程式概述

這個 Python 程式使用 PyQt5 作為圖形用戶界面 (GUI)，並使用 OpenCV 進行圖像操作。程式主要有三個部分，對應於三種不同的圖像處理技術，每個技術在主視窗class MainWindow(QMainWindow)不同的分頁中呈現：

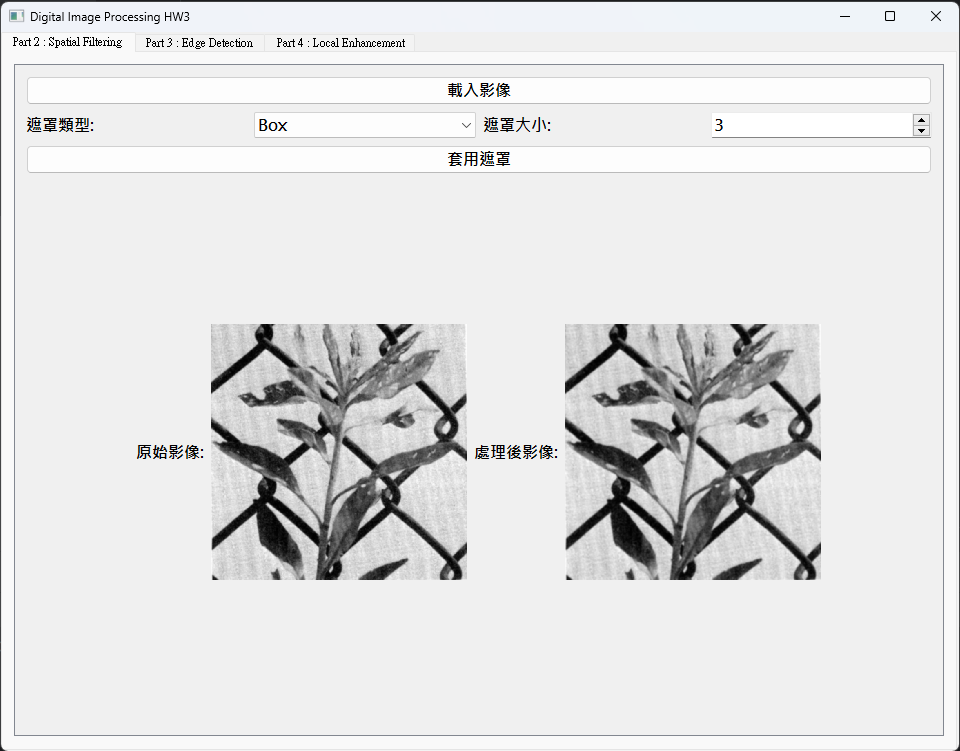
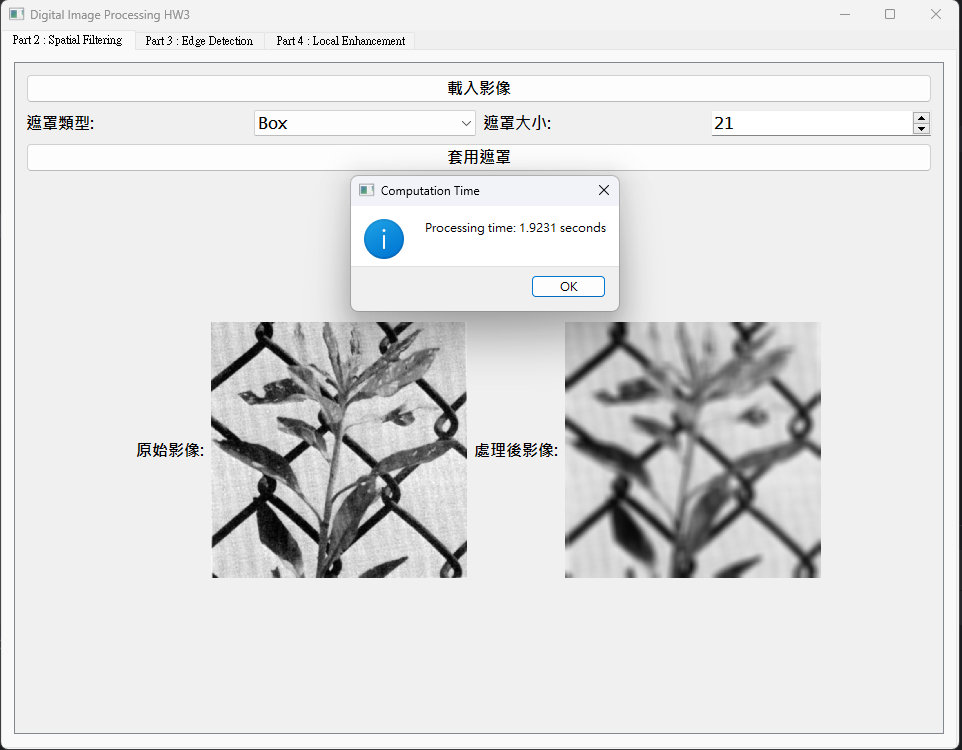
