

高等影像處理

作業四書面報告

學號： 61247014S

姓名： 陳昱誠

一、本作業所用之程式語言及編譯器：Python

二、程式功能

1. 輸入的影像為灰階影像，輸出則為卷積運算後之影像，輸出之影像可能分別會有影像平滑化以及邊緣偵測等效果。
2. 本程式可讀入的影像檔格式包含 JPG 檔、BMP 檔，以及 PPM 檔，輸出的影像檔格式不拘。

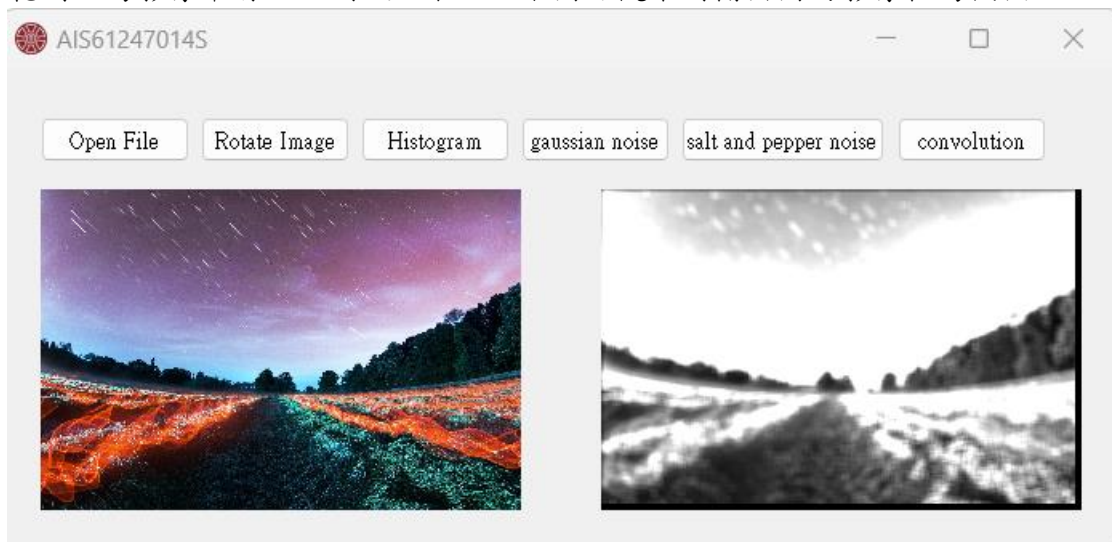
三、程式流程或演算法

(請附程式流程圖或演算法)

- 點擊「Open File」按鈕
- 顯示對話方塊選擇開啟檔案
- 在 PyQt 中的 canvas1&canvas2 顯示開啟之圖片
- 點擊「convolution」按鈕
- 對話框中輸入 5x5 矩陣之值
- 將 picInput 影像轉為灰階影像
- 依據輸入之矩陣進行 convolution 並輸出於 canvas2

四、測試結果(請附至少三組程式畫面截圖，並附相關說明)

