**國立臺南大學**

**資訊工程學系**

**影像處理**

**Final Project**

**資工四 S11059003曾俊杰**

**西元2025年6月16日**

**目錄**

[**一、** **文件簡介** 5](#_Toc200931004)

[**二、** **頁面(Panels)** 6](#_Toc200931005)

[1. Figure標題 6](#_Toc200931006)

[2. 初始頁面(Main Panel) 6](#_Toc200931007)

[3. 文件頁面(Document Panel) 7](#_Toc200931008)

[4. 注意事項 7](#_Toc200931009)

[**三、** **初始頁面** 8](#_Toc200931010)

[1. 按鈕 – Upload 8](#_Toc200931011)

[2. 按鈕 – Process 10](#_Toc200931012)

[3. 按鈕 – Document 10](#_Toc200931013)

[4. Axe圖片顯示區塊 11](#_Toc200931014)

[5. 注意事項 11](#_Toc200931015)

[**四、** **文件頁面** 11](#_Toc200931016)

[1. 按鈕 – Back 11](#_Toc200931017)

[2. 文字顯示區塊 11](#_Toc200931018)

[3. 注意事項 12](#_Toc200931019)

[**五、** **圖像特性分析模塊** 13](#_Toc200931020)

[1. 屬性識別 13](#_Toc200931021)

[2. 分析圖片對比度 13](#_Toc200931022)

[3. 分析圖片清晰度 13](#_Toc200931023)

[4. 注意事項 14](#_Toc200931024)

[**六、** **問題診斷** 14](#_Toc200931025)

[1. 直方圖分析 14](#_Toc200931026)

[2. 躁聲診斷 15](#_Toc200931027)

[3. 模糊診斷 16](#_Toc200931028)

[4. 光線診斷 17](#_Toc200931029)

[5. 色彩診斷 18](#_Toc200931030)

[5. 注意事項 19](#_Toc200931031)

[**七、** **處理說明** 19](#_Toc200931032)

[1. 判斷背景 20](#_Toc200931033)

[2. 高斯躁聲 20](#_Toc200931034)

[3. 松柏躁聲 + 斑點躁聲 + Localvar躁聲 20](#_Toc200931035)

[4. 椒鹽躁聲 21](#_Toc200931036)

[5. Laplacian-based模糊 21](#_Toc200931037)

[6. 邊緣模糊 22](#_Toc200931038)

[7. 運動模糊 22](#_Toc200931039)

[8. 過度曝光 22](#_Toc200931040)

[9. 曝光不足 22](#_Toc200931041)

[10. 對比度不足 23](#_Toc200931042)

[11. 光照不均 23](#_Toc200931043)

[12. 色彩單調 23](#_Toc200931044)

[13. 色偏 23](#_Toc200931045)

[14. 色彩飽和度過低 24](#_Toc200931046)

[15. 色彩飽和度過高 24](#_Toc200931047)

[16. 注意事項 24](#_Toc200931048)

[**八、** **專案自製函式** 25](#_Toc200931049)

[**九、** **程式碼撰寫規則** 26](#_Toc200931050)

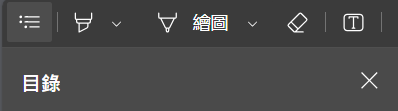
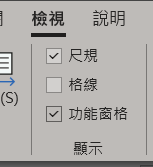
[**十、** **測試結果(請注意第(7)點)** 26](#_Toc200931051)

[**十一、** **參考文獻** 36](#_Toc200931052)

1. **文件簡介**

本文件將介紹專案的所有功能，共包含1個頁面，前情提要文件的撰寫規則。

1. 專案目標：製作一個修復古文獻APP
2. 輸入：有各種問題的圖片
3. 輸出：修復後的圖片
4. 限制條件：
5. 輸出和輸入數量、檔名固定
6. 只能使用課堂講過的相關功能(函式方法)
7. 成品簡介：描述輸入圖片特性，共有5種躁聲相關、3種模糊相關、4種光線相關和4種色彩相關問題診斷/解決方法
8. 注意事項：
9. 礙於Affine Transformations的判斷過於繁瑣，在輸入條件未知的情況下難以撰寫程式，且對既有輸入造成負面影響，所以本專案假設輸入無須使用Affine Transformations即可處理
10. 本文件共有4個階層，可以打開WORD/PDF功能窗格進行快速導覽，第1階層格式為：中文數字、標題；第2階層格式為：阿拉伯數字.標題；第3層格式為：(阿拉伯數字)標題；第4層格式為：羅馬數字.標題；第5層格式為：(羅馬數字)標題，僅將前2階層列入導覽書籤



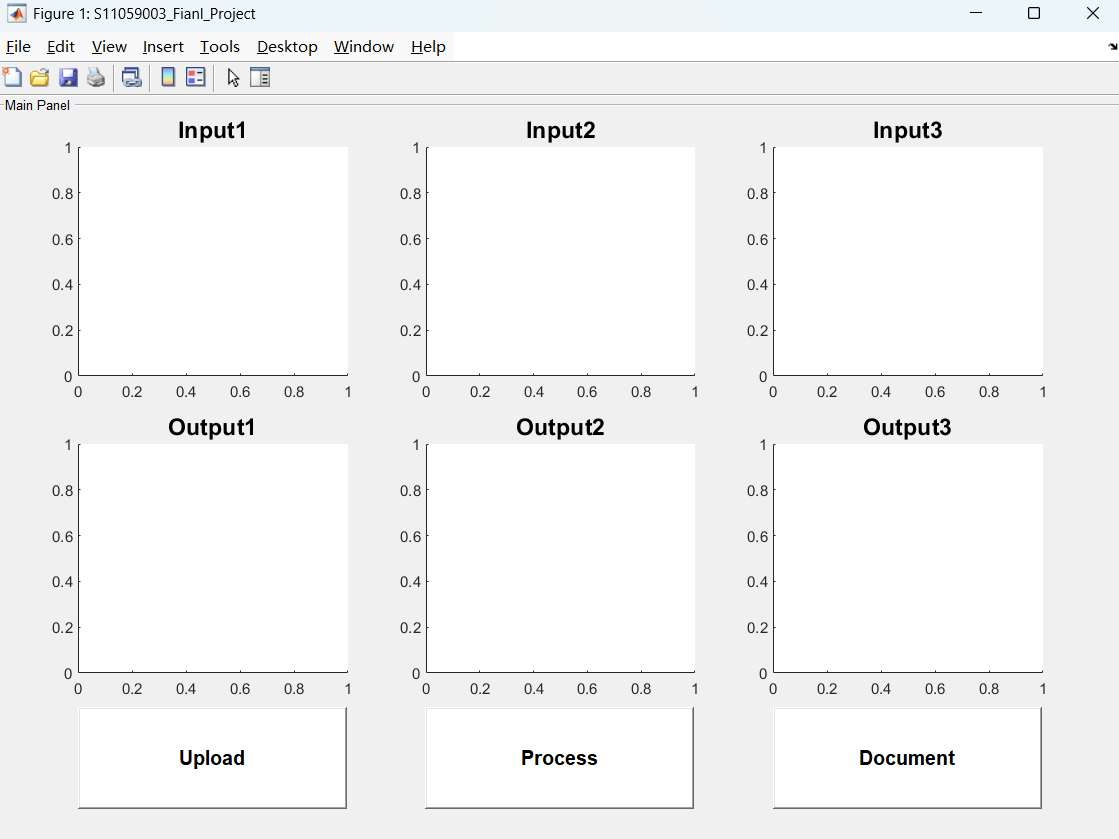
1. 本專案所有內容物皆為相對位置，並且有一定比例的大小，隨著Figure正常拉伸可以調整自己的位置，不會留在原地
2. **頁面(Panels)**

