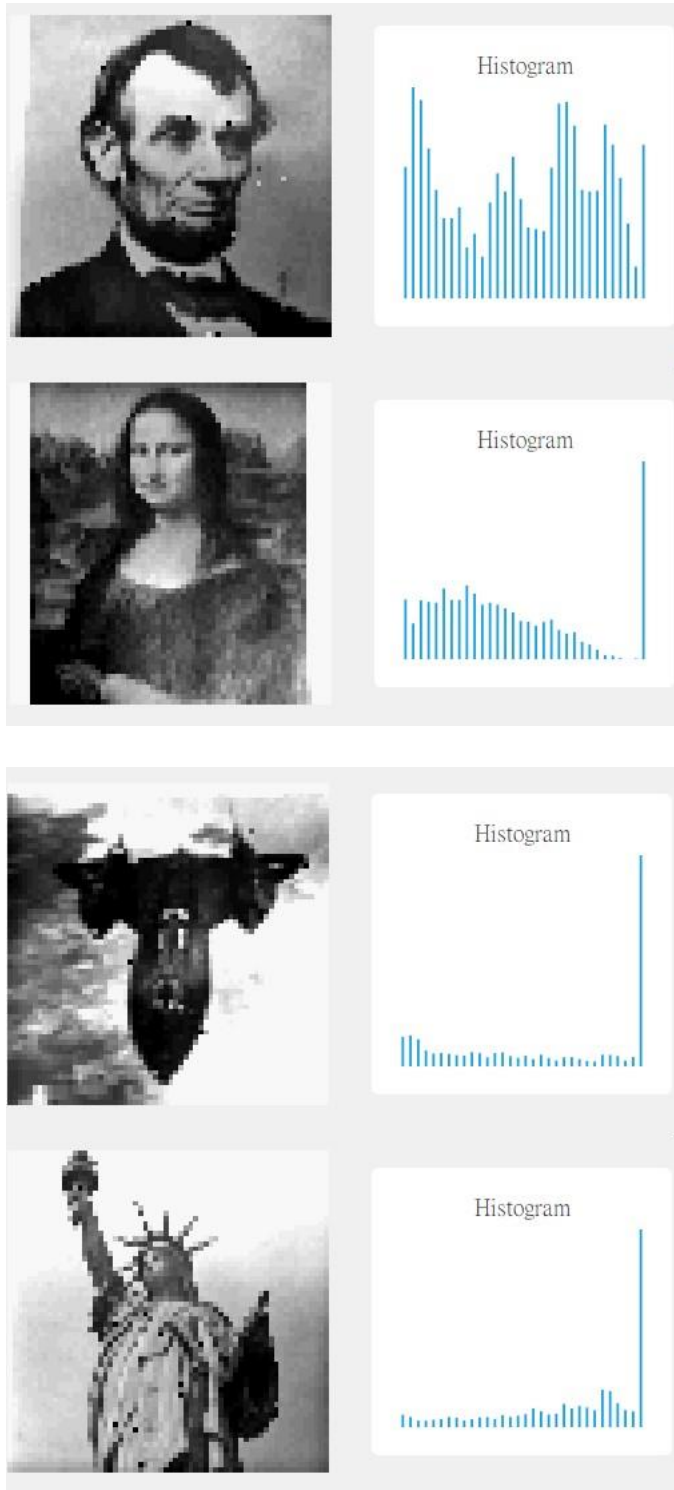


Principle and Applications of Digital Image Processing

Homework 1 Report 林東甫 R12631055

Part 1: (50%) Histogram of an Image



參照老師所製作的範例程式碼

```
ifstream imagefile;
imagefile.open(fileName.toStdString());

if (imagefile.is_open())
{
    for(i=0; i<64; i++)
    {
        for(j=0; j<64; j++)
        {
            // Get the character and convert it to integer ranging from 0 to 31
            image[i][j] = imagefile.get();
            // cout <<image[i][j]; // Use this line to check if the input is correct

            if( (image[i][j] >= '0') && (image[i][j] <= '9') )
            {
                image[i][j] = (image[i][j]-'0');
            }
            else
            {
