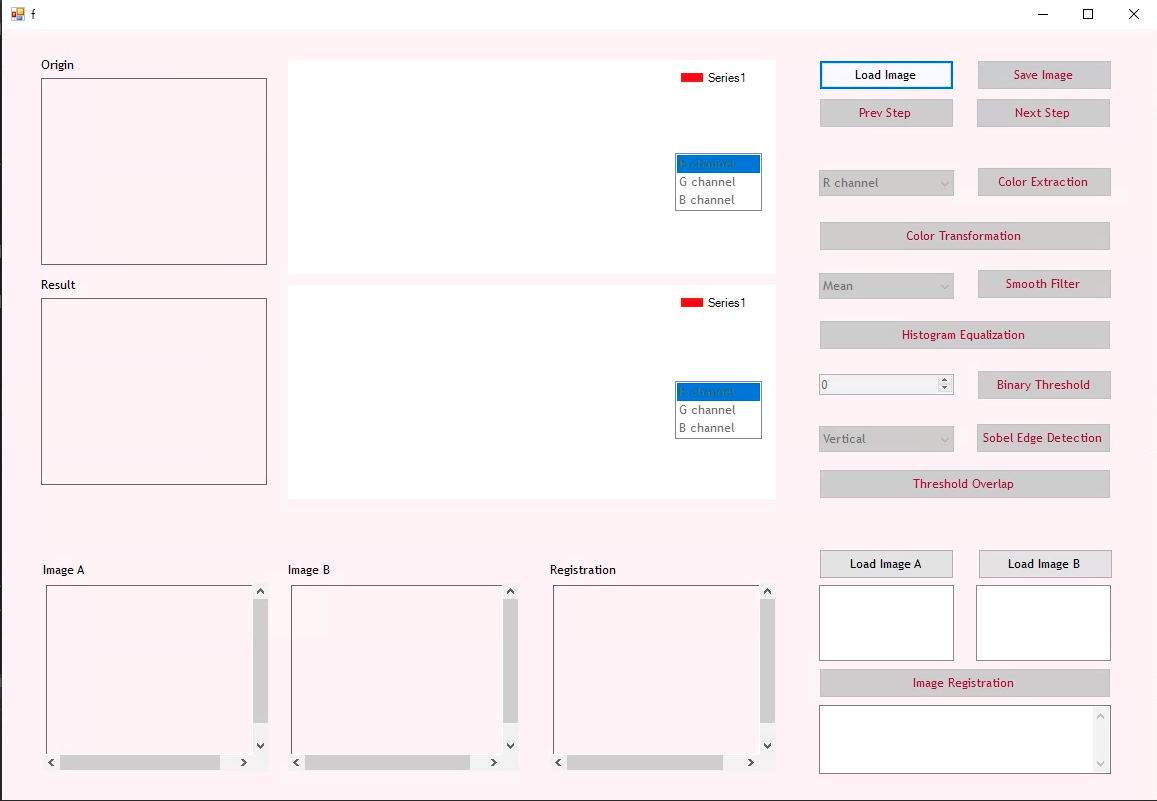
Image Processing HW1

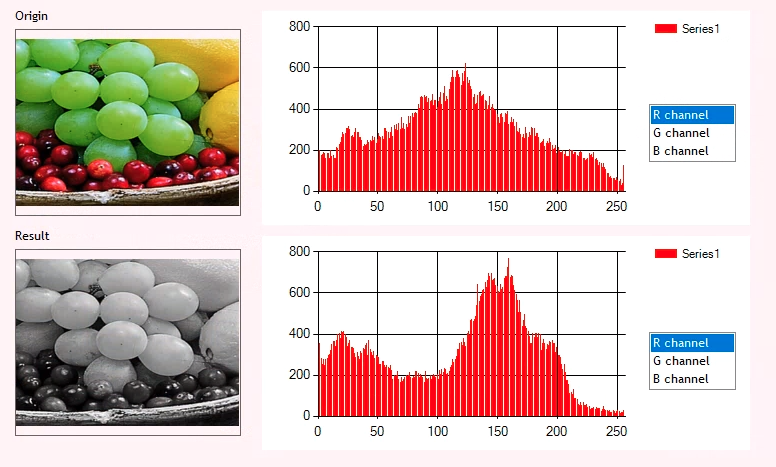
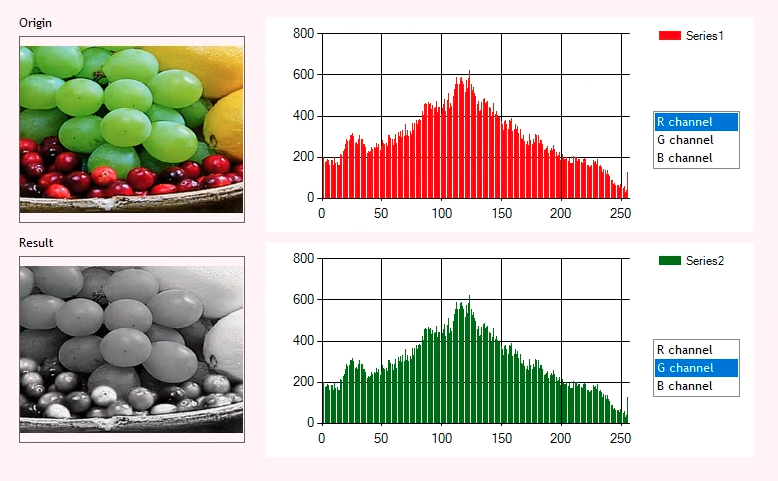
# GUI 界面

* 應用程式分割成上下兩部份，上部份是 1~6 題，下部份是第 7 題
* 點擊 Load Image 載入圖片
* 初始的圖片會同時顯示在 Origin 和 Result 兩個圖片框
* 經過影像處理的圖片會在 Result 的圖片框，而 Origin 圖片框則呈現 Result 的上一步驟，方便比對
* 第 7 題需要載入兩張圖片，故將之分開
* 第 7 題傳入圖片後，在兩張圖片上點擊相應的點，即可點擊 Image Registration 按鈕



# RGB Extraction & Transformation

（下圖左：RGB extraction。下圖右：RGB transformation）



## RGB Extraction

### Problem

Extract R, G, B channel to each image.

### Method

遞迴 Bitmap image 的每個 pixel，取得每個 pixel 的 color，根據選擇的 channel，例如 R channel，將該 pixel 的 G, B 值設定成和 R 值一樣

### Results（上圖左）

會變成灰階圖，將 R, G, B channel 的值設定成一樣的就會是灰階圖

### Discussion

無

### Conclusion

R, G, B channel 的值設定成一樣的就會是灰階圖

## RGB Transformation

* Problem

Change color image to grayscale image.

### Method

遞迴 Bitmap image 的每個 pixel，取得每個 pixel 的 color，三個通道的值各自乘權重後相加，將 pixel 的 R, G, B 值都設成相加的結果