介紹專案中，每個頁面的畫面與內容物，在頁面左上角會標示當前所在的Panel名稱。

1. **Figure標題**

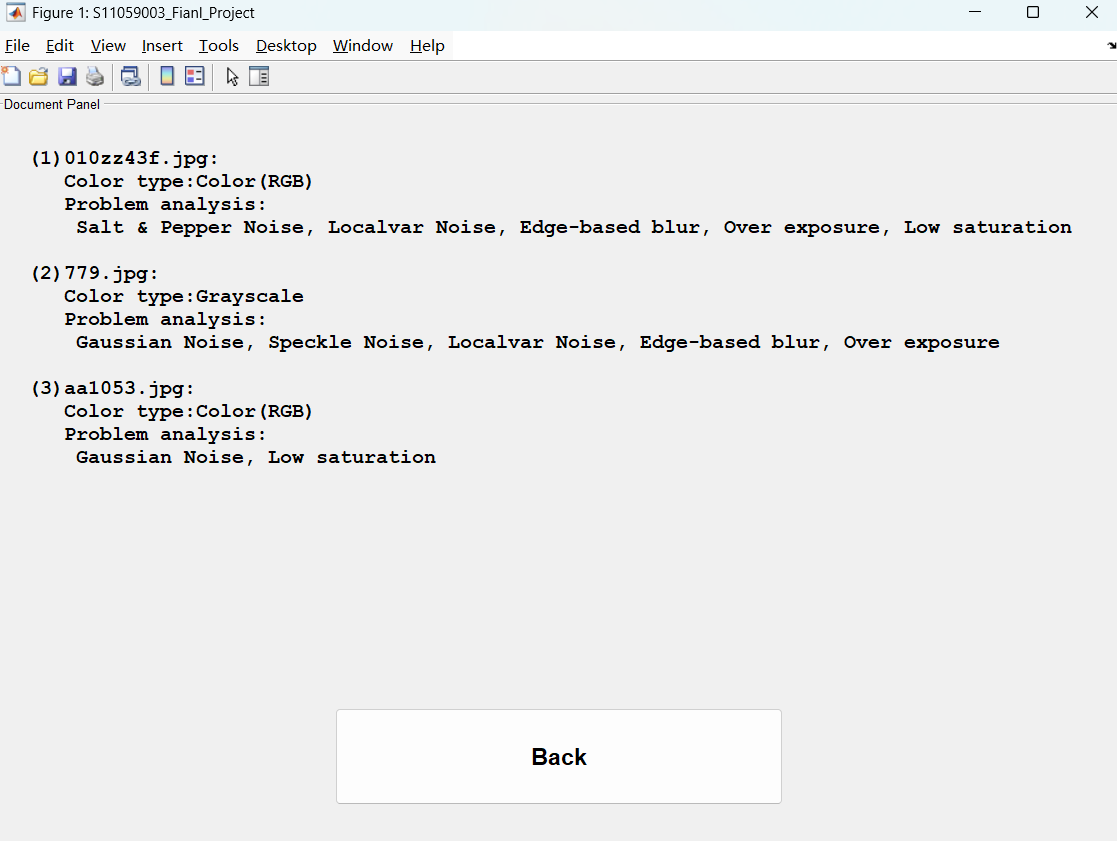


1. **初始頁面(Main Panel)**
2. 進入方式：執行.m程式或在文件頁面按下Back按鈕
3. 內容：3個按鈕、6個Axe顯示圖片區塊





1. **文件頁面(Document Panel)**
2. 進入方式：按下Document按鈕
3. 內容：1個按鈕、1個文字區塊





1. **注意事項**
2. 本專案不需要使用MATLAB Figure內建的功能，請勿觸碰以免造成程式崩潰。



1. 本專案沒有「關閉程式」按鈕，敬請使用Figure內建的關閉按鈕(右上角)

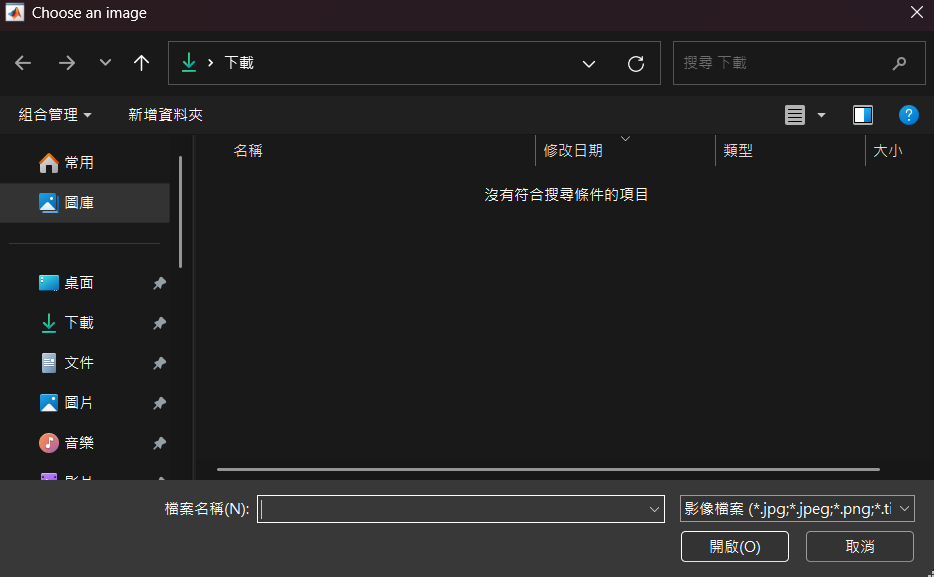


1. **初始頁面**

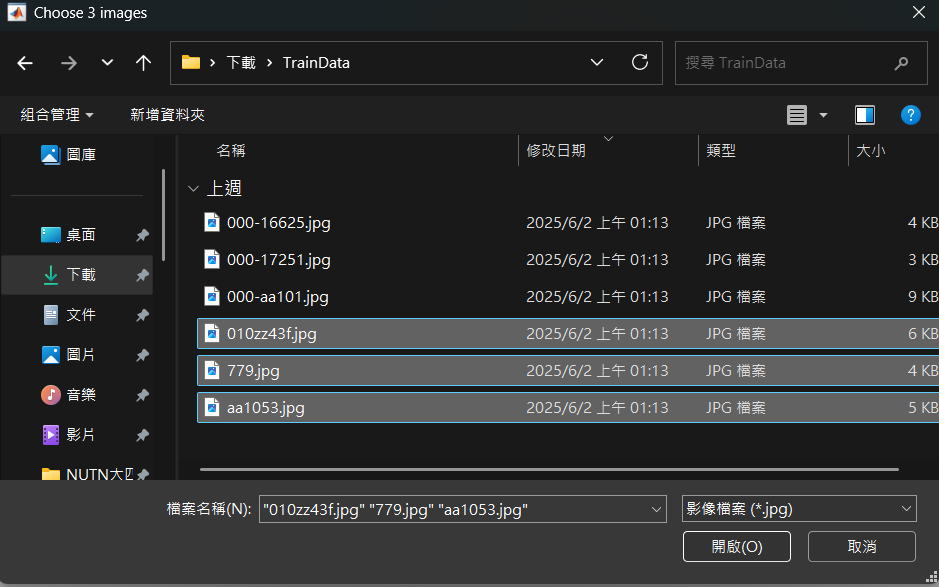
詳細介紹專案中，初始頁面的功能。

1. **按鈕 – Upload**

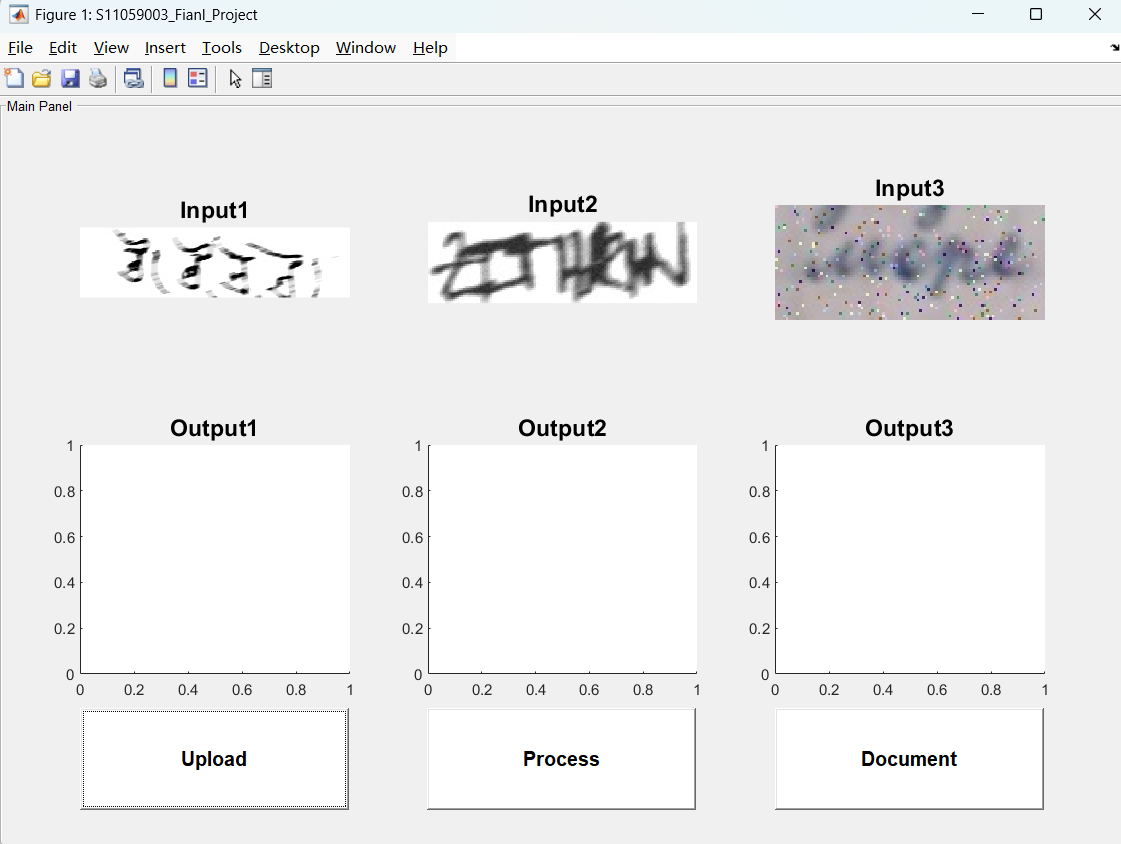
上傳輸入圖片，按下後將彈出1個文件選擇框，可在本地電腦中選擇任意3張圖片上傳，圖片副檔名限制為.jpg/.png圖片，由於程式內規範只搜尋指定副檔名的圖片，所以除此以外的檔案不會出現。