範例一為影像平滑化之測試結果。由下圖可觀察到輸出圖的影像較為模糊。



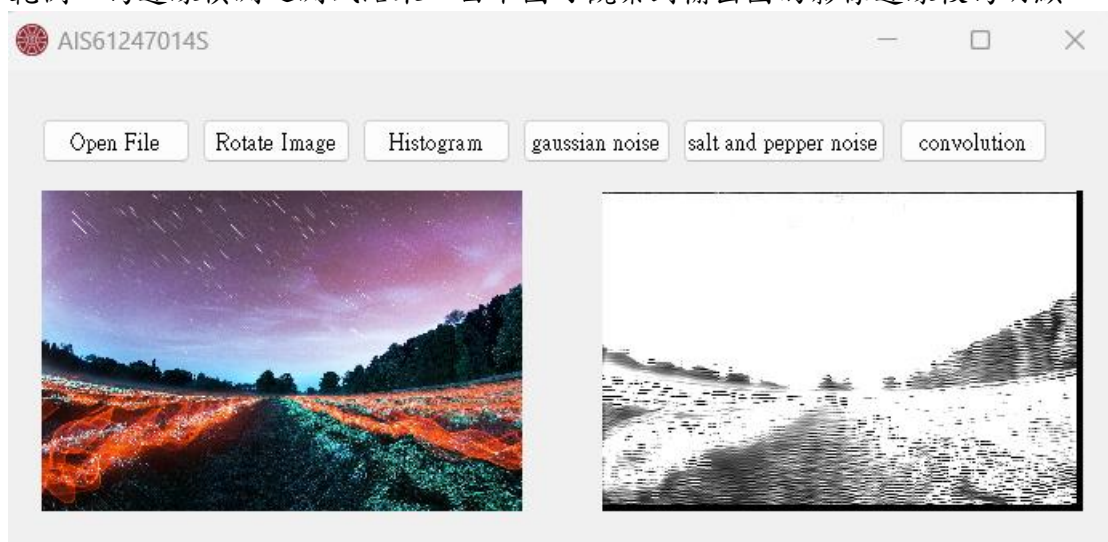
範例一之 convolution mask，為 5*5 數值全為 0.1 之矩陣。