R: 0.299, G: 0.587, B: 0.114

### Results（上圖右）

原圖會變成灰階圖

### Discussion

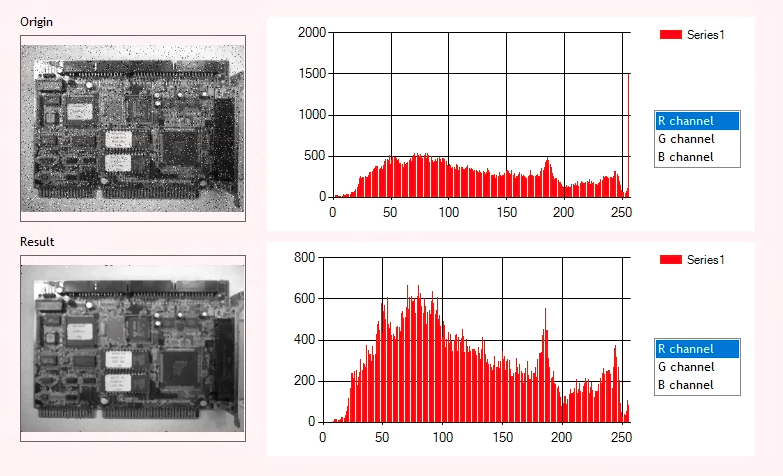
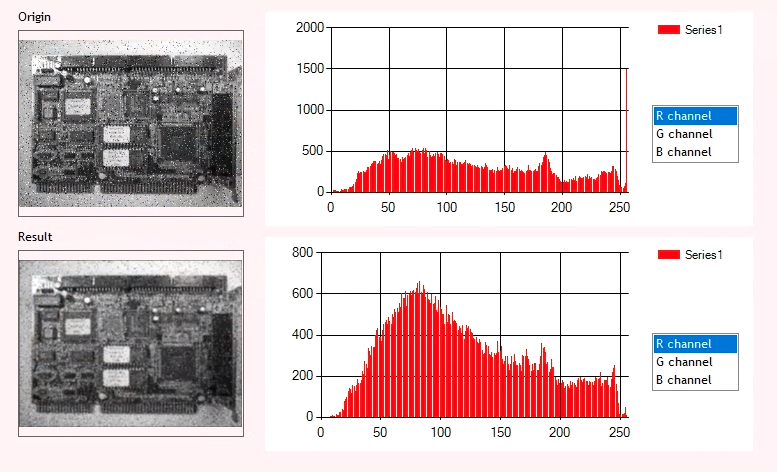
無

### Conclusion

R, G, B channel 的值設定成一樣的就會是灰階圖

# Smooth Filter

（下圖左：mean filter。下圖右：median filter）



## Mean Filter

### Problem

Implement mean filter.

### Method

將原圖加 padding = 1 之後，以 3x3 矩陣 kernel 遞迴圖片，將 kernel 內pixels 的 color 取平均後，再將 kernel 內 pixels 的 color 設定成平均後的值

### Results（上圖左）

會變得模糊，但是噪點並沒有被移除

### Discussion

無

### Conclusion

因為是取平均，噪點的數值也會一起算入平均，所以無法有效移除

## Median Filter

### Problem

Implement median filter.

### Method

將原圖加 padding = 1 之後，以 3x3 矩陣 kernel 遞迴圖片，將 kernel 內pixels 的 color 排序取中位數後，再將 kernel 內 pixels 的 color 設定成中位數的值

### Results（上圖右）

噪點被有效移除

### Discussion

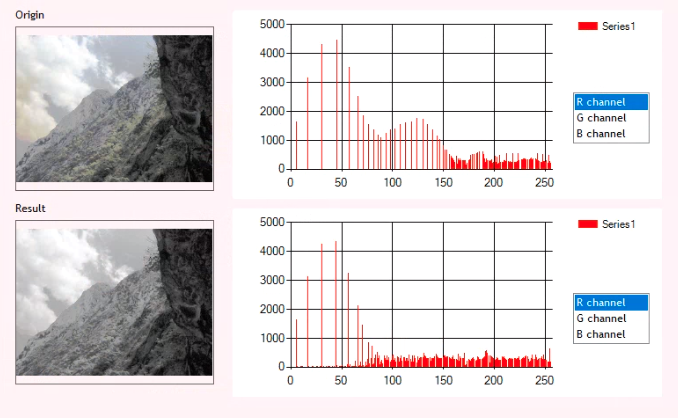
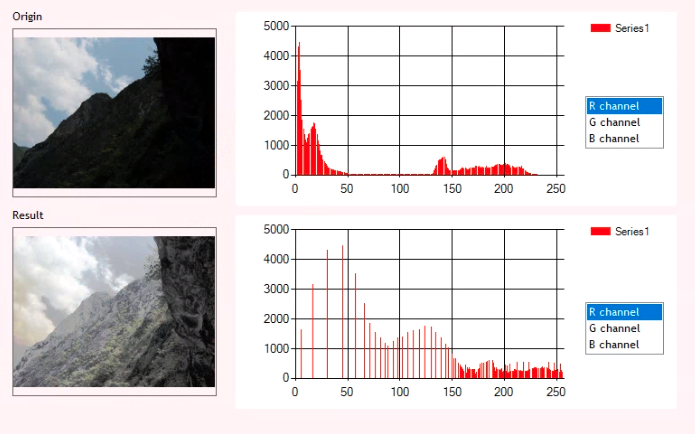
無