選擇指定3張圖片後，按下開啟。

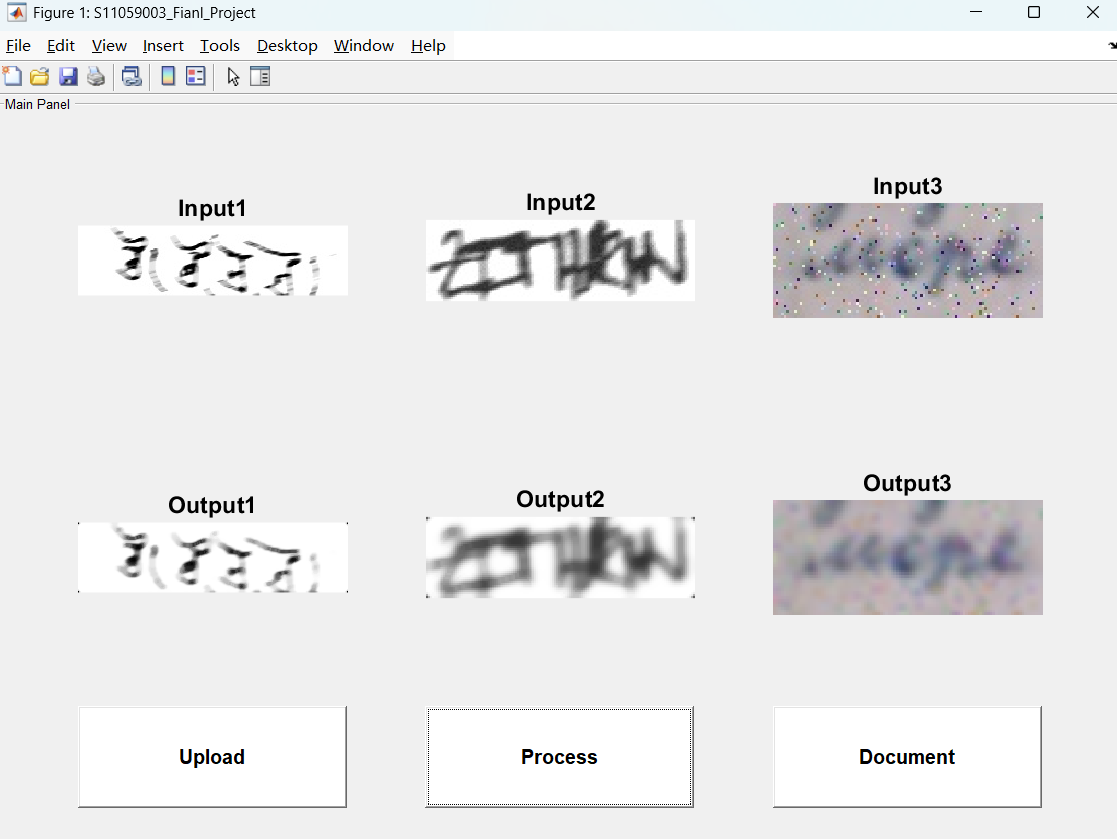


文件選擇框會自動關閉，同時，儲存輸入圖片，當作原始圖片，將其顯示在Input1, Input2, Input3圖片顯示區塊。



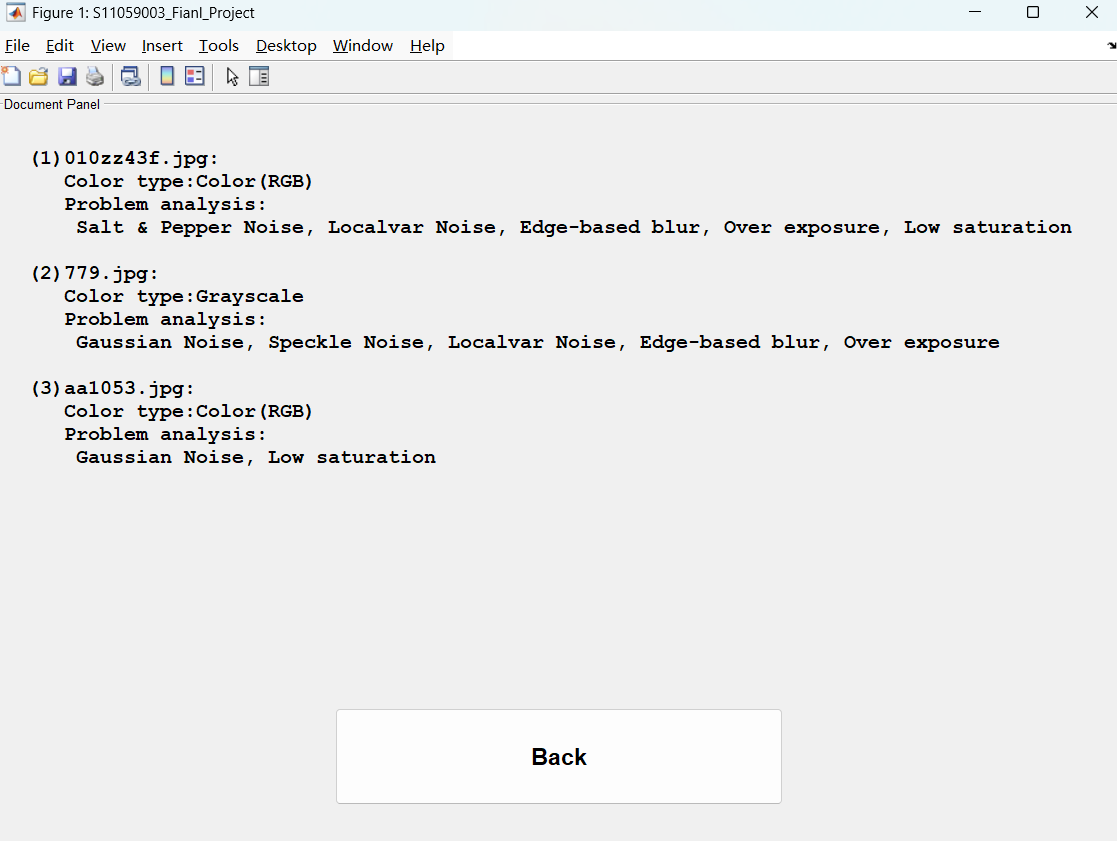
1. **按鈕 – Process**

按下後，即可開始修復輸入圖片(此畫面只是示意圖，不代表最終結果)，將其顯示在Output1, Output2, Output3圖片顯示區塊。



1. **按鈕 – Document**

按下後，即可進入文件頁面



1. **Axe圖片顯示區塊**
2. Input1, Input2, Input3：儲存Upload Image所選的圖片(一次3張)
3. Output1, Output2, Output3：儲存經過Process功能後的圖片(一次3張)
4. **注意事項**
5. Upload Image允許重新上傳相同或不同的圖片，無論彩色或灰階，一旦上傳完成，不會自動清空當前Processed Image儲存的圖片(如果存在)
6. 如果上傳時，所選圖片少於3張，不會執行上傳動作(保持原有的Input1, Input2, Input3)
7. 如果在沒有上傳任何圖片前按下Process按鈕，則不會執行修復圖片
8. **文件頁面**

