

# 國立臺南大學

## 資訊工程學系

### 影像處理

### Midterm Project

資工四 S11059003 曾俊杰

西元 2025 年 4 月 30 日

## 目錄

一、 文件簡介.....	4
二、 頁面(Panels).....	5
1. Figure 標題.....	5
2. 初始頁面(Main Panel).....	5
3. 注意事項.....	6
三、 初始頁面.....	6
1. 按鈕 – Upload.....	6
2. 按鈕 – Processes .....	8
3. Axe 圖片顯示區塊.....	8
4. 注意事項.....	9
四、 圖像特性分析模塊 .....	9
1. 屬性識別.....	9
2. 分析圖片清晰度.....	9
3. 分析圖片對比度.....	10
4. 注意事項.....	10
五、 問題診斷.....	11
1. 直方圖分析.....	11
2. 躁聲檢測.....	11
3. 模糊類型.....	12
4. 光線問題.....	13
5. 色彩問題.....	14
5. 注意事項.....	15

六、 處理說明.....	15
1. 高斯躁聲.....	16
2. 松柏躁聲 + 斑點躁聲 + Localvar 躁聲.....	16
3. 椒鹽躁聲.....	17
4. Laplacian Variance.....	17
5. Edge Density.....	18
6. Motion Blur Directional Sharpening .....	18
7. 過度曝光 + 光照不均 .....	19
8. 曝光不足.....	19
9. 對比度不足.....	20
10. 色彩單調.....	20
11. 色偏.....	21
12. 色彩飽和度異常.....	21
13. 色彩躁聲.....	22
14. 注意事項.....	23
七、 專案自製函式 .....	23
八、 程式碼撰寫規則 .....	24
九、 測試結果(請注意第(5)點).....	24
十、 參考文獻.....	32

## 一、 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：製作一個修復照片 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：
  - i. 輸出和輸入數量、檔名固定
  - ii. 只能使用課堂講過的相關功能(函式方法)
- (5) 成品簡介：描述輸入圖片特性，共有 5 種躁聲相關、8 種模糊相關、9 種光線相關和 13 種色彩相關問題診斷/解決方法
- (6) 注意事項：
  - i. 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
  - ii. 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