### Conclusion

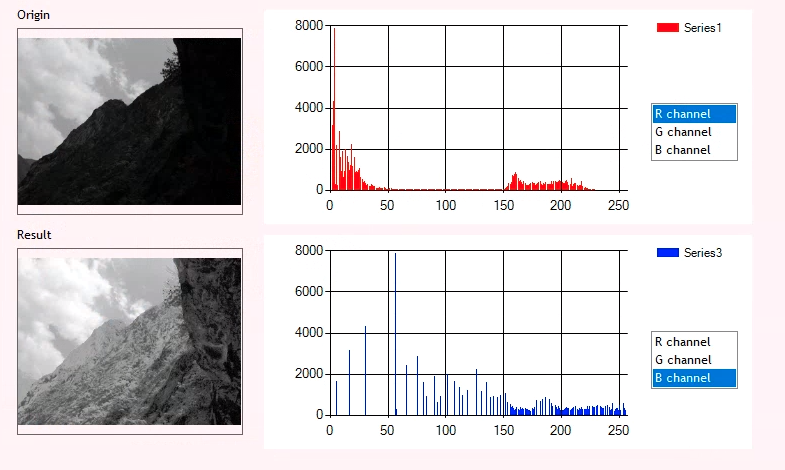
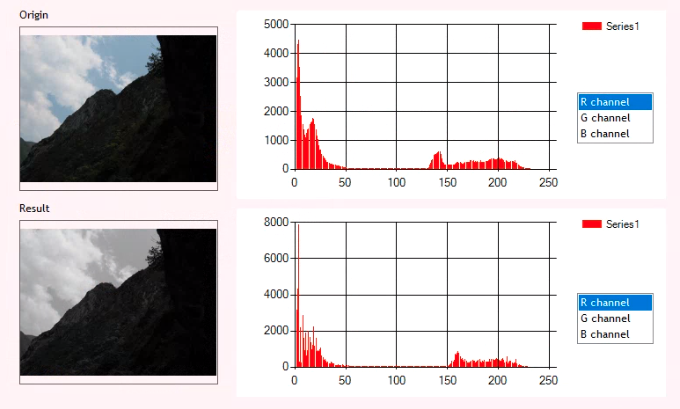
因為取中位數，故低於中位數的噪點會被直接過濾