詳細介紹專案中，文件頁面的功能。

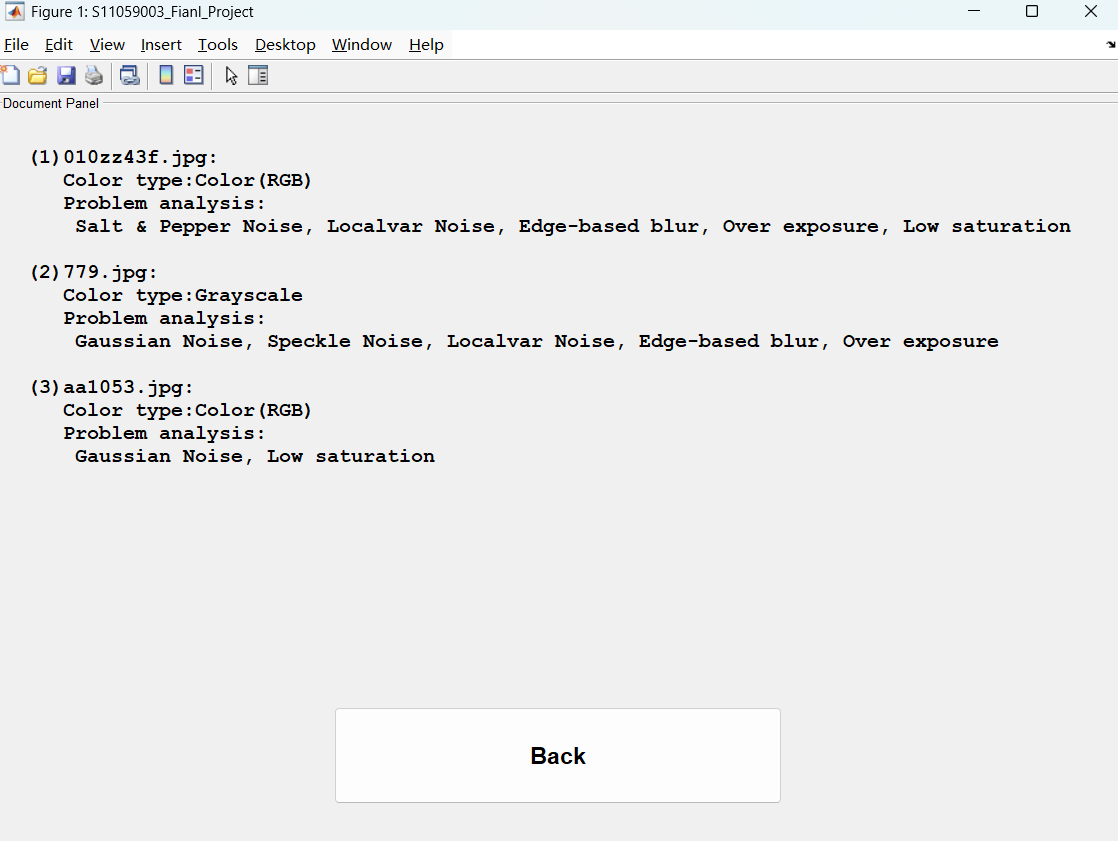
1. **按鈕 – Back**

按下後，即可返回初始頁面。

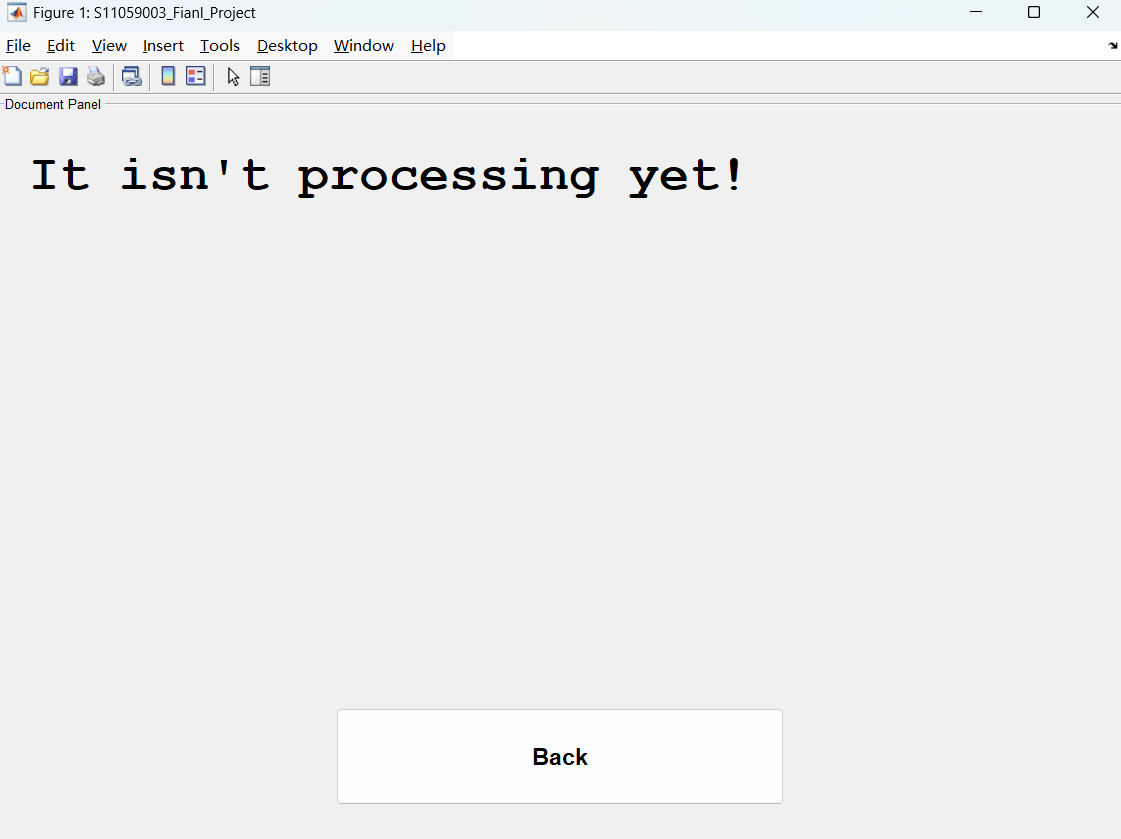
1. **文字顯示區塊**

按照順序顯示上傳三張圖片的

1. 圖片名稱
2. Color type：圖片類型是RGB或灰階，只有2種
3. Problem analysis：圖片問題名稱，包含5種躁聲、3種模糊、4種光線和4種色彩問題



1. **注意事項**
2. 如果在沒有完成Process按鈕之要求(修復完照片)前按下Document按鈕，文件頁面的文字區塊會顯示”It isn’t processing yet!”



1. 在文字顯示區塊Problem analysis中，若是問題很多，超出一行，可能導致排版有些微偏差，但並不會影響該功能(文字顯示)
2. **圖像特性分析模塊**

詳細介紹專案中，圖像特性分析模塊的功能，該部分負責分析每張輸入圖像的基本屬性與品質，包含以下功能：

1. **屬性識別**
2. 透過imfinfo()取得屬性
3. 圖像類型(彩色Color(RGB)或灰階Grayscale)：透過imfinfo中的ColorType屬性取得
4. 分辨率(Resolution)：讀取圖像的寬度與高度(寬x高)，透過imfinfo中的Width和Height屬性取得
5. 色彩深度(Bit Depth)：透過imfinfo中的BitDepth屬性取得
6. **分析圖片對比度**

透過Michelson Contrast對比度指標，分析圖像明暗程度差異：

1. Michelson Contrast：最大與最小亮度差 / 最大與最小亮度差和，常用於週期性圖案
2. 對比度計算後，會根據數值區間給出可解讀的等級說明，分為正常與過低，是為判定是否存在對比度(過低)問題的依據
3. **分析圖片清晰度**

透過3種清晰度指標，分析圖像細節程度差異：

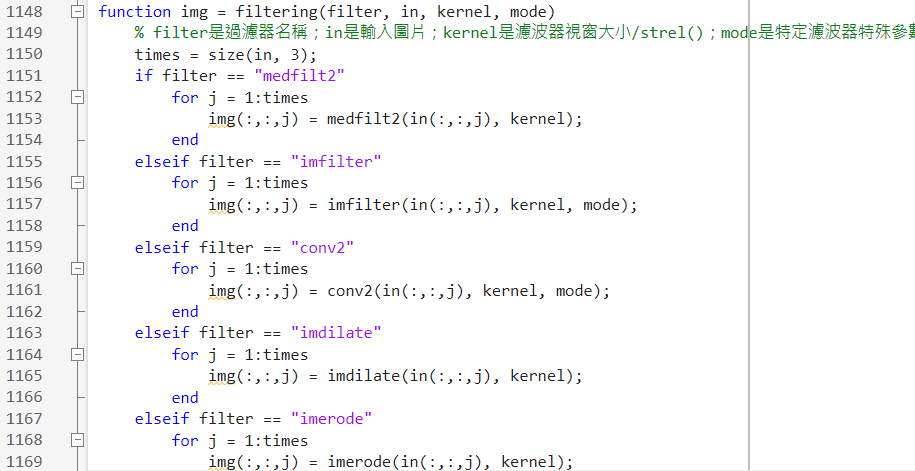
1. Laplacian Variance：使用拉普拉斯濾波器計算邊緣強度變異
2. Edge Detection：以Sobel邊緣檢測計算邊緣密度
3. 清晰度計算後，會根據數值區間給出可解讀的等級說明，分為正常與有問題，是為判定是否存在模糊問題的依據
4. **注意事項**
5. 圖像類型我參考過[該網站](https://onlinepngtools.com/check-if-png-is-grayscale)的解釋，當彩色圖像的三個通道內容都是相同的時候，在視覺上它應該被判定為灰階圖，但由於課堂上並沒有相關描述，所以我最終還是採用檢查通道數的方式判斷，通道數為3是彩色圖像，反之則不是
6. 由於課堂中與PDF說明中都沒有明確指出如何分析圖片清晰度和對比度，我上網自行找尋相關指標，但得到的答案也是模稜兩可，因此，我並不能確定這2種指標是正確的、可靠的
7. 該部分的文件(.txt)格式固定，可見於「測試結果」
8. [對比度指標來源(General)](https://ww2.mathworks.cn/matlabcentral/answers/381648-how-to-find-the-contrast-value-of-an-image)
9. [對比度指標來源(其他)](https://www.schorsch.com/en/kbase/glossary/contrast.html)
10. [清晰度指標來源](https://blog.winerva.com/2024/04/12/%E5%BD%B1%E5%83%8F%E6%B8%85%E6%99%B0%E5%BA%A6%E5%88%86%E6%9E%90/)
11. **問題診斷**

