

高等影像處理

作業四書面報告

學號： M11102134

姓名： 張惠深

一、本作業所用之程式語言及編譯器

程式語言：Python 3.9.13

環境：Jupyter Notebook

直譯器：Python 直譯器

套件：PyQt5、pyinstaller(.Py 打包為.exe)、OpenCv 4.6.0

二、程式功能

1. 本程式可讀入的影像檔格式包含 JPG 檔、BMP 檔，以及 PPM 檔，輸出的影像則為經平滑後之影像或經邊緣偵測後影像。
2. 本程式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(依 user set masks 而定)，影像平滑化與邊緣偵測需由使用者輸入不同的 convolution masks，大小至少為 5X5 pixels。

三、程式流程或演算法

1. 介面介紹

1. 載入圖片，按下開啟圖片按鈕，顯示載入圖片，並自動儲存該圖像之灰階圖。
2. 於最右邊 5*5 輸入框內自訂 convolution masks，需全部填滿數值，否則程式自動進入平滑化模式。
3. 自訂完成 convolution masks 後，點擊卷積運算按鈕，根據使用者自訂之 mask，進行卷積運算，並顯示平滑化或邊緣偵測之結果。
4. 按下圖片旋轉按鈕，即可對影像進行旋轉。
5. 按下儲存圖片按鈕，即可對影像進行儲存。
6. 按下右上視窗 X 按鈕，離開程式

四、測試結果(請附至少三組程式畫面截圖，並附相關說明)



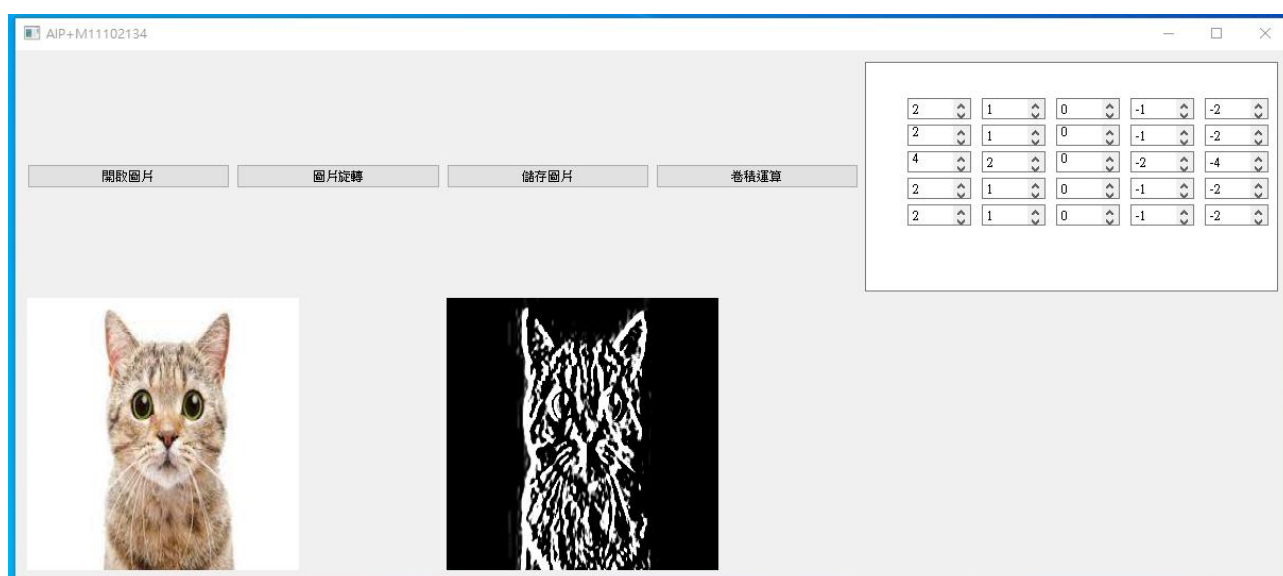
範例一 點擊開啟圖片載入 Lena 影像，設定 5*5 kernel (1)，點擊卷積運算，顯示經影像平滑後之 Lena 影像。



範例二 點擊開啟圖片載入狒狒影像，設定 5*5 kernel (0.04)，點擊卷積運算，顯示經影像平滑後之狒狒影像。



範例三 點擊開啟圖片載入 Lena 影像，設定 5*5 kernel，點擊卷積運算，顯示 edge detection 後之 Lena 圖像。



範例四 點擊開啟圖片載 貓咪 影像，設定 5*5 kernel，點擊卷積運算，顯示 edge detection 後之 貓咪 圖像。

五、 程式撰寫心得(至少 100 字)

透過這次的作業幫助我更加了解平滑化及邊緣偵測之原理，尤其是在自訂 Kernel 部分，原來平滑化及邊緣偵測兩者只差異在 kernel 定義部分，兩者同樣要經 convolution，只是結果不同罷了，這次作業讓我獲益良多，尤其在實際撰寫卷積運算的部分，幫助我更加了解運算的原理及過程，透過程式撰寫與課堂聽講互相驗證，讓我更加了解到卷積運算在影像處理上的應用。