# Histogram Equalization

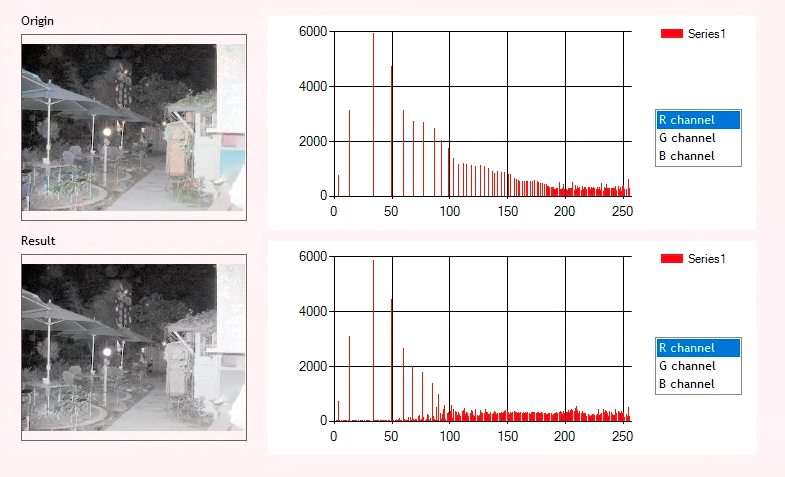
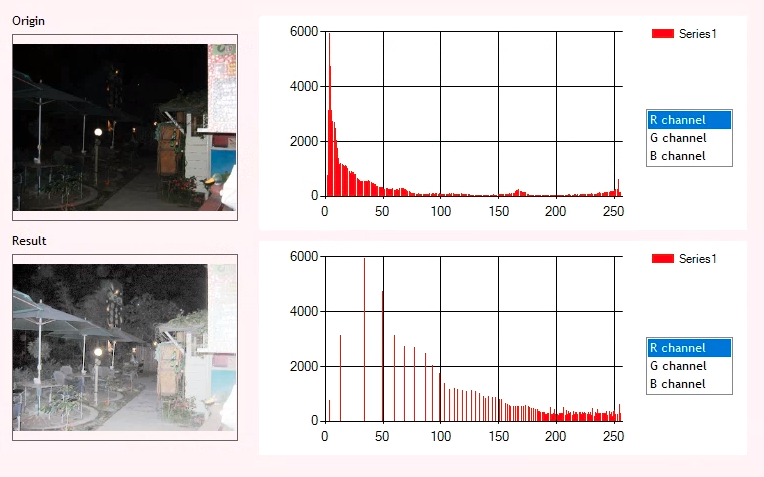
（color image 做 histogram equalization（下圖左）後轉 grayscale（下圖右））



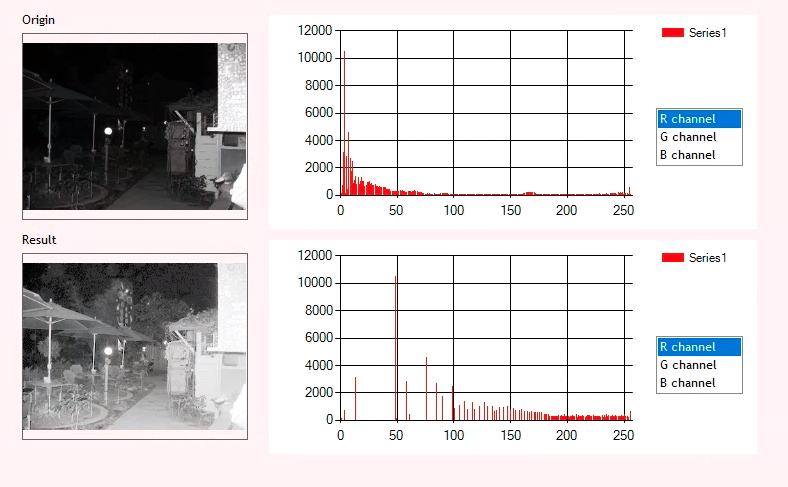
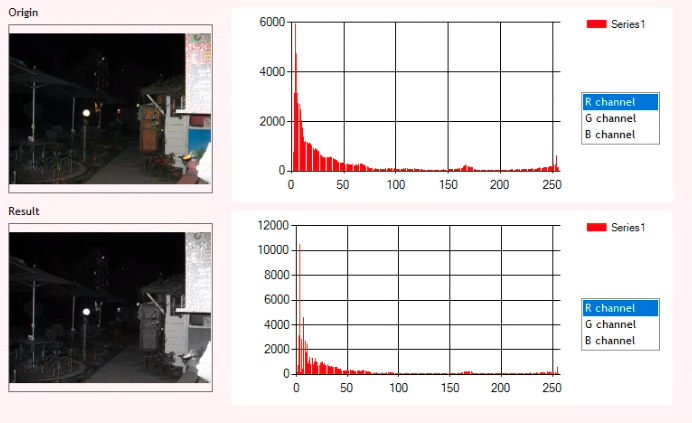
（color image 轉 grayscale（下圖左）後做 histogram equalization（下圖右））