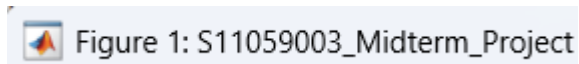


- iii. 本專案所有內容物皆為相對位置，並且有一定比例的大小，隨著 Figure 正常拉伸可以調整自己的位置，不會留在原地

## 二、 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。

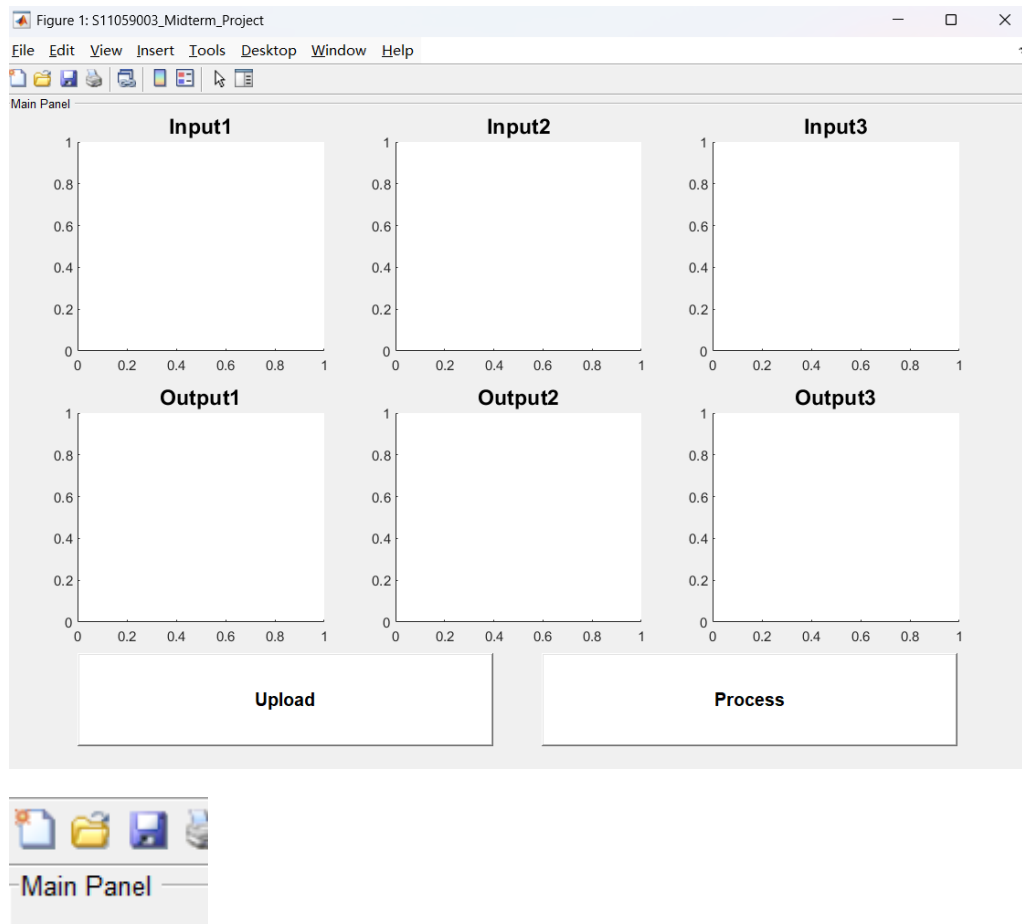
### 1. Figure 標題



### 2. 初始頁面(Main Panel)

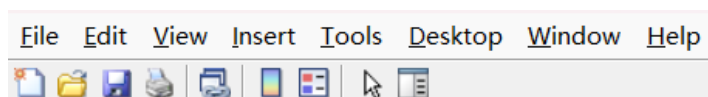
(1) 進入方式：執行.m 程式

(2) 內容：2 個按鈕、6 個 Axe 顯示圖片區塊



### 3. 注意事項

- (1) 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。



- (2) 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)