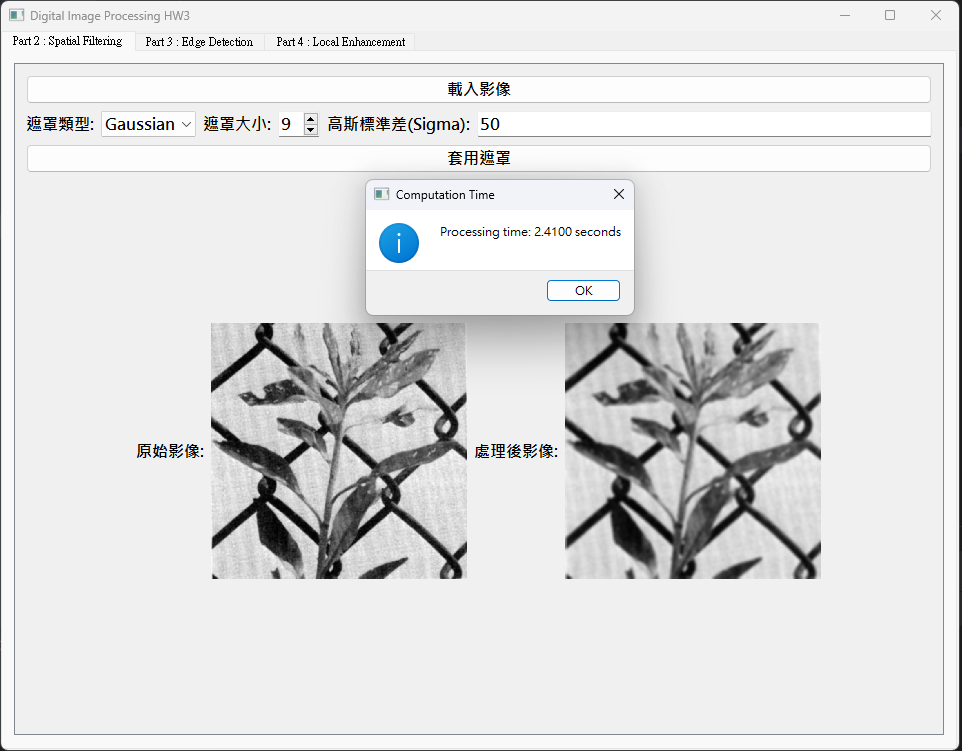
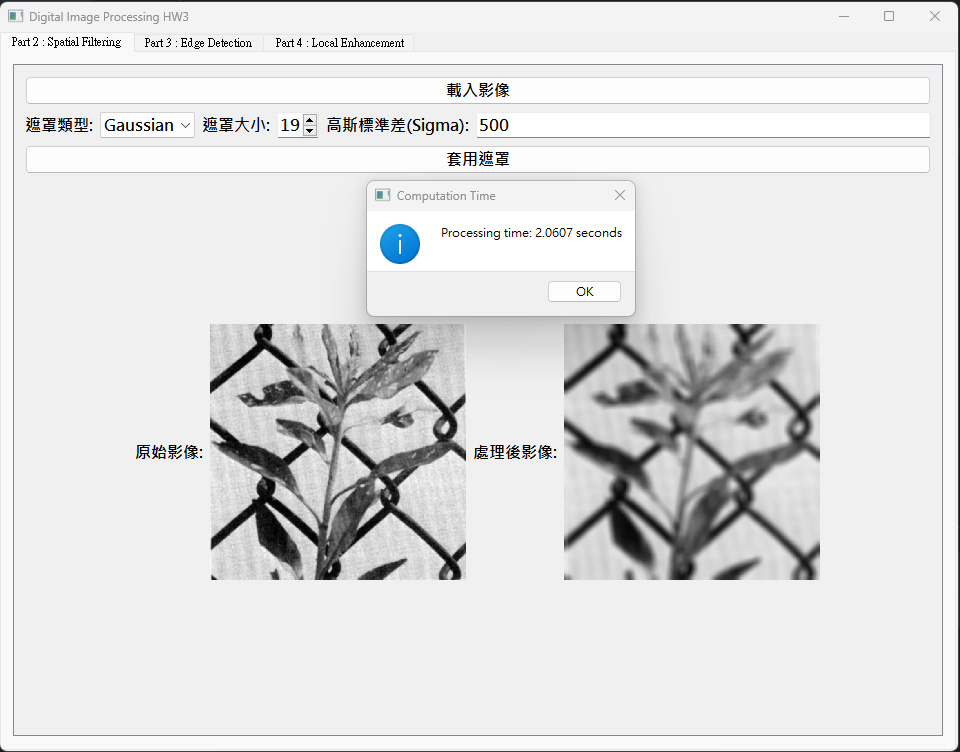
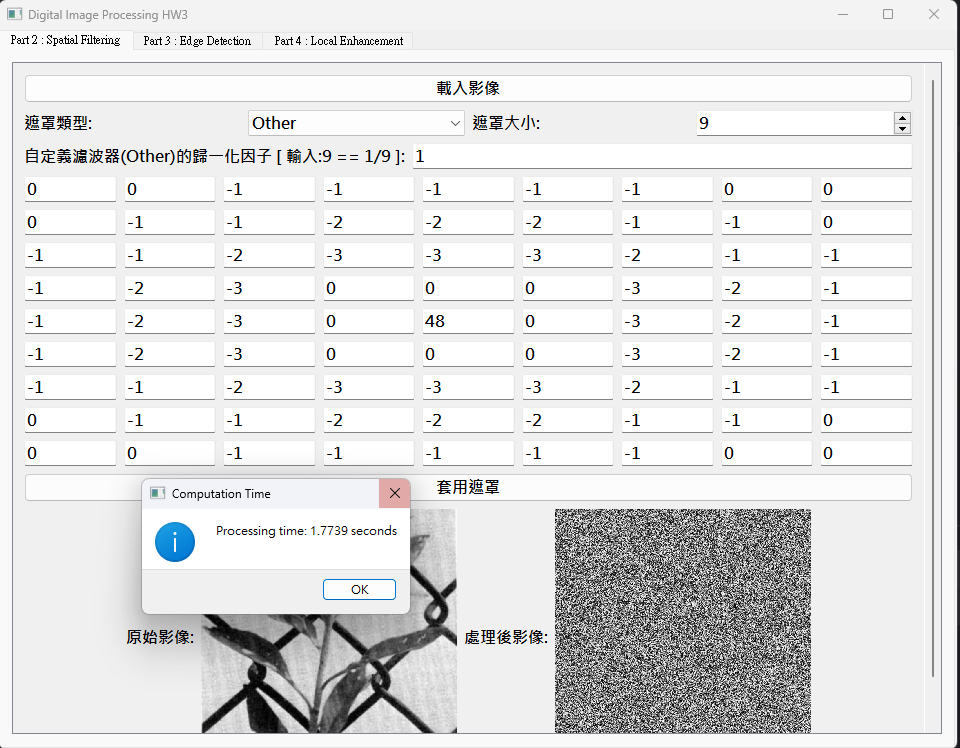
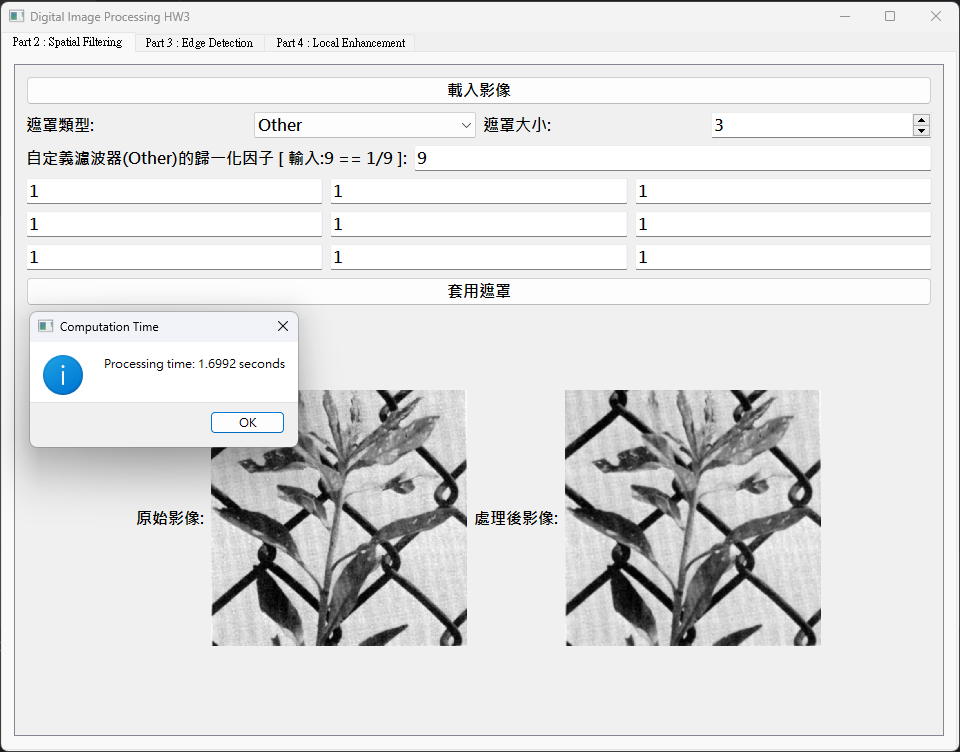
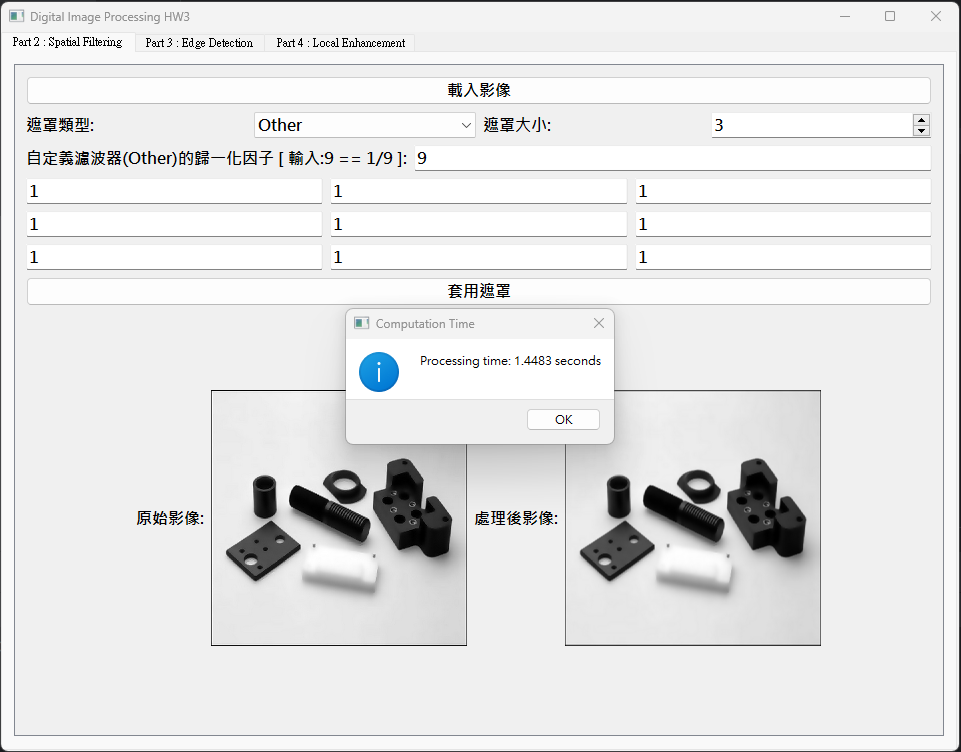