（color image 做 histogram equalization（下圖左）後轉 grayscale（下圖右））



（color image 轉 grayscale（下圖左）後做 histogram equalization（下圖右））



## Histogram Equalization

### Problem

Implement histogram equalization.

### Method

計算每個 color 出現的次數，遞迴 image 取得 pixel 的 color 值，以長度 256 的矩陣儲存（ histogram ），根據 CDF \* 25 將每個 pixel 的 color 值重新分配

### Results（上圖左）

圖片的內容會變得比較明顯

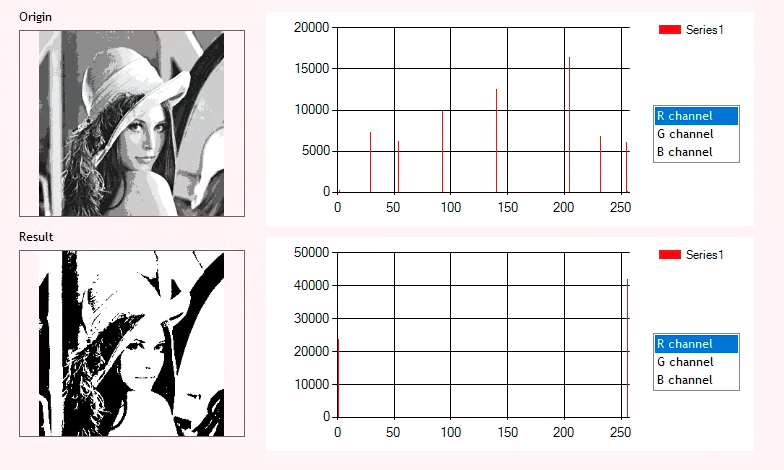
### Discussion

將 color image 先做 histogram equalization 再轉成 grayscale 會比反過來的還要更均勻一點

### Conclusion

是滿實用的影像處理

# User Defined Threshold



## User-defined Threshold

### Problem

Implement user-defined threshold

### Method

遞迴圖片的每個 pixel 取得 color 的數值，根據使用者的輸入來判斷將 pixel color 設成 255 還是 0

### Results（上圖左）

非黑即白的圖

### Discussion

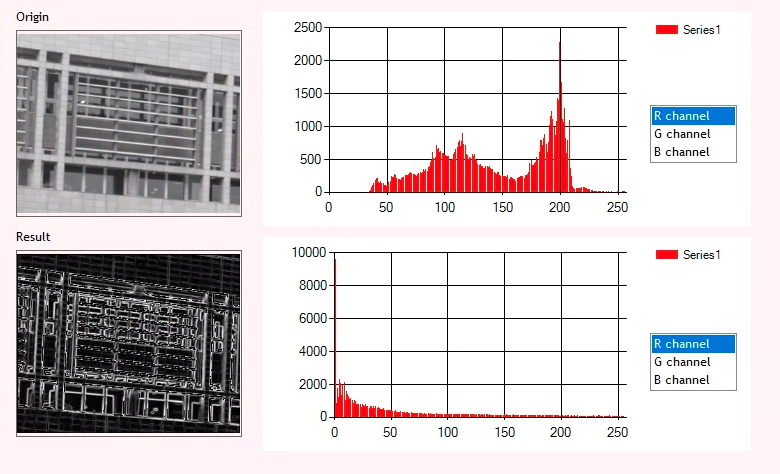
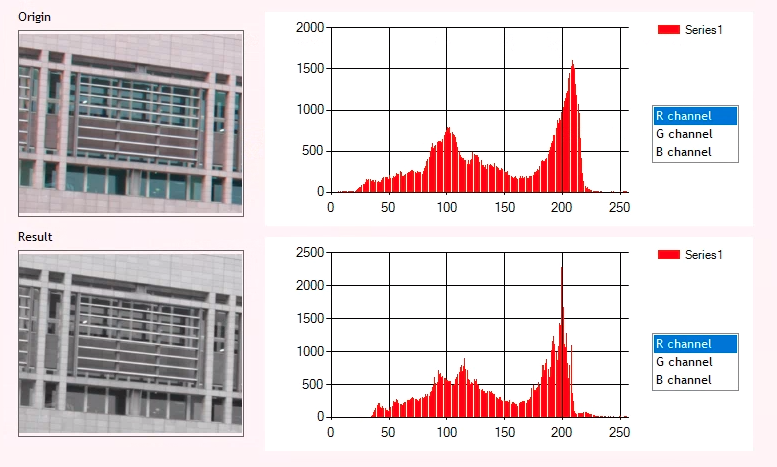
無