                image[i][j] = ((image[i][j]-'A') + 10);
            }
        }
        a = imagefile.get(); // Discard the end of line character
        // cout << a; // Use this line to check if the input is correct
    }
    imagefile.close();
}

// Set up QImage for displaying it in the QLabel label

QImage img(64, 64, QImage::Format_RGB32);
for(i=0;i<64;i++)
{
    for(j=0;j<64;j++)
    {
        // Set the pixel value of the QImage
        img.setPixel(j,i,qRgb(image[i][j]*8,image[i][j]*8,image[i][j]*8));
    }
}

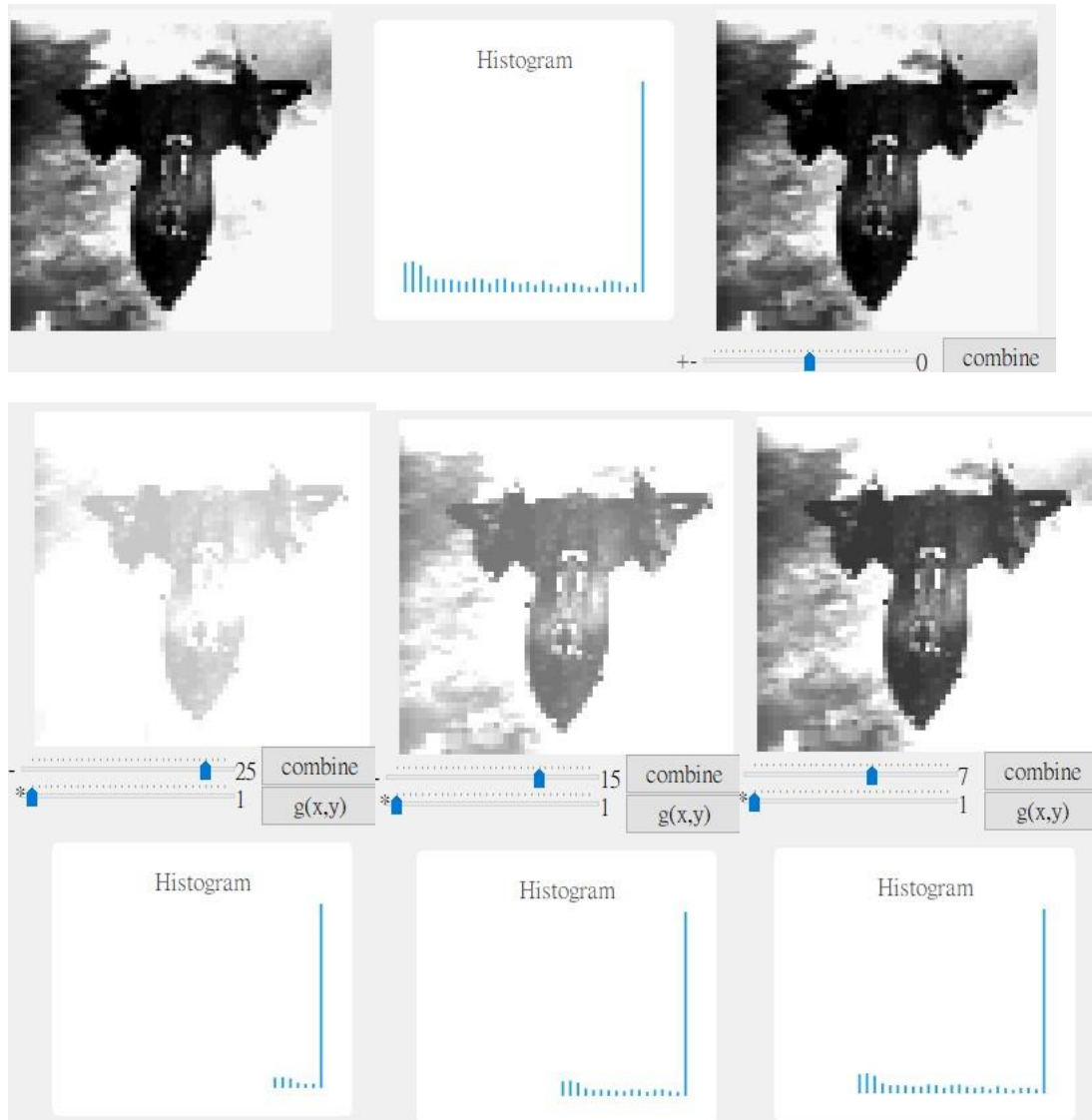
for( i=0; i<32; i++ ) histogram[i] = 0; /* Initialize the array */
for(i=0; i<64; i++)
{
    for(j=0; j<64; j++)
    {
        histogram[ image[i][j] ]++;
    }
}
```