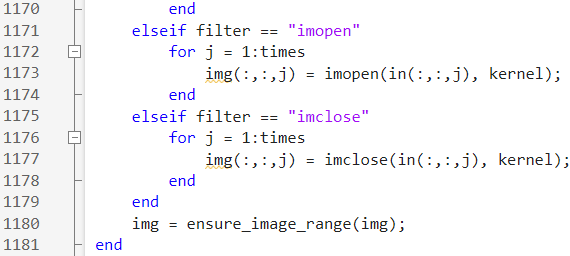
簡略介紹專案中，問題診斷的功能，該部分負責分析輸入圖像的品質缺陷，包括噪聲、模糊、曝光與色彩問題，使用統計與圖像處理方法進行自動化偵測，作為後續強化策略的依據，包含以下功能：

1. **直方圖分析**
2. 標準化亮度分布：統計每個灰階值(0~255)出現的頻率，可判斷影像是偏暗、偏亮、均衡、集中某段灰階
3. 總像素數：用來支援某些比例型判斷(如極亮極暗佔比)
4. 平均亮度(Mean Intensity)：反映整體影像偏暗或偏亮
5. 亮度標準差(Std Intensity)：用於判斷影像對比度高低：越小表示影像灰階範圍窄(低對比)
6. **躁聲診斷**
7. Gaussian Noise
   1. 診斷方法：利用stdfilt計算局部標準差，若整體亮度變異夠大，且局部變異均勻(方差/均值比小於1.0)且大於某閾值，即判斷為高斯噪聲
   2. 理由：高斯噪聲具有零均值、恆定方差特性，導致圖像各區域局部變異高且一致
8. Poisson Noise
9. 診斷方法：利用灰階值與像素平均關係判定光照是否偏暗，對暗區與亮區分別計算方差與均值比(變異/平均)，若兩者比值接近，視為Poisson噪聲
10. 理由：Poisson噪聲為訊號依賴型，方差與均值成正比，在光線強弱區應表現一致
11. Salt & Pepper Noise
12. 診斷方法：計算極暗與極亮像素的比例(小於5、大於250)，使用中值濾波與原圖比較，若高差異點比例合理(避免誤判高斯噪聲或圖像邊緣)，則判定為椒鹽噪聲
13. 理由：椒鹽噪聲的特徵為孤立的極值點，易在中值濾波後表現為顯著變化
14. Speckle Noise
15. 診斷方法：以中亮與高亮區為分析對象，計算變異與亮度平方比(變異正規化)，若兩區比值相近，且達一定門檻，表示為Speckle噪聲
16. 理由：Speckle噪聲為乘法性噪聲，其變異與亮度平方呈線性關係
17. Localvar Noise
18. 診斷方法：分段圖像亮度區間(4級)，計算每區的局部變異，若不同亮度區的局部變異變化比例明顯(比值 > 1.5 且 < 5)，代表噪聲強度隨亮度變化
19. 理由：Localvar為變異依據信號變化之噪聲，表現在不同亮度區的局部標準差也有顯著差異
20. 檢測結果儲存：在detected\_noises中存入躁聲名稱
21. 'Gaussian Noise'
22. 'Poisson Noise'
23. 'Salt & Pepper Noise'
24. 'Speckle Noise'
25. 'Localvar Noise'
26. **模糊診斷**
27. Laplacian Variance
28. 診斷方法：應用Laplacian運算子後取變異值，反映邊緣細節程度，若變異小於閾值(200)，表示圖像細節少，為模糊
29. 理由：Laplacian 對邊緣敏感，模糊圖像回應值低
30. 邊緣密度
31. 診斷方法：使用Sobel邊緣檢測，計算邊緣像素密度，若邊緣密度過低(<0.05)，視為模糊
32. 理由：模糊圖像邊緣過渡平滑，可檢測邊緣變少
33. 運動模糊
34. 診斷方法：使用水平與垂直Sobel，計算兩向邊緣數量比值，若邊緣密度不平衡(>1.5 或 <0.67)，可能存在方向性模糊
35. 理由：Localvar為變異依據信號變化之噪聲，表現在不同亮度區的局部標準差也有顯著差異
36. 檢測結果儲存：在detected\_blur中存入模糊名稱
37. 'Laplacian-based blur'
38. 'Edge-based blur'
39. 'Motion blur'
40. **光線診斷**
41. 過度曝光
42. 診斷方法：計算直方圖中高亮度像素的比例(像素值 > 230)，搭配圖像平均亮度進行判斷
43. 理由：過度曝光的圖像會出現大量亮部區域，造成像素值集中在直方圖右側，並導致平均亮度偏高
44. 曝光不足
45. 診斷方法：計算直方圖中低亮度像素的比例(像素值 < 25 )，搭配圖像平均亮度進行判斷
46. 理由：曝光不足會導致圖像普遍過暗，像素值集中在直方圖左側，平均亮度偏低
47. 對比度不足
48. 診斷方法：使用Michelson對比度進行判斷(定義為 (max - min) / (max + min))。若低於0.4即視為對比度不足
49. 理由：對比度低會使圖像細節不明顯，灰階值集中在中間區域，畫面缺乏明暗層次
50. 光照不均
51. 診斷方法：將灰階圖像分成多個區塊，計算各區塊平均亮度之間的變異(方差、最大最小差、相鄰差異)
52. 理由：當圖像不同區域亮度差異大(例如亮的一邊、暗的一邊)，代表光照分布不均，會影響視覺一致性與辨識準確度
53. 檢測結果儲存：在lighting\_issues中存入光線問題名稱
54. 'Over exposure'
55. 'Under exposure'
56. 'Low contrast'
57. 'Non-uniform lighting'
58. **色彩診斷**
59. 色彩單調
60. 診斷方法：分析RGB三通道的標準差，若三者皆小於25且平均小於18，判定為色彩單調
61. 理由：標準差小表示顏色變化少，畫面整體缺乏色彩層次與活力，通常看起來偏灰或偏單色
62. 色偏：比較 R、G、B 三個通道的平均值差異與比值，任兩通道平均值差距 > 15 且其中一通道佔比 > 1.15，視為色偏
63. 診斷方法：比較RGB三通道平均值，若最大通道與其他差異 > 15 且通道比例 > 1.15，則視為色偏
64. 理由：正常圖像應呈現色彩平衡，若某一通道過強，會導致整體圖像偏紅、偏藍或偏綠等現象
65. 色彩飽和度過高/過低
66. 診斷方法：計算飽和度近似值(max(R,G,B) - min(R,G,B)) / (max(R,G,B) + eps)，並分析其平均值與標準差
67. 平均 < 0.2：判定為低飽和度
68. 平均 > 0.75且標準差 > 0.2：判定為高飽和度
69. 理由：飽和度代表色彩的純度，過低使畫面顏色灰濛、缺乏生氣；過高則可能導致顏色不自然、刺眼
70. 檢測結果儲存：在detected\_color中存入色彩問題名稱
71. 'Low color variation'
72. '{channel\_name} color cast'，channel\_name = Red/Blue/Green
73. 'Low saturation'
74. 'High saturation'
75. **注意事項**
76. 該部分已經盡量說明使用到的公式與方法，但是實際上我並不了解這些方法，可能有大量疏漏，建議可直接查看程式碼
77. 該部分的文件(.txt)格式不固定(依據圖片存在的問題)，可見於「測試結果」
78. **處理說明**

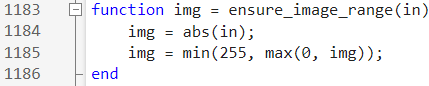
簡略介紹專案中，參數頁面的功能，該部分負責處理輸入圖像的品質缺陷，包括噪聲、模糊、曝光與色彩問題，使用濾波器與圖像處理方法進行缺陷排除(removal)或降低(reduction)問題所造成的負面影響，根據程式碼順序撰寫，以部分函式簡述使用了什麼處理，包含以下功能：