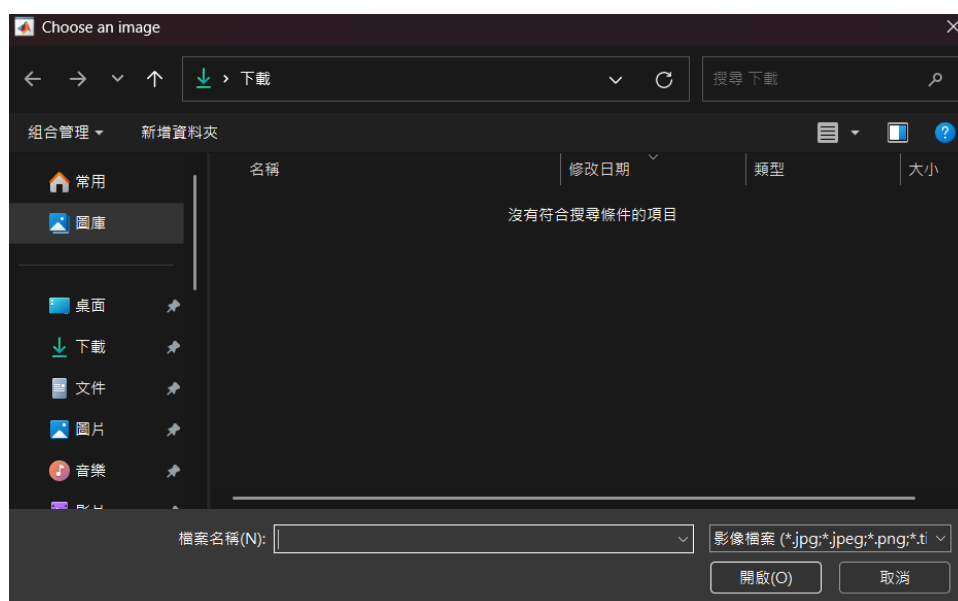


## 三、 初始頁面

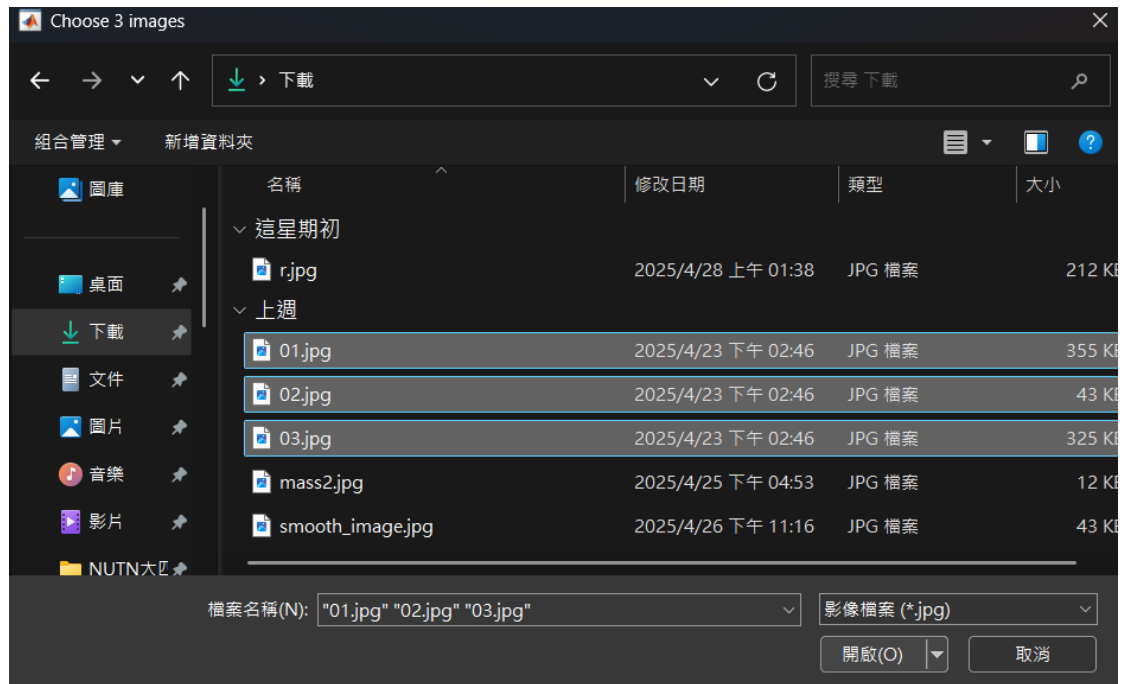
詳細介紹專案中，初始頁面的功能。

### 1. 