### Conclusion

覺得人腦補的能力滿好的，不存在的線條還是可以想像得出來

# Sobel Edge Detection

（color image 轉 grayscale（下圖左）後做 edge detection（下圖右））



## Sobel Edge Detection

### Problem

Implement sobel edge detection

### Method

先將圖片加上 padding = 1 後，用 3x3 array 當作 kernel 遞迴掃過每個 pixel，根據 vertical, horizontal, or combined 的需求以不同的 kernel 掃過

### Results（上圖左）

找到圖片中的邊緣

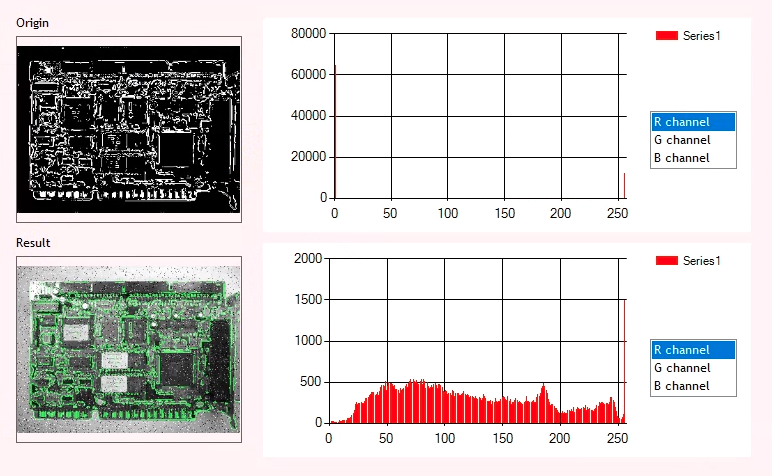
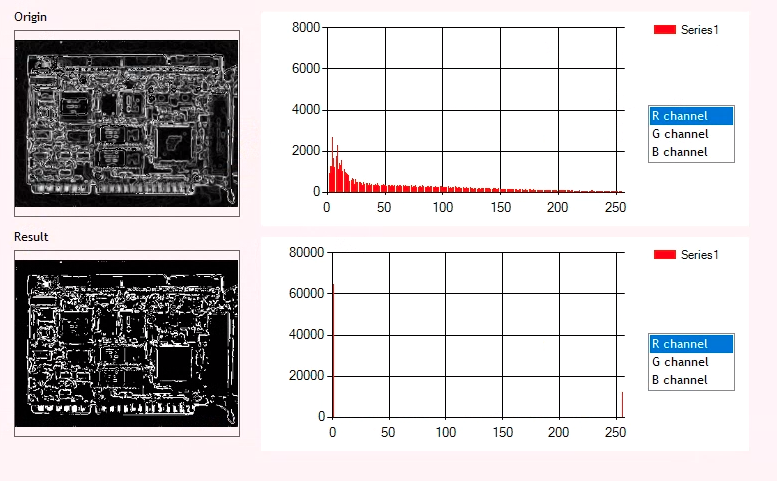
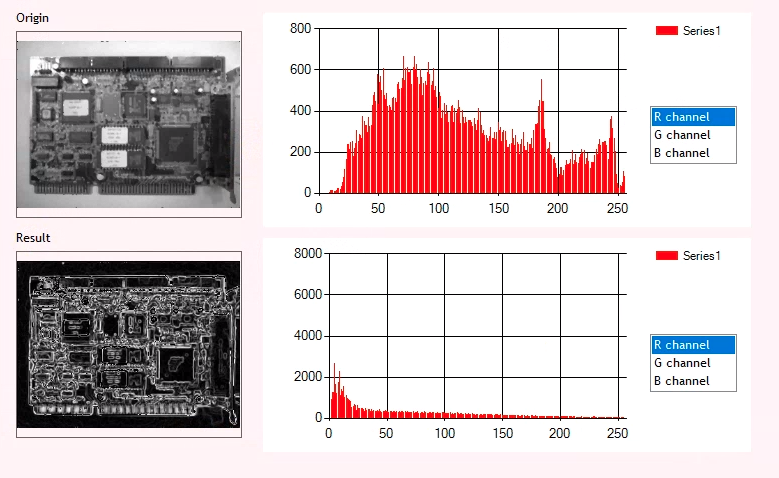
### Discussion