number input window

0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1
0.1	0.1	0.1	0.1	0.1

save

範例二為邊緣偵測之測試結果。由下圖可觀察到輸出圖的影像邊緣較為明顯。



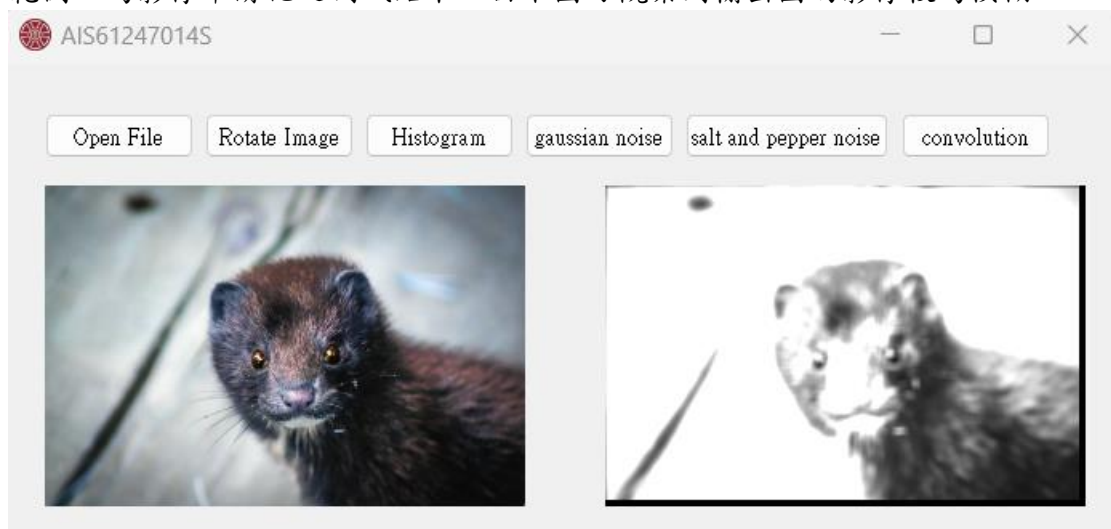
範例二之 convolution mask，為 5*5 之矩陣，數值如圖所示。

number input window

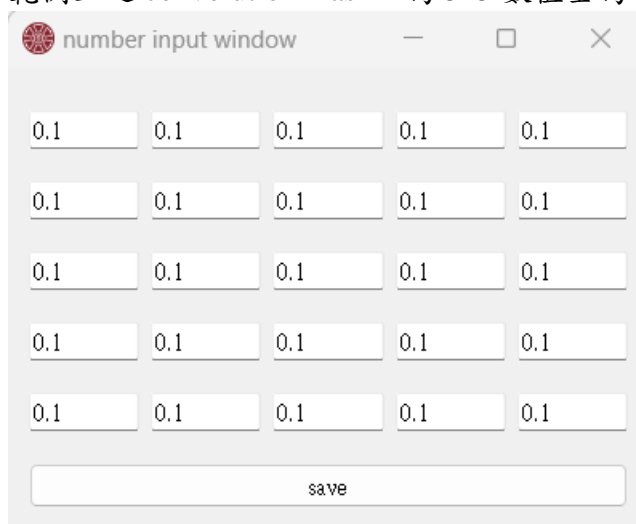
1	1	1	1	1
-1	-1	-1	-1	-1
1	1	1	1	1
-1	-1	-1	-1	-1
1	1	1	1	1

save

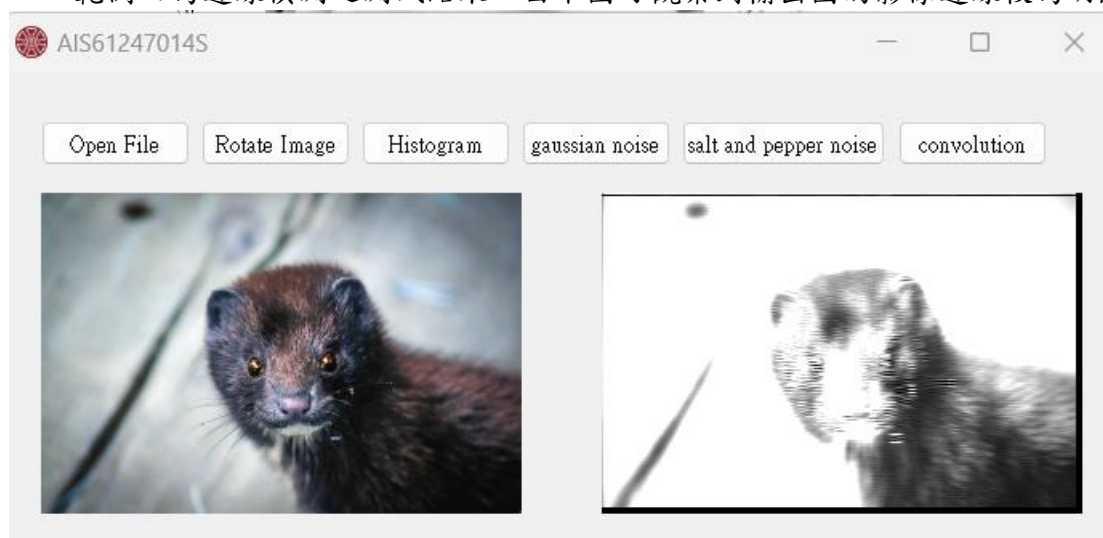
範例三為影像平滑化之測試結果。由下圖可觀察到輸出圖的影像較為模糊。



範例三之 convolution mask，為 5*5 數值全為 0.1 之矩陣



範例四為邊緣偵測之測試結果。由下圖可觀察到輸出圖的影像邊緣較為明顯



範例四之 convolution mask，為 5*5 之矩陣，數值如圖所示。



0.5	0.5	0.5	0.5	0.5
-0.5	-0.5	-0.5	-0.5	-0.5
0.5	0.5	0.5	0.5	0.5
-0.5	-0.5	-0.5	-0.5	-0.5
0.5	0.5	0.5	0.5	0.5

save

五、 程式撰寫心得(至少 100 字)

第四次的作業比上一次的作業簡單一些，除了 convolution 本身的計算程式，這次遇到最大的問題是於視窗設計中如何使值有效傳遞並進行運算，在 PyQt 中提供 slot 方式去避免異部操作時讀值的問題，然後在邊緣偵測以及平滑化的參數測試測了蠻多次，看了許多文章以及參考資料才順利找出合適的參數並完成此次作業，希望下次的作業不要再像上次那麼難 QAQ。