1. **判斷背景**
2. 問題簡述：針對白底黑字與黑底白字的特別判斷
3. 處理方法
4. is\_white\_background：判斷背景是否是白色，值為0或1
5. text\_area\_ratio：計算文字區域占圖像的比例，判斷圖像中文字的密度，決定處理策略，
6. 以上2點會決定以下處理方法的方法或參數有所分歧，由於邏輯非常複雜，建議直接對照程式碼
7. **高斯躁聲**
8. 問題簡述：當檢測到高斯躁聲
9. 處理方法：Gaussian Filter
10. 部分函式：imfilter(A, fspecial('gaussian', [kernel kernel], sigma), 'replicate')
11. kernel = 3
12. sigma = 0.2/0.15/1.2
13. **松柏躁聲 + 斑點躁聲 + Localvar躁聲**
14. 問題簡述：
15. Speckle Noise：當檢測到Speckle Noise
16. Poisson Noise：當檢測到Poisson Noise
17. Localvar Noise：當檢測到Localvar Noise
18. 處理方法：Gaussian Filter/Median filter
19. 部分函式：
20. medfilt2(A, [kernel kernel])
21. kernel = 3
22. imfilter(B, fspecial('gaussian', [kernel2 kernel2], sigma), 'replicate')
23. kernel2 = 5
24. sigma = 0.1
25. **椒鹽躁聲**
26. 問題簡述：當檢測到椒鹽躁聲
27. 處理方法：Morphological Operations
28. 部分函式：imopen(A, se), imclose(A, se), imerode(A, se)
29. se = strel('disk', 1)
30. **Laplacian-based模糊**
31. 問題簡述：檢測到Laplacian-based模糊
32. 處理方法：Morphological Closing
33. 部分函式：imclose(A, se) \* 2
34. se\_h = strel('line', 2, 0)
35. se\_v = strel('line', 2, 90)R = conv2(A, kernel, 'same')
36. **邊緣模糊**
37. 問題簡述：當檢測到Edge-based模糊
38. 處理方法：Edge Enhancement
39. 部分函式：imclose(A, se), imopen(A, se), imdilate(A, se)
40. se = strel('disk', 1)
41. **運動模糊**
42. 問題簡述：當檢測到Motion Blur
43. 處理方法：Directional Morphological Operations
44. 部分函式：imclose(A, se), imopen(A, se)
45. se = strel('line', 2, 0)或strel('line', 2, 90)
46. **過度曝光**
47. 問題簡述：當檢測到過度曝光
48. 處理方法：Pixel Value Control
49. 部分函式：像素值限制控制
50. **曝光不足**
51. 問題簡述：當檢測到曝光不足(嚴重/中度/輕微)
52. 處理方法：Pixel Value Enhancement
53. 部分函式：像素值提升控制
54. **對比度不足**
55. 問題簡述：當檢測到對比度不足
56. 處理方法：Morphological Gradient
57. 部分函式：imdilate(A, se), imerode(A, se)
58. se = trel('disk', 1)/strel('disk', 2)
59. **光照不均**
60. 問題簡述：當檢測到光照不均
61. 處理方法：Background Estimation Correction
62. 部分函式：imopen(A, se), imclose(A, se)
63. se = strel('disk', 5)
64. **色彩單調**
65. 問題簡述：當檢測到色彩單調
66. 處理方法：Morphological Color Enhancement
67. 部分函式：imdilate(A, se), imerode(A, se)
68. se = trel('disk', 1)/strel('disk', 2)
69. **色偏**
70. 問題簡述：當檢測到色偏
71. 處理方法：Color Balance Correction
72. 部分函式：imopen(A, se)
73. se = strel('disk', 3)
74. **色彩飽和度過低**
75. 問題簡述：當檢測到飽和度過低
76. 處理方法：Chroma Enhancement
77. 部分函式：色度增強計算
78. **色彩飽和度過高**
79. 問題簡述：當檢測到飽和度過高
80. 處理方法：Chroma Control
81. 部分函式：色度控制計算
82. **注意事項**
83. 該部分已經盡量說明使用到的公式與方法，但是實際上我並不完全了解這些方法，可能有大量疏漏，建議可直接查看程式碼
84. 該部分的文件格式不固定(依據圖片存在的問題)，可見於「測試結果」，尤其是想要測試特定的處理，方法就寫在裡面，助教可前去查看，以節省時間
85. 縱然有諸多防護措施，但是百密一疏，若測試人員找到額外的步驟會引起報錯，那是本專案的疏失之處，敬請海涵
86. **專案自製函式**
87. function img = filtering(filter, in, kernel, mode)





1. 用來執行不同的濾波器/形態學方法，自適應彩色或灰階圖片(不需要轉換)
2. filter = 過濾器名稱(字串)
3. in = 輸入圖片
4. kernel = 濾波視窗大小/結構元素，如fspecial('gaussian', [3 3])、strel(‘disk’, 1)
5. mode = 特定濾波器特殊參數，如'same'
6. function img = ensure\_image\_range(in)



1. 用來確保圖片範圍正確
2. in = 輸入影像
3. **程式碼撰寫規則**
4. 所有變數都主要以小寫字母組成，例如temp
5. 所有函式(function)的格式都是由開頭大小英文字組成，例如Function Abc\_Def\_Ghi()
6. 若要使用常數，應該先提前定義完成
7. 變數的取名絕大多數具有一定含意，相似用途的變數取名有關
8. **測試結果(請注意第(7)點)**

使用3張圖片(測試資料)輸入

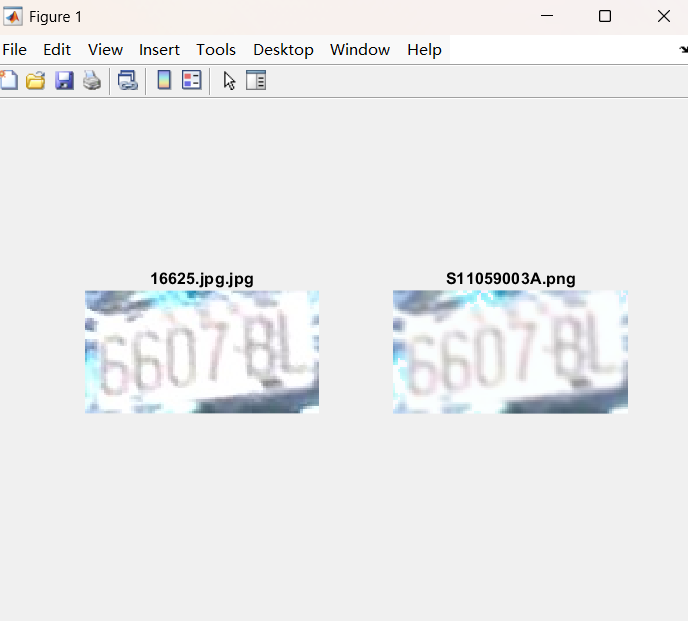
1. 第一張圖片(原名16625.jpg)
2. 原圖



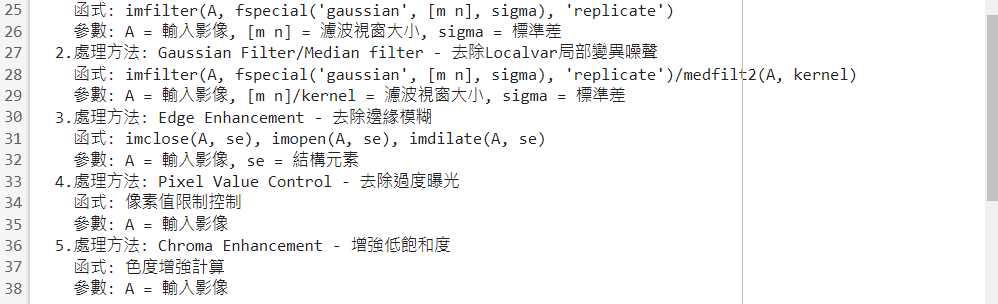
1. 處理後

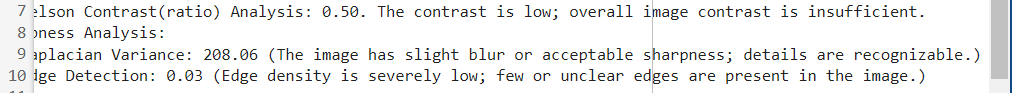


1. 對比圖(非本專案顯示，自製程式)