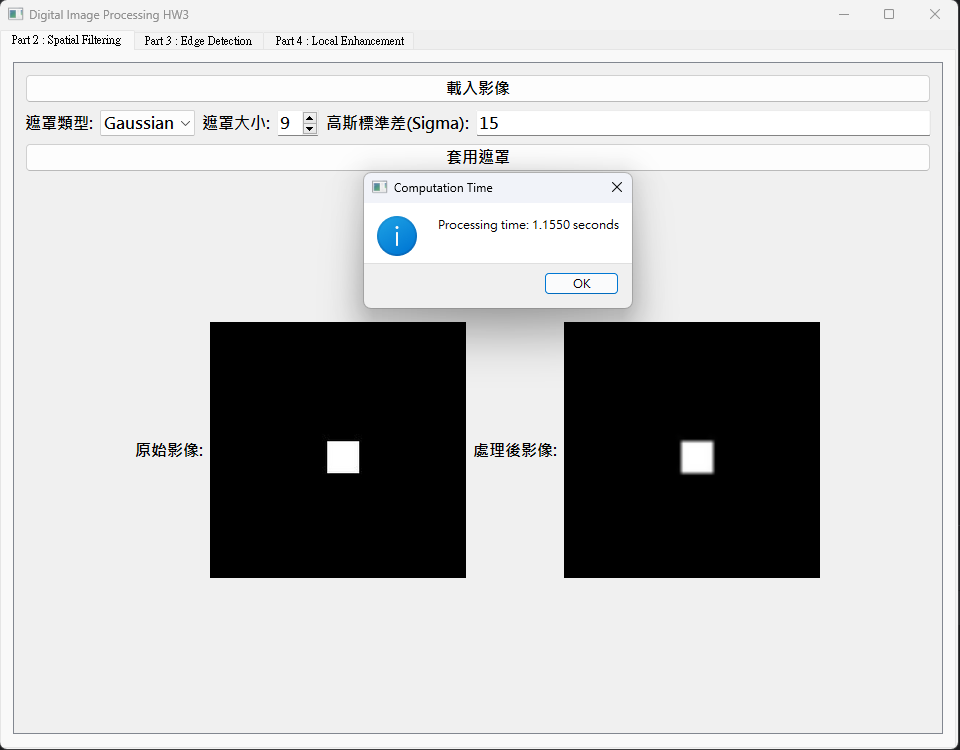
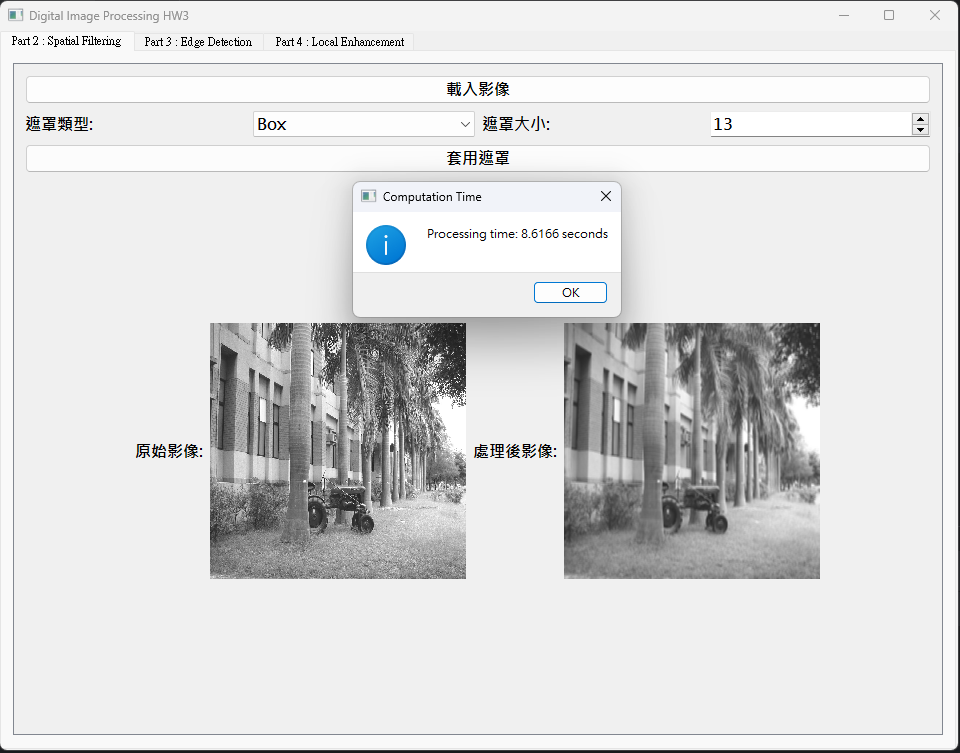
1. Part 2：空間濾波（應用不同遮罩類型的濾波操作）。  
2. Part 3：邊緣檢測（使用 Marr-Hildreth 和 Sobel 邊緣檢測技術）。  
3. Part 4：局部增強（使用局部直方圖統計增強方法來增強圖像對比度）。