先將.64 檔案讀取進來，將二維陣列的 ASCII charater 轉換 0~31 interger。

接著製作成一張圖片上各點的 pixel，最後將二維陣列中的 int 統計成直方圖。

Part 2: (50%) Arithmetic Operations of an Image Array

1. Add or subtract a constant value to each pixel in the image.

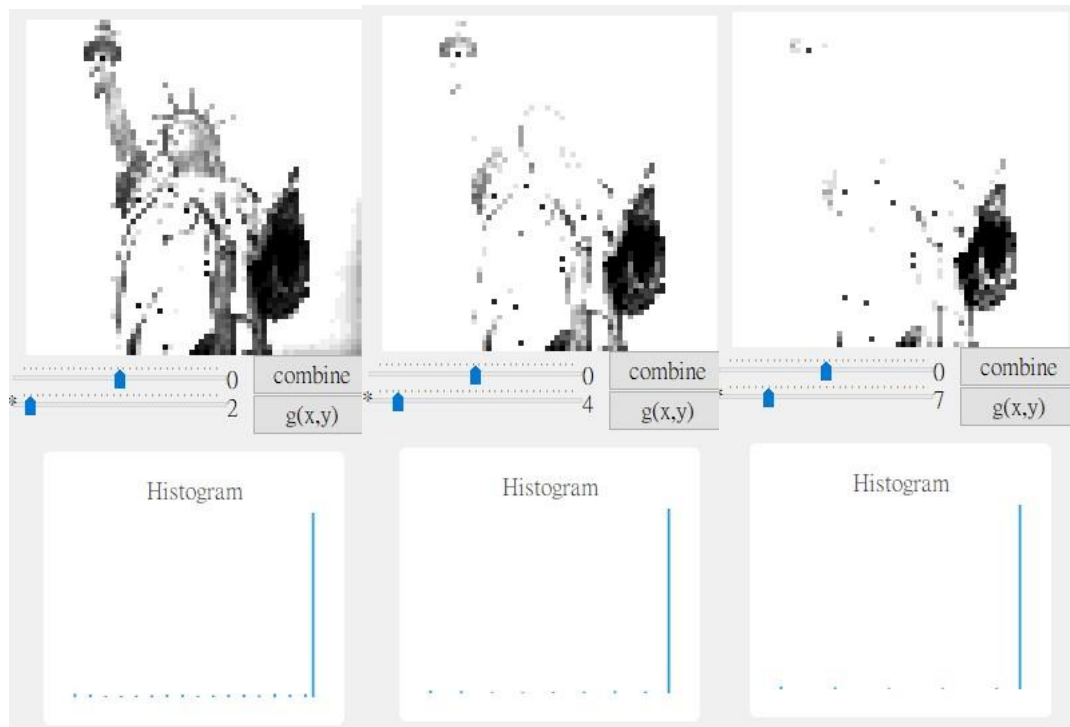


將每一個 pixel 都加上(或減去)一固定常數，以 qBound 設定邊界為 0-255

可以觀察到數值越高畫面整體較明亮直到近乎全白，反之較黑直到近乎全黑。

```
QImage img(64, 64, QImage::Format_RGB32);
for(i=0; i<64; i++)