1. 文件內容



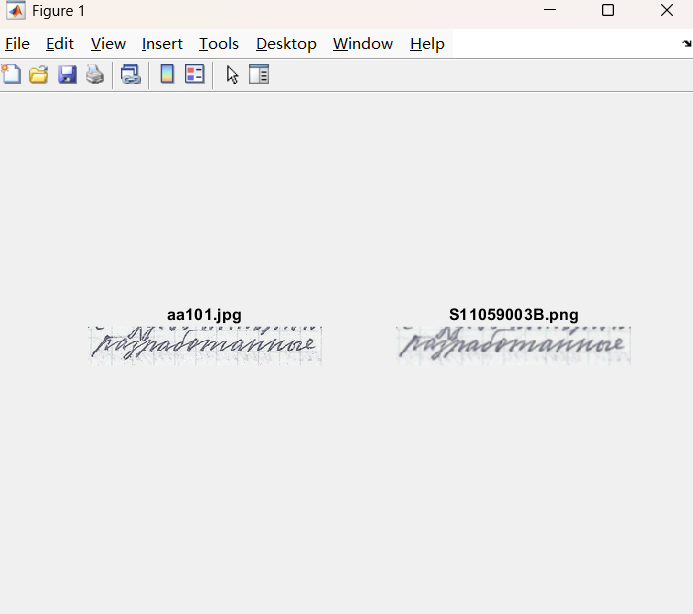
1. 第二張圖片(原名aa101.jpg)
2. 原圖



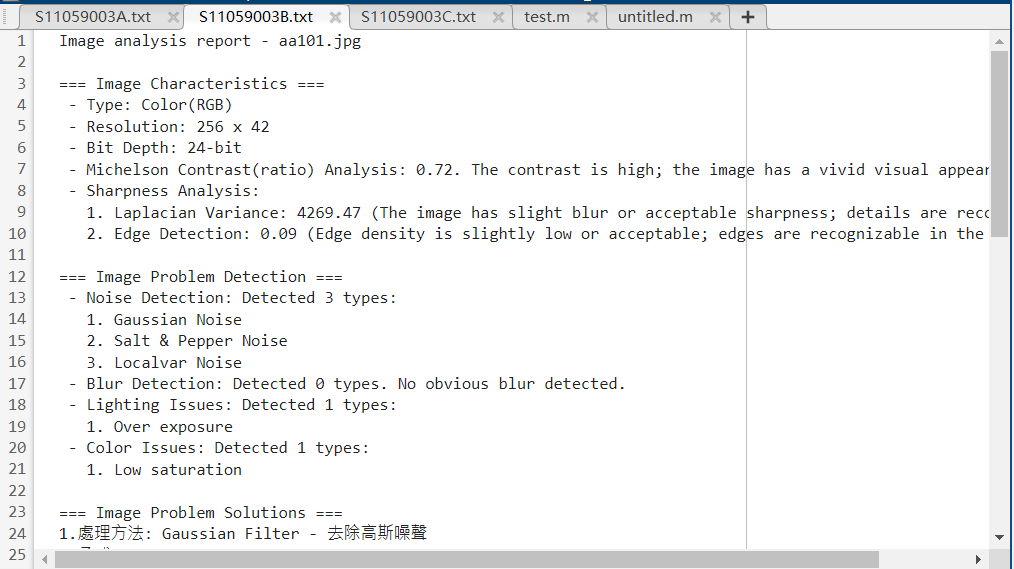
1. 處理後

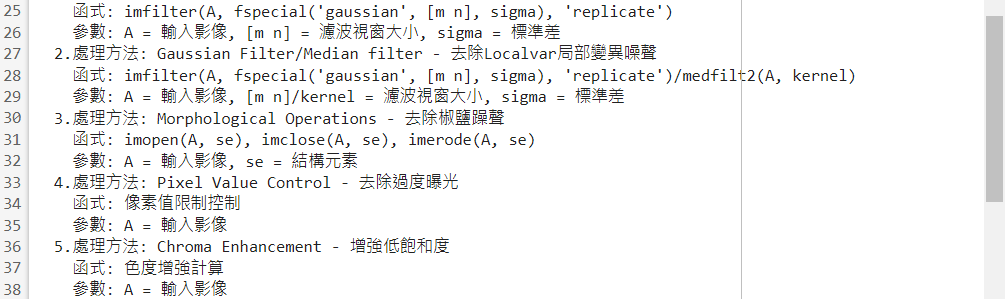


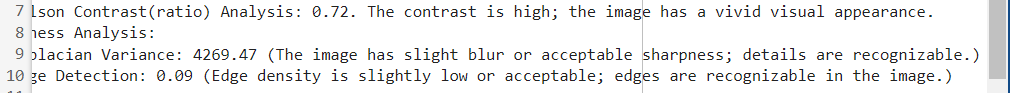
1. 對比圖(非本專案顯示，自製程式)



1. 文件內容







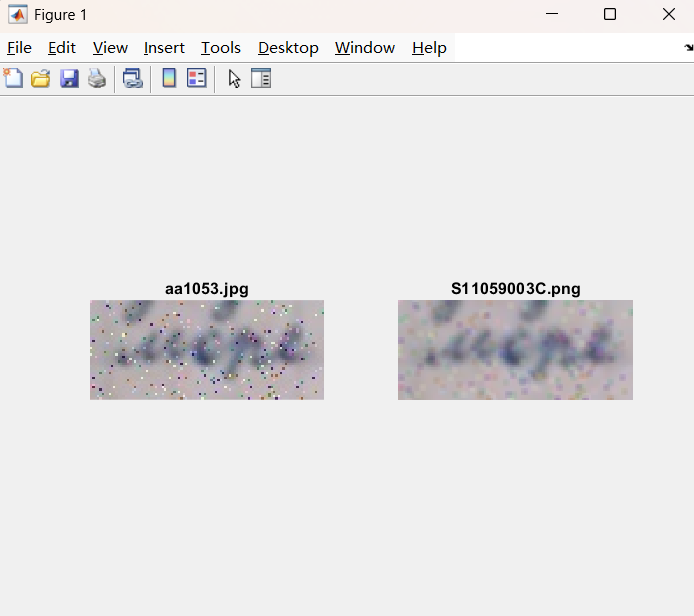
1. 03.jpg(原名aa1053.jpg)
2. 原圖



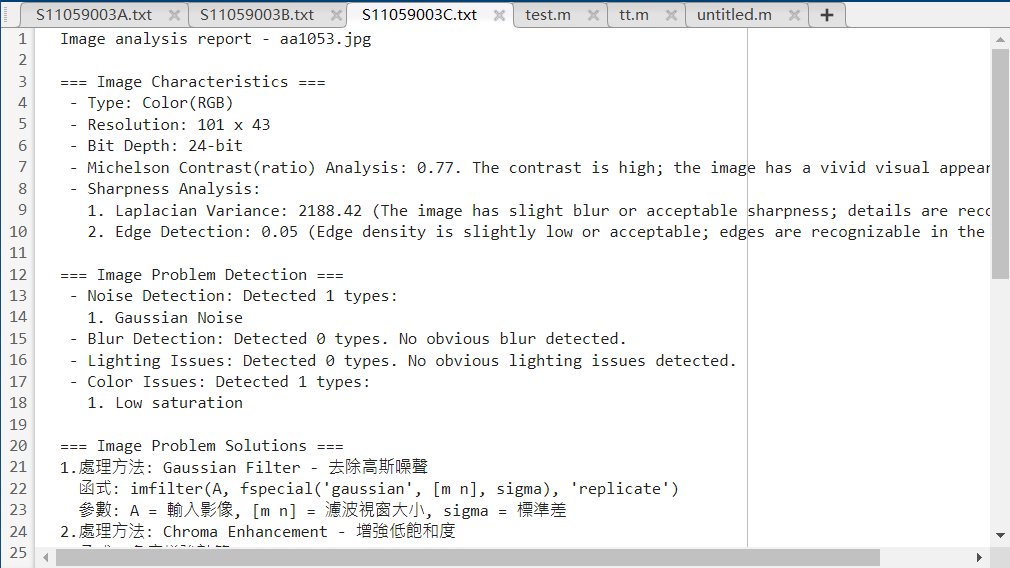
1. 處理後



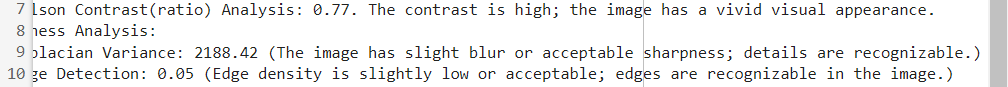
1. 對比圖(非本專案顯示，自製程式)