## Part 2：空間濾波操作 class SpatialFilteringApp()

主要功能:  
- 遮罩類型：可應用 Box、Gaussian 和自定義遮罩。  
- 遮罩大小：可以調整遮罩大小（範圍從3到21）。  
- Sigma 輸入：對於 Gaussian 遮罩，可以手動調整 sigma 或設置為自動。  
- 自定義過濾器：用戶可以手動輸入值來創建自己的遮罩。

工作原理:  
1. 用戶加載一張圖像。  
2. 選擇過濾器類型和遮罩大小，或是自訂遮罩，或是輸入參數。  
3. 應用過濾器，並顯示處理後的圖像和彈出視窗顯示圖像處理時間。

程式執行截圖說明:

1. Box 3\*3
2. Box 21\*21 
3. Gaussian 9\*9 sigma=50 
4. Gaussian 19\*19 sigma=500 
5. 自定義 9\*9 過大會出現滾動條
6. 自定義Box 3\*3 
7. 測試其他圖片 自定義Box 3\*3 
8. 測試其他圖片 高斯9\*9 sigma=15 
9. 測試其他圖片 Box 13\*13 

程式細節:  
- manual\_gaussian\_kernel 函數生成用於圖像濾波的高斯核。  
- manual\_filter2D 函數使用用戶指定的遮罩執行2D卷積。