{
    for(j=0; j<64; j++)
    {
        // Set the pixel value of the QImage
        int x = (image[i][j] + position)*8;
        x = qBound(0,x,255);
        img.setPixel(j,i,qRgb(x,x,x));
    }
}
```

2. Multiply a constant to each pixel in the image.



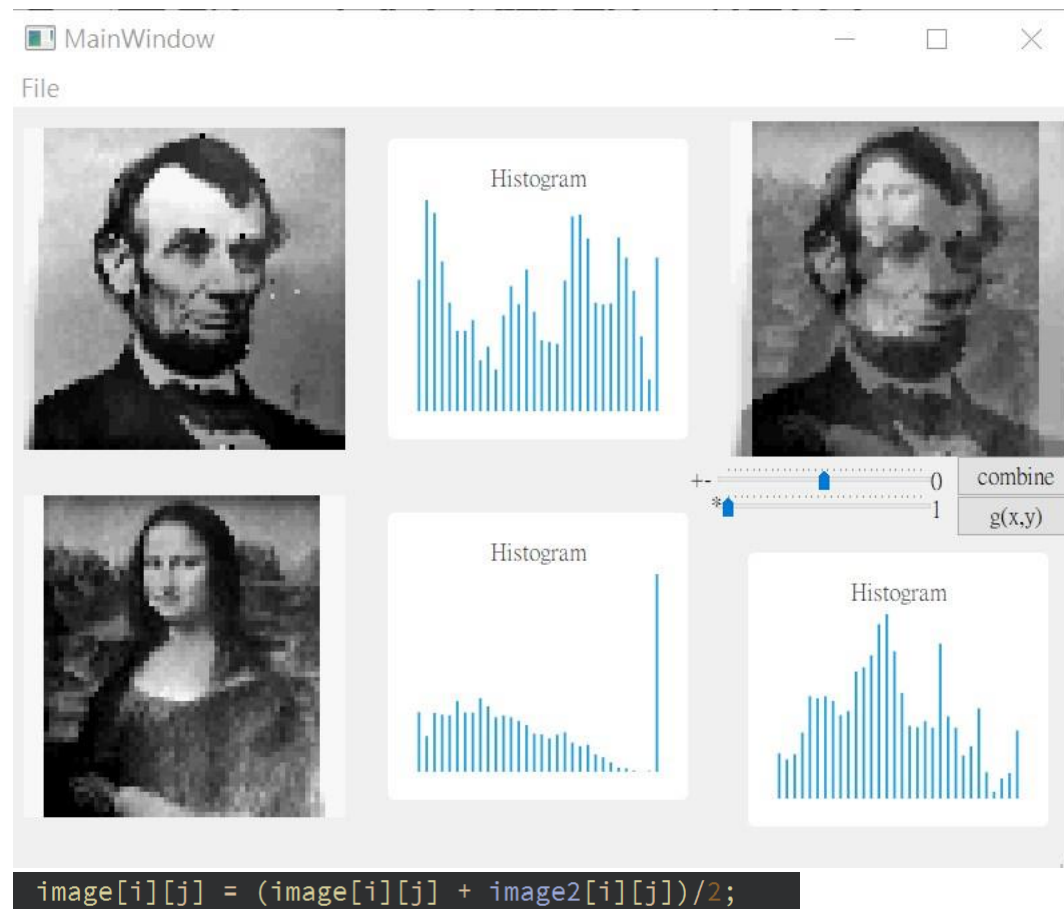
```
int x = (image[i][j] * position * 8);  
x = qBound(0,x,255);  
img.setPixel(j,i,qRgb(x,x,x));
```