按鈕 – Upload

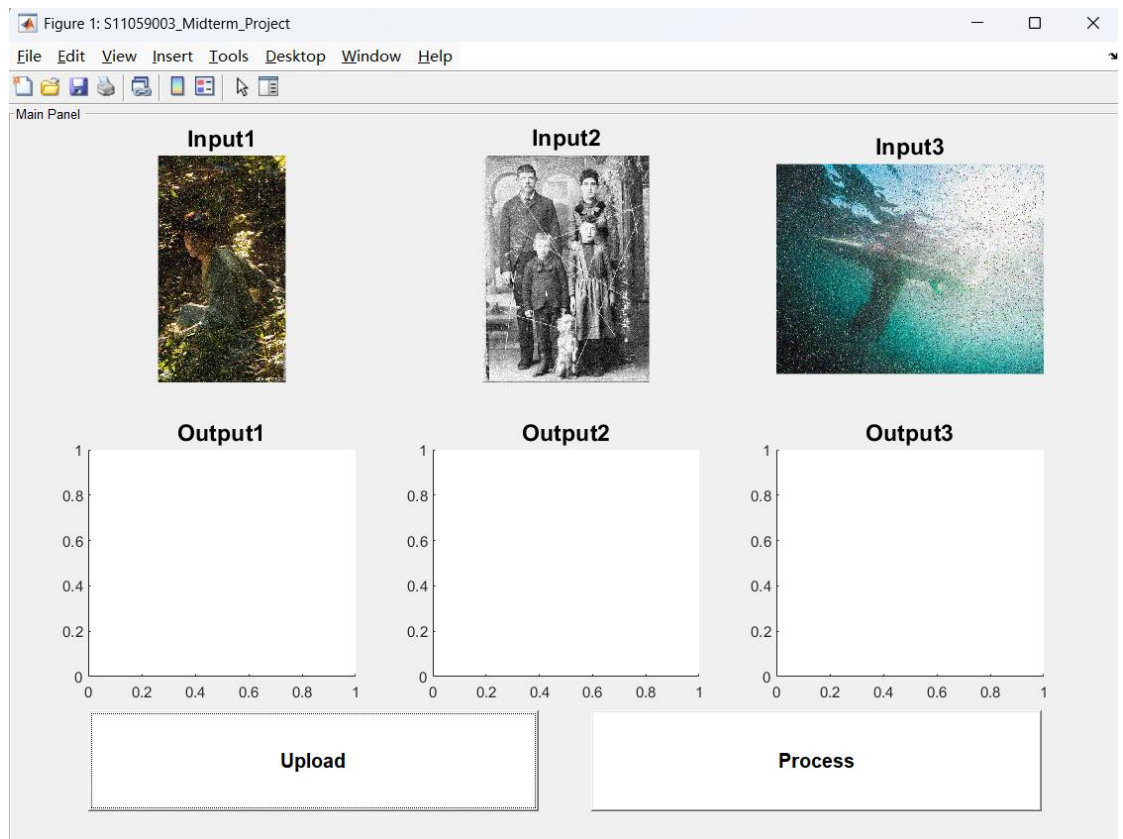
上傳輸入圖片，按下後將彈出 1 個文件選擇框，可在本地電腦中選擇任意 3 張圖片上傳，圖片副檔名限制為.jpg 圖片，由於程式內規範只搜尋指定副檔名的圖片，所以除此以外的檔案不會出現。