### 討論：遮罩大小對處理後圖像和計算時間的影響:

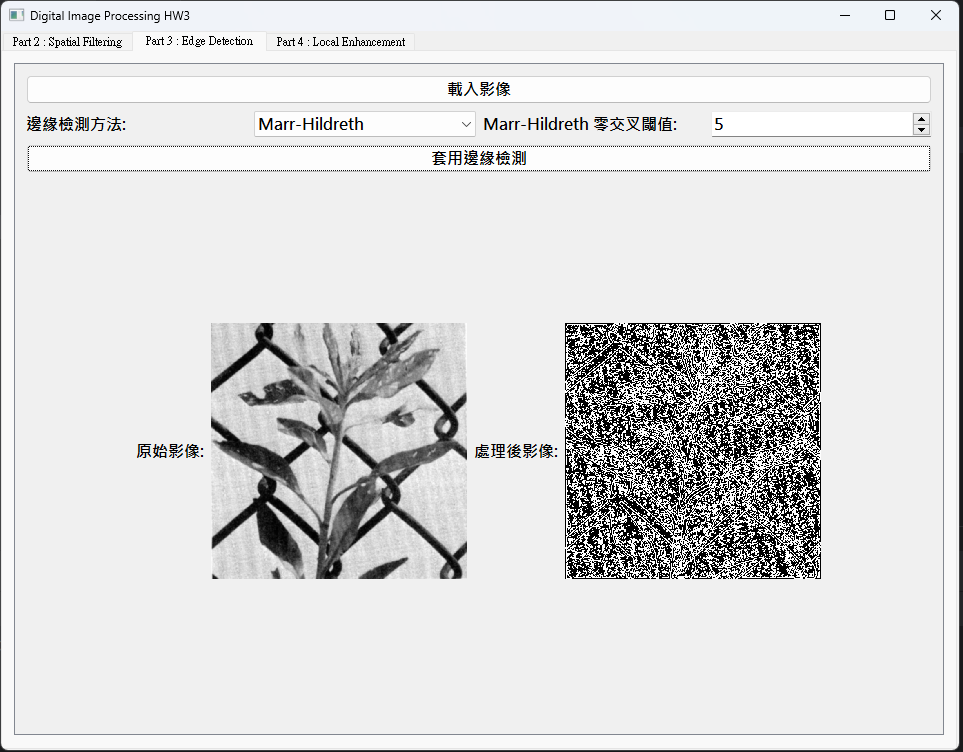
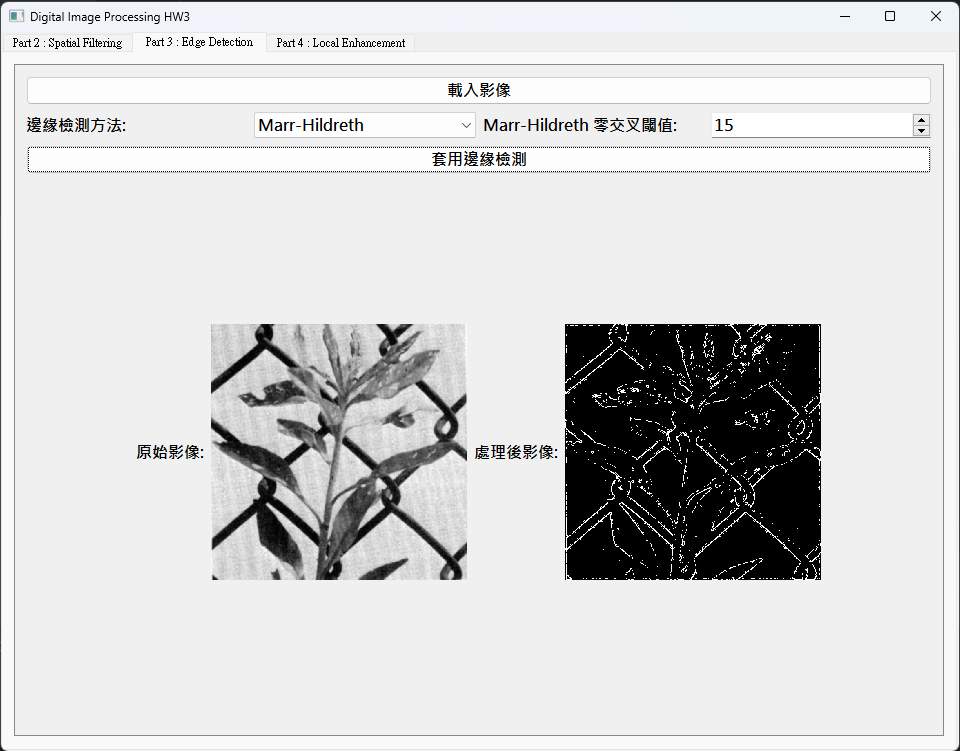
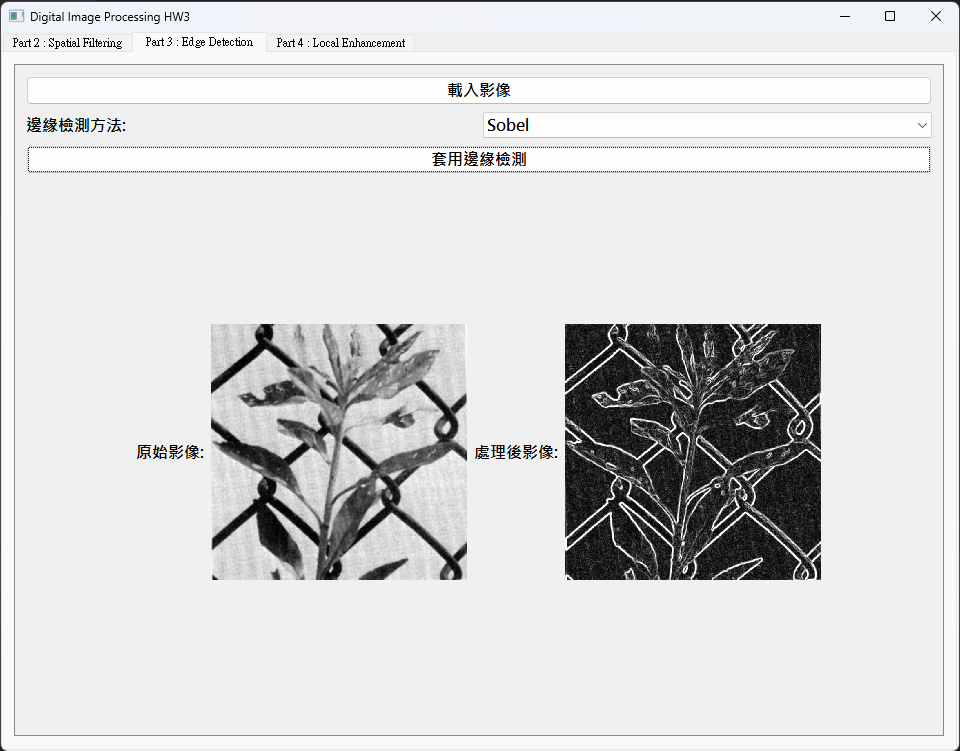
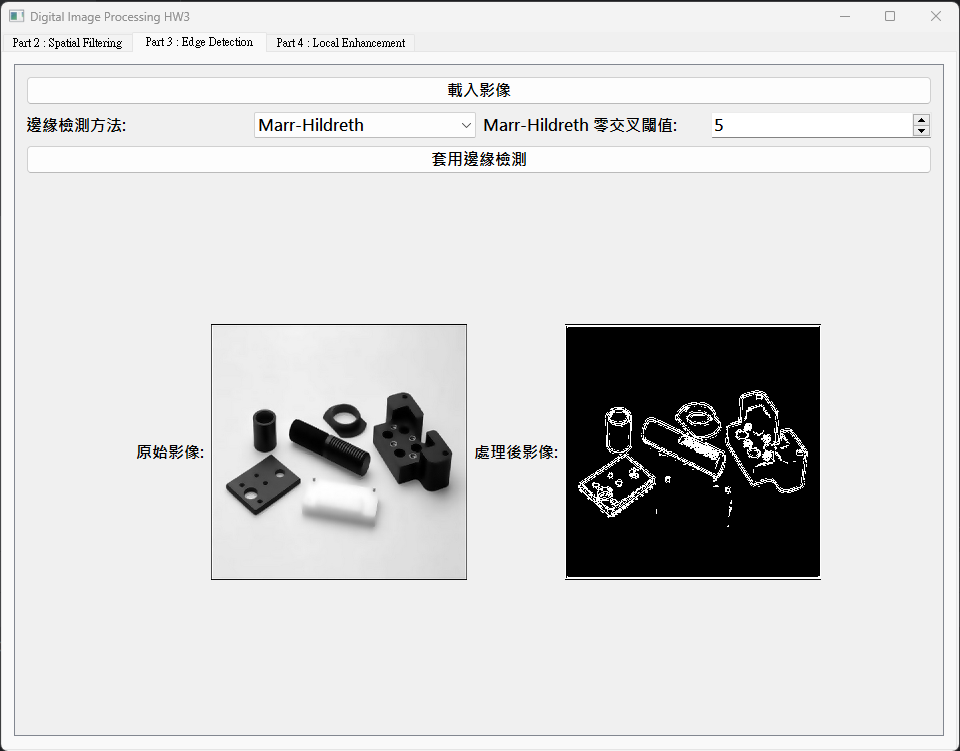
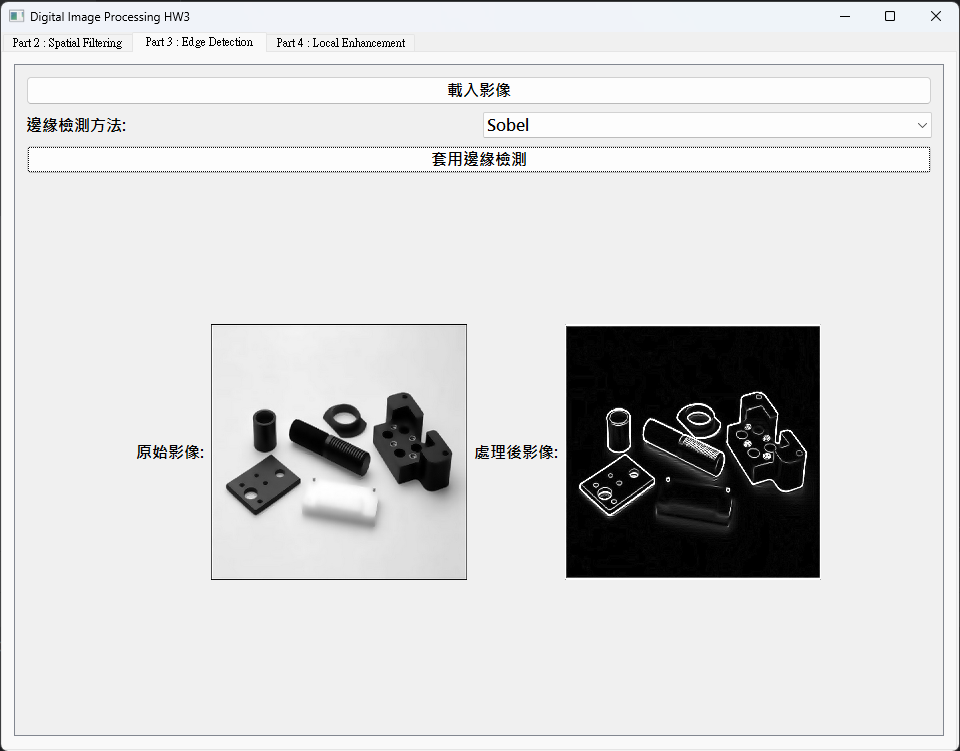
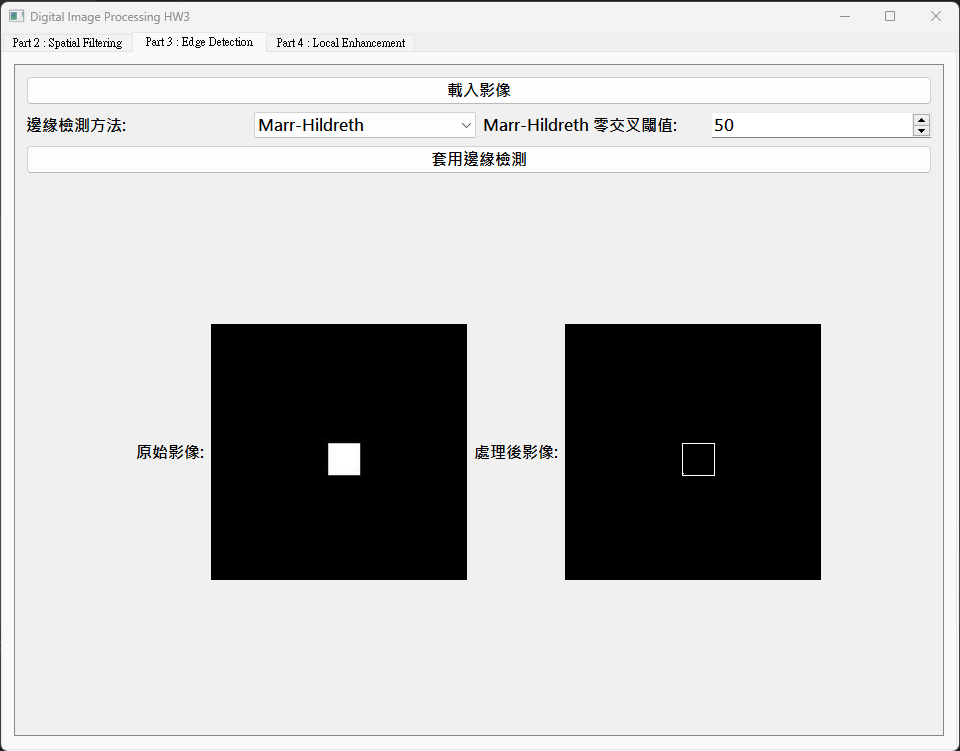