將每一個 pixel 都乘上一固定常數，可以觀察到明亮效果更劇烈，同時因同一

乘數的關係，統計直方圖上的 pixel 樣本點集中在該常數的倍數的各組當中。

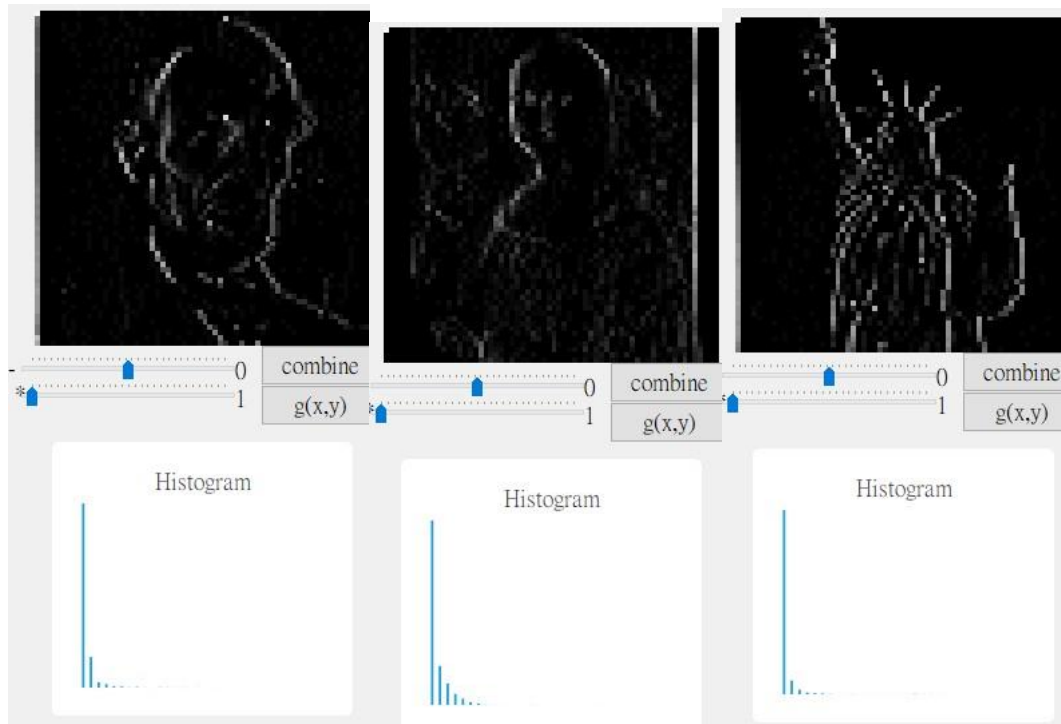
有個別點維持同樣深黑色，這是由於 0 的乘法同一律，0 乘上任何數皆為 0。

3. Create a new image which is the average image of two input images.



取兩張影像的平均值，將兩張影像各點相加後除以二，由於取了平均所以不需要擔心超過 upper bound 因為是隨機影像取平均的關係，統計直方圖看似很像常態分佈，可以看做是兩張影像的疊加。

4. Create a new image $g(x,y)$ in which the value of each pixel is determined by calculating the pixel values of the input image $f(x,y)$ using the following equation: $g(x,y) = f(x,y) - f(x-1,y)$



```
int x = (image2[i][j]-image[i][j-1])*8;  
x = qBound(0,x,255);  
img.setPixel(j,i,qRgb(x,x,x));
```

將一張影像上的所有像素點減去前一行(左邊)相鄰的像素點，因為是實作上是減法所以要小心結果不要超出邊界，整張影像幾乎是全黑的，惟有一些較亮點勾勒出輪廓，這恐怕是因為輪廓邊界越明顯的相鄰處的色差越大，而一般非邊界處色塊中的色差則相對不大。