無

### Conclusion

找到圖片中的邊緣

# Threshold and Overlap



## Threshold and Overlap

### Problem

Implement overlap

### Method

由上一題找到 edge 後過 binary threshold，將 color 為 0 的 pixel 設 G channel = 255，稱為 mask。根據 mask G channel = 255 的 pixel 位置將原圖的Ｇ channel 設成 255

### Results（上圖左）

疊圖

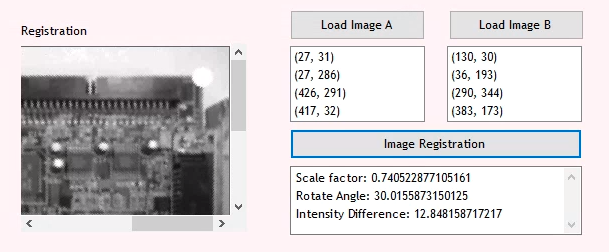
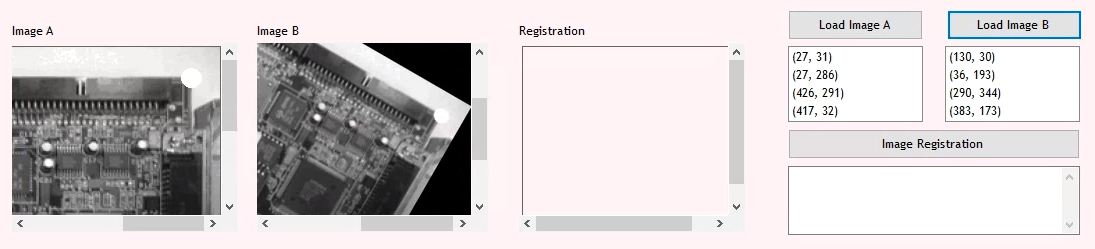
### Discussion

無

### Conclusion

將邊緣標註為綠色

# Image Registration



## Image Registration

### Problem

Implement image registration

### Method

根據兩張圖片點擊 4 個點的順序，取其中的 3 個點，將原座標乘上後來座標的反矩陣，即可得到 transform matrix，將 matrix 中的值取出運算後得到旋轉角度、縮放比例、和偏移量

### Results（上圖左）

image registration

### Discussion

無

### Conclusion

圖片因為縮小後再放大回來有點模糊