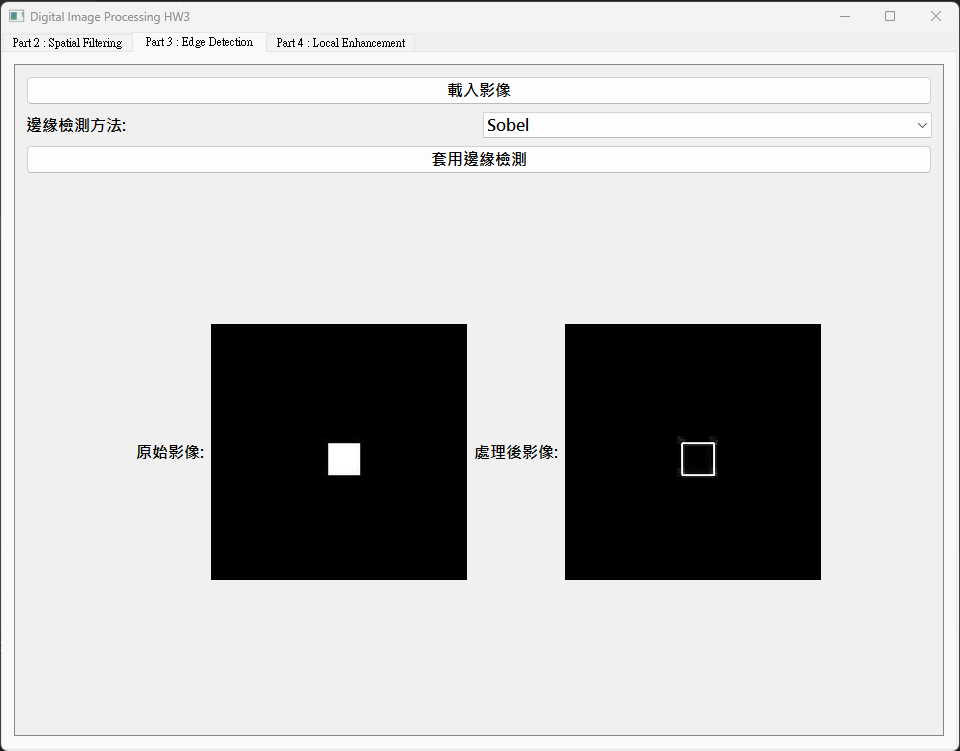
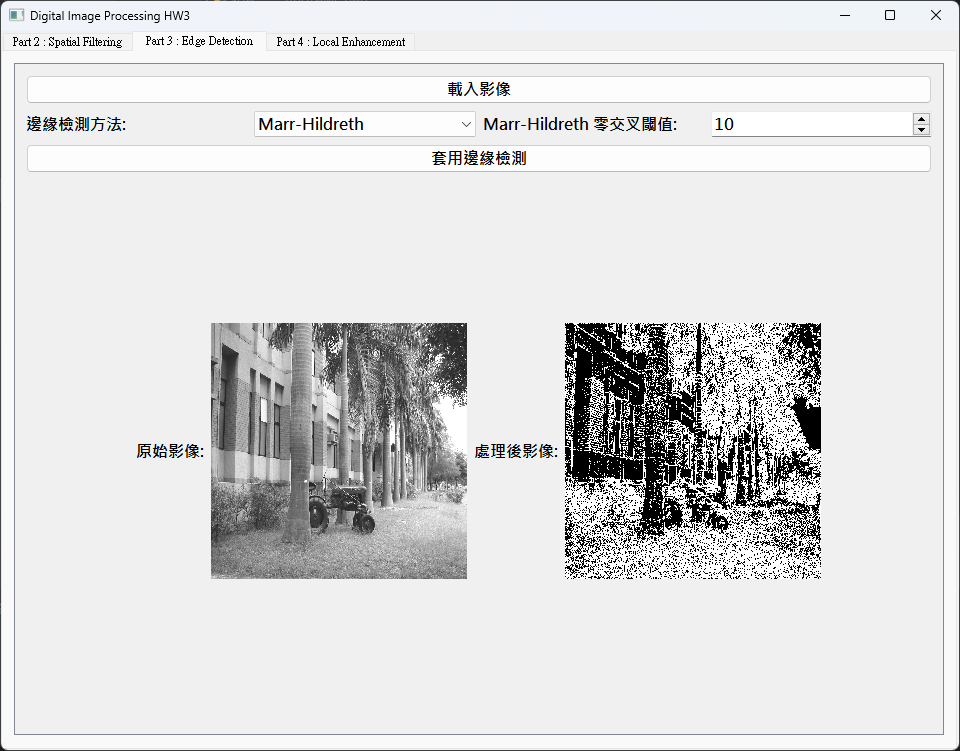
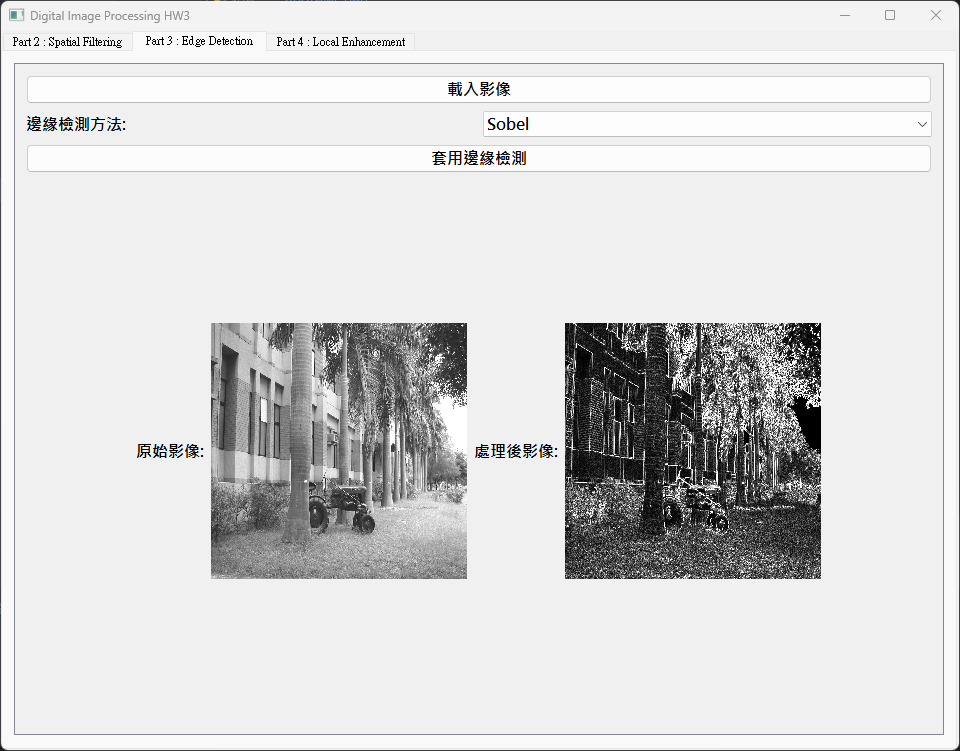
- 小遮罩大小（例如3x3）：這些遮罩產生較少的平滑效果，保留圖像的大部分細節，但去噪效果較弱。計算時間較快，因為每次卷積操作涉及的像素較少。  
- 大遮罩大小（例如21x21）：較大的遮罩引入更多的平滑效果，有助於去除更多噪聲並使圖像更加模糊。然而，較大的遮罩會增加計算負荷，導致更長的處理時間，因為每個像素的計算時間增加。