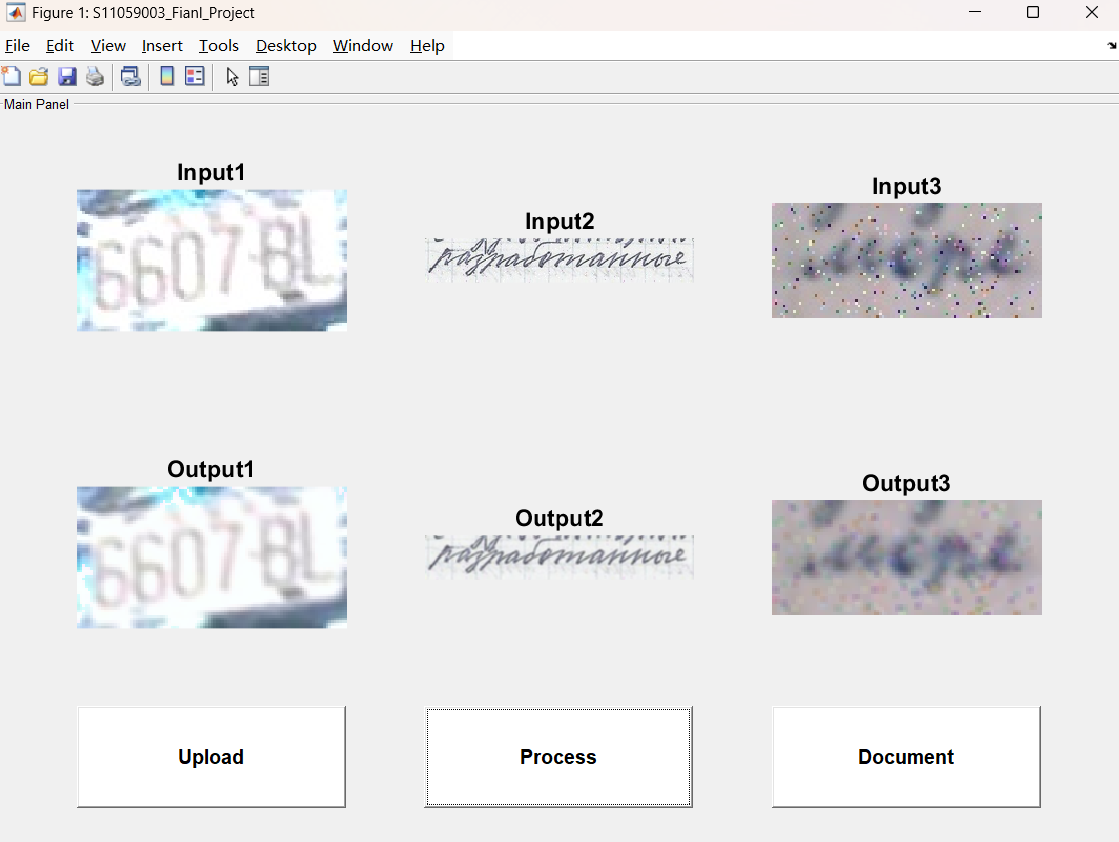
1. 文件內容



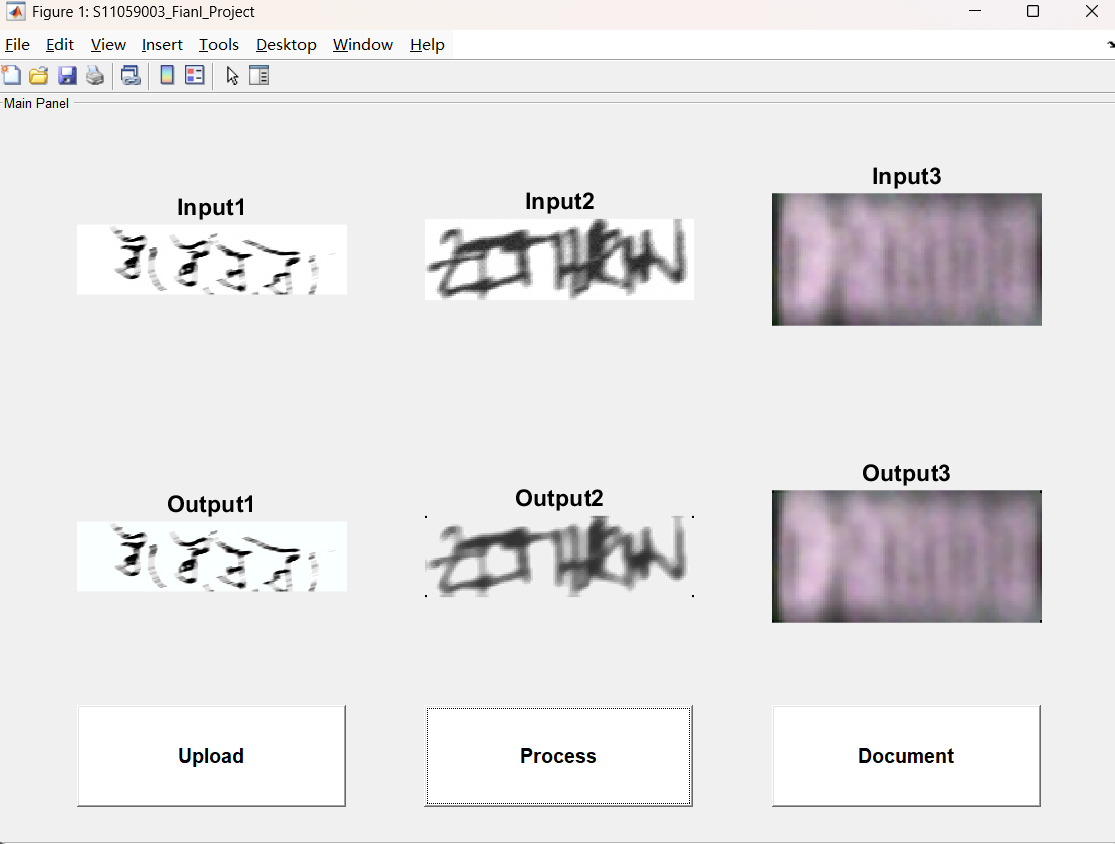




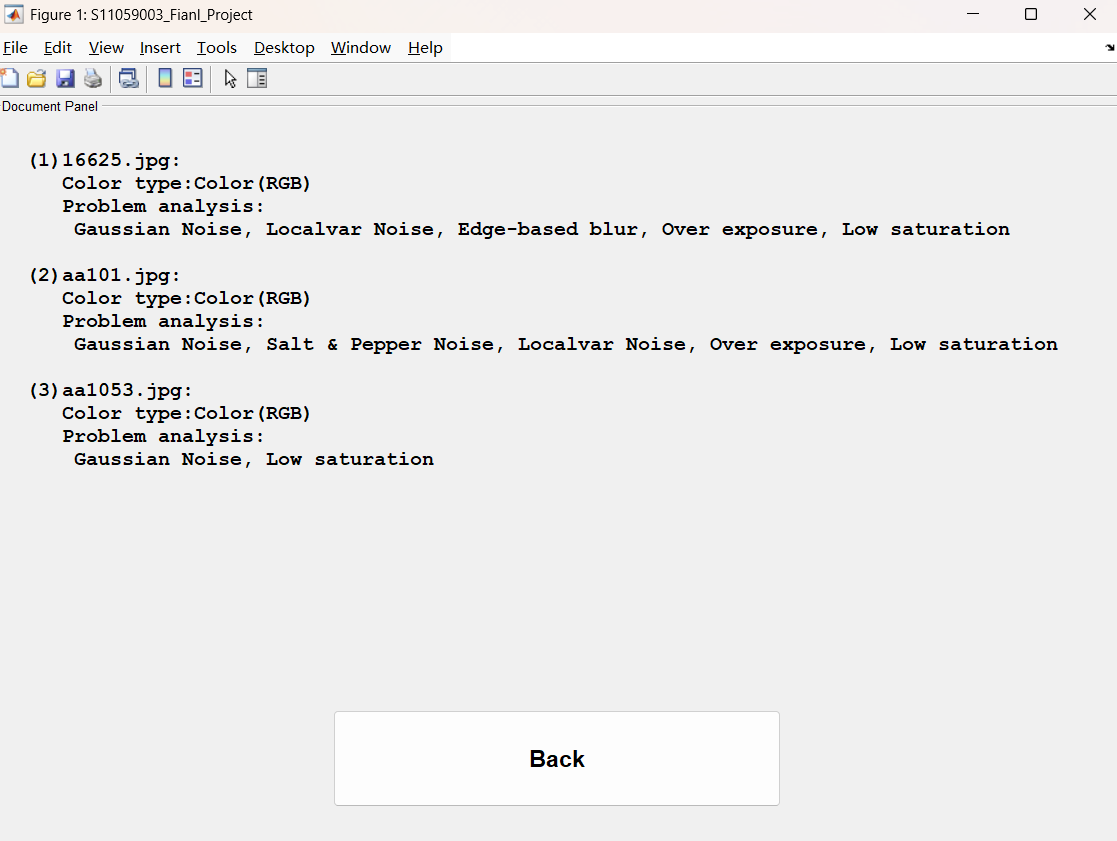
1. 程式執行結果(按下Process之後)



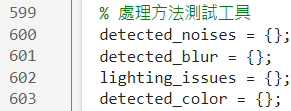
1. 程式執行結果(按下Process之後)，另外3張圖片



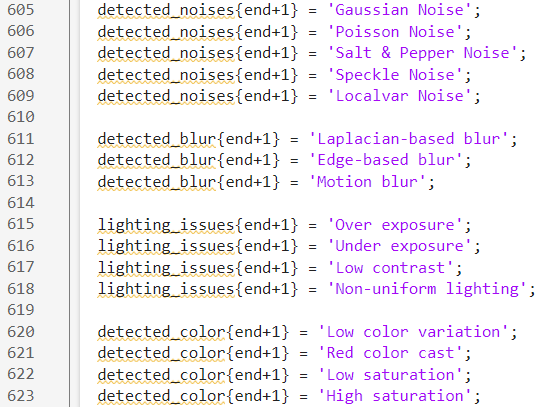
1. 程式執行結果(按下Document之後)



1. 測試「處理方法」說明
2. 位於599至623行之間
3. 打開註解(全選範圍後Ctrl+Shift+R)
4. 清空已經診斷好的問題(也就不會使用對應種類的處理方法)



1. 選擇需要測試的方法，其餘全部註解，即可僅使用想要測試的處理方法



1. **參考文獻**
2. https://onlinepngtools.com/check-if-png-is-grayscale
3. https://blog.winerva.com/2024/04/12/影像清晰度分析/
4. https://medium.com/%40sagardhungel/laplacian-and-its-use-in-blur-detection-fbac689f0f88
5. https://en.wikipedia.org/wiki/Contrast\_(vision)?utm\_source=chatgpt.com
6. https://opencv.org/blog/autofocus-using-opencv-a-comparative-study-of-focus-measures-for-sharpness-assessment/?utm\_source=chatgpt.comhttps://ww2.mathworks.cn/matlabcentral/answers/106380-can-we-find-what-kind-of-noise-present-in-an-image
7. https://ww2.mathworks.cn/matlabcentral/answers/21244-noise-detector-in-images
8. https://ww2.mathworks.cn/matlabcentral/answers/868138-how-do-i-calculate-poisson-image-noise
9. https://ww2.mathworks.cn/matlabcentral/answers/544103-how-to-remove-speckles-from-image
10. https://ww2.mathworks.cn/matlabcentral/answers/153377-how-do-i-use-imnoise-localvar
11. https://ww2.mathworks.cn/matlabcentral/answers/1750330-how-to-detect-blurred-images-in-a-dataset
12. https://ww2.mathworks.cn/matlabcentral/answers/393515-how-to-check-the-lighting-condition-of-an-image-programatically
13. https://ww2.mathworks.cn/matlabcentral/answers/689554-how-to-reduce-chromatic-aberrations-in-color-images