## Part 3：邊緣檢測方法 class EdgeDetectionApp()

主要功能:  
- Marr-Hildreth：使用高斯拉普拉斯 (LoG) 進行邊緣檢測，首先對圖像進行平滑處理，然後檢測零交叉點。  
- Sobel：使用 Sobel 核計算圖像的梯度來檢測邊緣。

工作原理:  
1. 用戶加載一張圖像。  
2. 選擇邊緣檢測方法（Marr-Hildreth 或 Sobel）。  
3. 對於 Marr-Hildreth，用戶可以設置零交叉閾值來檢測邊緣。  
4. 顯示結果。

程式執行截圖說明:

1. Marr-Hildreth threshold=5
2. Marr-Hildreth threshold=15 (此圖不錯的值)
3. Sobel 
4. Marr-Hildreth threshold=5 (此圖不錯的值，但圖中白色的物體消失) 
5. Sobel 
6. Marr-Hildreth threshold=50 (此圖不錯的值) 
7. Sobel
8. Marr-Hildreth threshold=10 (此圖不錯的值) 
9. Sobel 

程式細節:  
- marr\_hildreth\_edge\_detection 函數使用高斯平滑後的拉普拉斯核來檢測邊緣。  
- sobel\_edge\_detection 函數使用 Sobel 核在 x 和 y 方向上計算梯度大小。

### 討論：零交叉閾值對 Marr-Hildreth 邊緣檢測的影響:

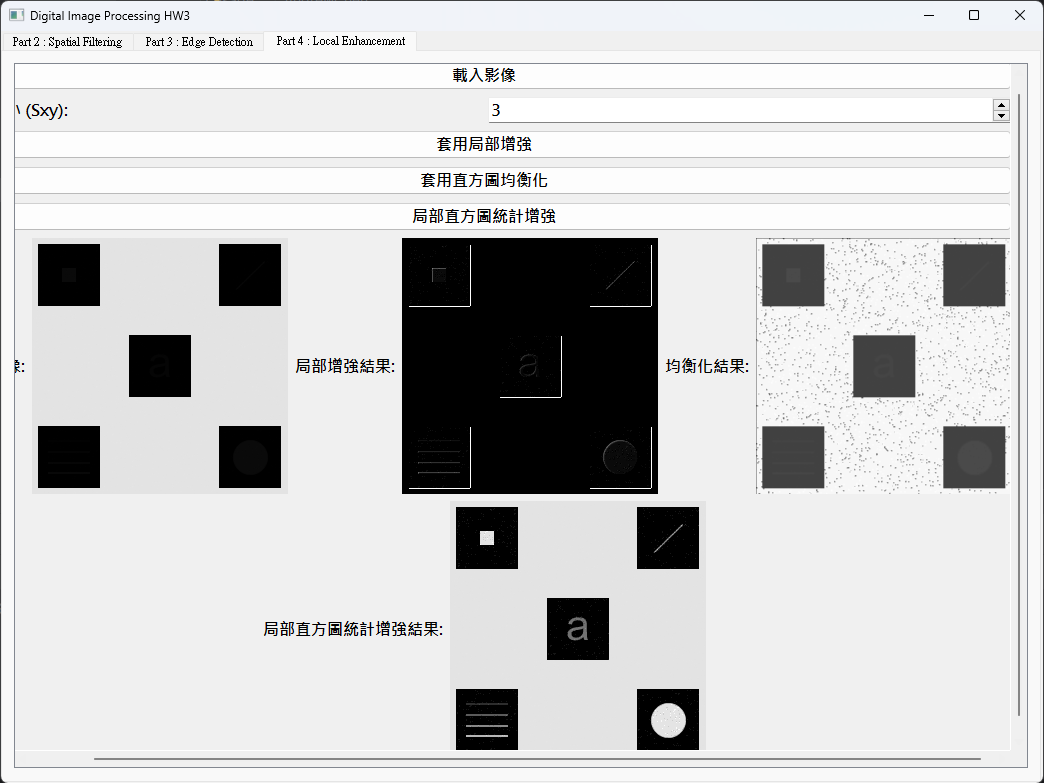
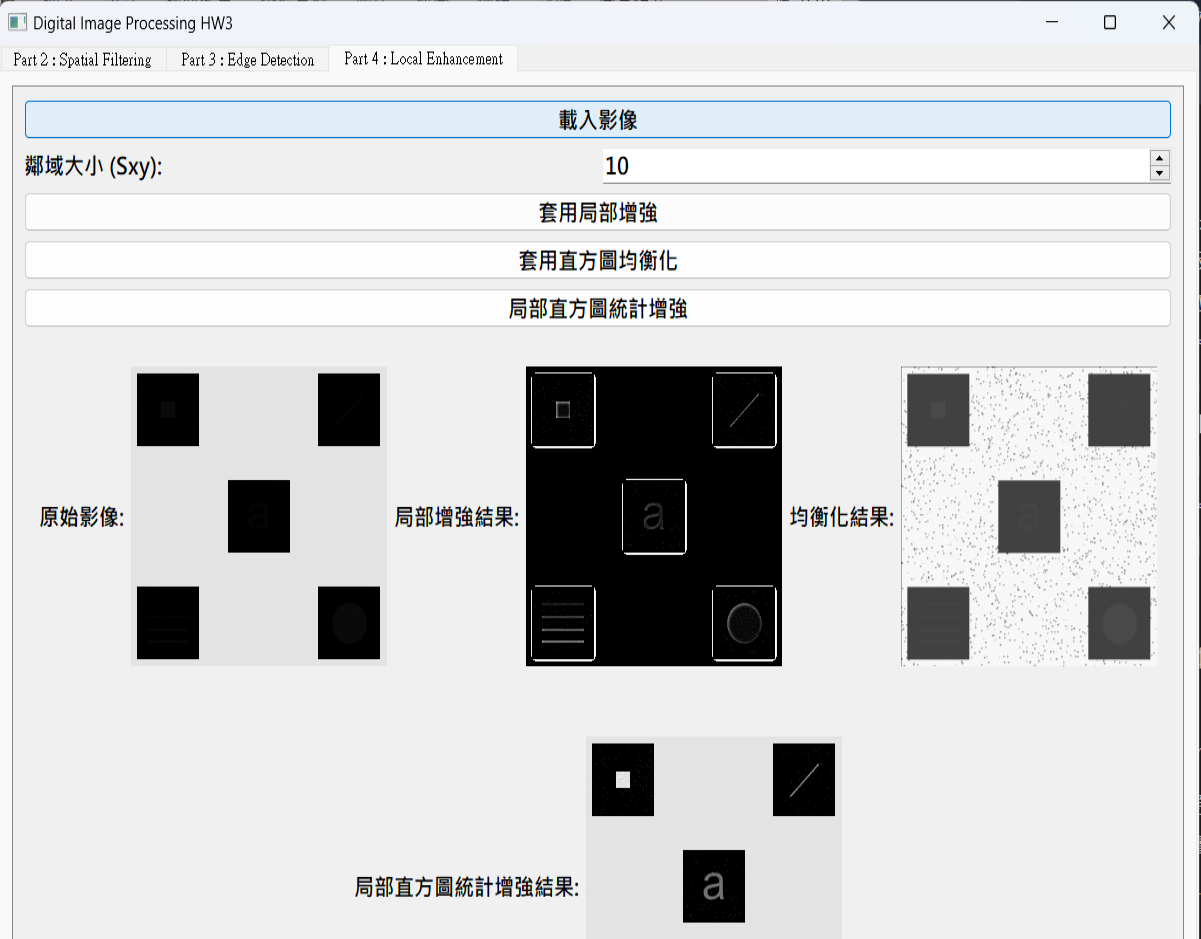
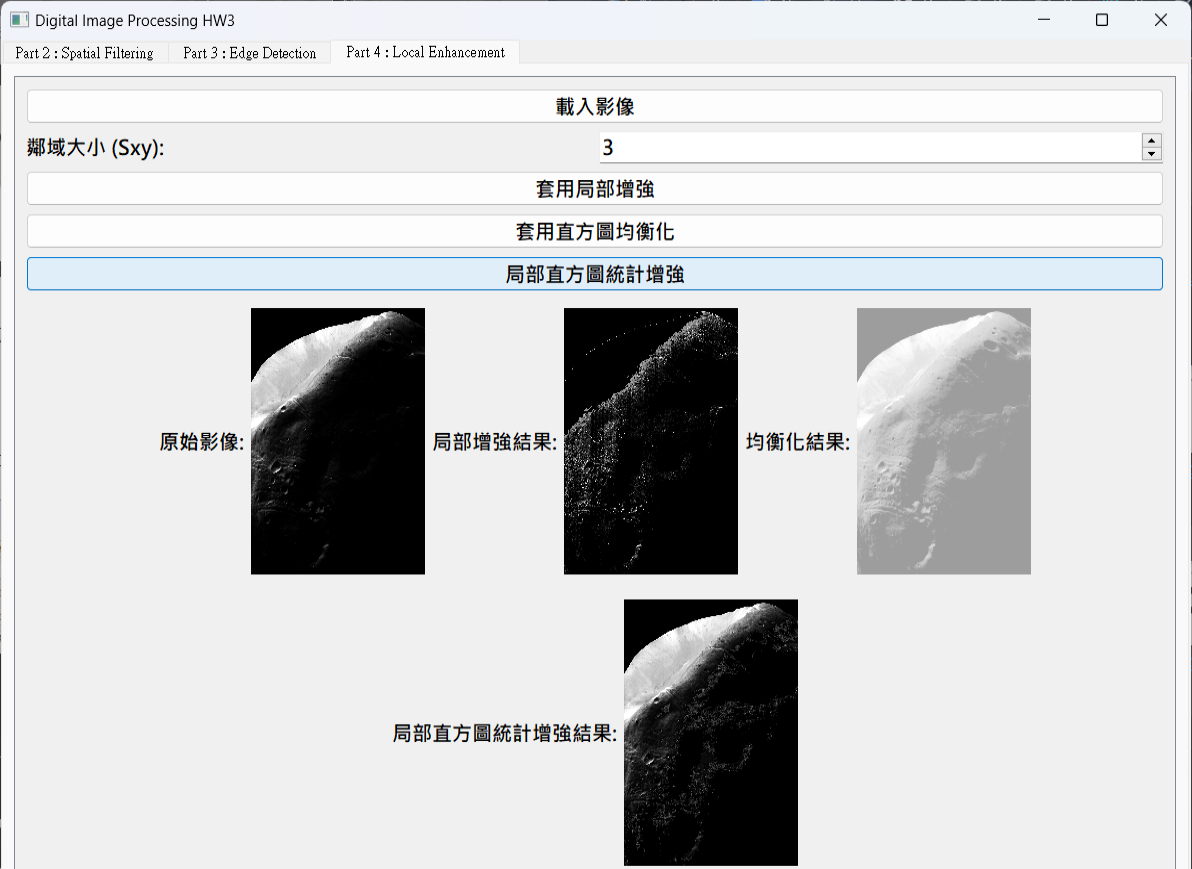
- 低閾值（例如1）：低閾值意味著即使拉普拉斯值的微小變化也會被視為邊緣。這可能會檢測到許多邊緣，包括弱邊緣和噪聲。  
- 高閾值（例如100）：較高的閾值通過忽略小變化來減少檢測到的邊緣數量，僅關注較強的邊緣。這會導致更乾淨的輸出，但可能會錯過圖像中的細微邊緣。

## Part 4：局部增強方法 class LocalEnhancementApp()

主要功能:  
- 局部增強：通過增強低對比度區域來調整圖像的對比度，同時保持圖像的整體亮度。  
- 直方圖均衡化：應用手動直方圖均衡化來全局增強圖像對比度。  
- 局部直方圖統計增強：使用局部統計（均值和標準差）來增強特定區域的對比度。

工作原理:  
1. 用戶加載一張圖像。  
2. 選擇鄰域區域大小 Sxy 並應用局部增強。  
3. 顯示結果，用戶還可以應用直方圖均衡化或局部直方圖統計增強。

程式執行截圖說明:

1. 使用提供的原始圖像（image 4-1.jpg），重現圖3.27b。Sxy=3
2. 使用提供的原始圖像（image 4-1.jpg），重現圖3.27b。Sxy=10 
3. 使用image 4-2.jpg

程式細節:  
- local\_enhancement 函數通過基於鄰域的均值和標準差調整每個像素的亮度。  
- manual\_histogram\_equalization 函數計算圖像直方圖的累積分佈函數 (CDF)，並使用它來重新映射像素值以進行對比度增強。  
- local\_histogram\_statistics 函數通過使用局部統計（如均值和標準差）增強暗區域。

### 討論：鄰域區域大小 (Sxy) 對局部增強圖像處理結果的影響:

- 小區域大小（例如 Sxy = 3）：這會導致在小區域內的局部對比度增強，僅增強小特徵或細節。它保留了圖像的清晰度，但可能無法處理較大區域。  
- 大區域大小（例如 Sxy = 20）：較大的區域會考慮更多像素來計算局部統計，從而導致更全局的對比度變化。這可以在整個圖像中產生更一般的增強效果，但細節可能會減少。

# 總結:

1. Part 2（空間濾波）：專注於圖像的平滑和過濾操作，具有可自定義的遮罩。遮罩的大小直接影響圖像的外觀和計算時間。  
2. Part 3（邊緣檢測）：提供兩種邊緣檢測方法，Marr-Hildreth 對零交叉閾值很敏感。較低的閾值檢測更多的邊緣，而較高的閾值則集中於強邊緣。  
3. Part 4（局部增強）：使用基於局部統計和直方圖的方法來增強圖像對比度。鄰域區域大小 (Sxy) 影響局部化或全局化的增強效果，較小的區域強調細節，較大的區域影響更廣泛的區域。