選擇指定 3 張圖片後，按下開啟。

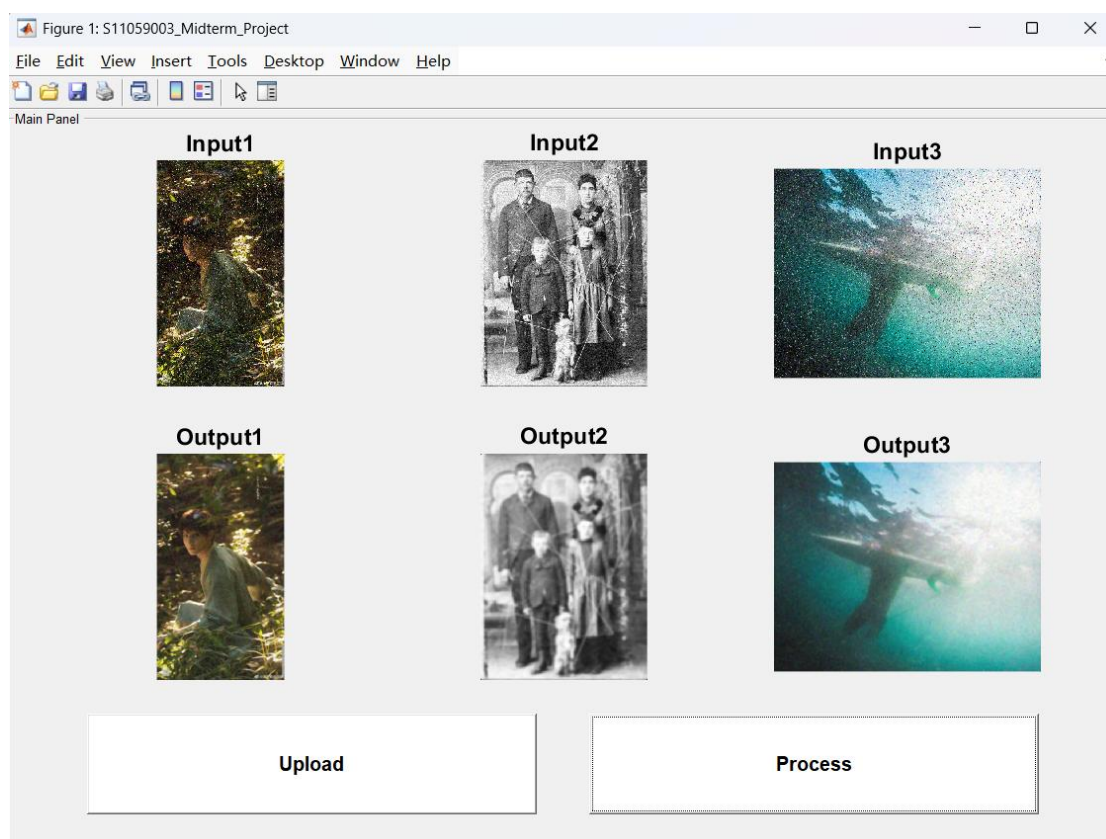


文件選擇框會自動關閉，同時，儲存輸入圖片，當作原始圖片，將其顯示在 Input1, Input2, Input3 圖片顯示區塊。



## 2. 按鈕 – Processes

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。



## 3. Axe 圖片顯示區塊

- (1) Input1, Input2, Input3：儲存 Upload Image 所選的圖片(一次 3 張)
- (2) Output1, Output2, Output3：儲存經過 Processed 功能後的圖片(一次 3 張)



## 4. 注意事項

- (1) 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)
- (2) 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)
- (3) 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

## 四、 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

- (1) 透過 imfinfo()取得屬性
- (2) 圖像類型(彩色 Color(RGB)或灰階 Grayscale)：透過 imfinfo 中的 ColorType 屬性取得
- (3) 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
- (4) 色彩深度(Bit Depth)：透過 imfinfo 中的 BitDepth 屬性取得

### 2. 分析圖片清晰度

- (1) 透過 4 種常見對比度指標，分析圖像明暗程度差異：
- (2) General Contrast：最大與最小亮度差

- (3) Luminance Contrast：最大與最小亮度差 / 最小亮度
- (4) Simple Contrast：最大亮度 / 最小亮度
- (5) 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
- (6) 每個對比度計算後，會根據數值區間給出可解讀的等級說明

### 3. 分析圖片對比度

- (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) 每種指標都會提供相對應的數值描述，如「非常清晰」、「一般」、「模糊」等

### 4. 注意事項

- (1) 圖像類型我參考過[該網站](#)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
- (2) 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
- (3) 該部分的文件格式固定，可見於九、測試結果
- (4) [對比度指標來源\(General\)](#)
- (5) [對比度指標來源\(其他\)](#)
- (6) [清晰度指標來源](#)

## 五、 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

- (1) 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
- (2) 總像素數：用來支援某些比例型判斷(如極亮極暗佔比)
- (3) 平均亮度(Mean Intensity)：反映整體影像偏暗或偏亮
- (4) 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)

### 2. 躁聲檢測

- (1) Gaussian Noise：計算整體亮度標準差與 histogram 分佈是否近似 bell shape
- (2) Poisson Noise：利用原圖與 `imnoise(I,'poisson')` 差異進行估測
- (3) Salt & Pepper Noise：判斷是否存在大量接近 0 或 255 的極端像素
- (4) Speckle Noise：分析圖像的局部方差特徵(乘性雜訊)
- (5) Localvar Noise：檢查影像中不同區塊的方差變化是否劇烈
- (6) 檢測結果儲存：在 `detected_noises` 中存入躁聲名稱
  - i. 'Gaussian Noise'
  - ii. 'Poisson Noise'

- iii. 'Salt & Pepper Noise'
- iv. 'Speckle Noise'
- v. 'Localvar Noise'

### 3. 模糊類型

- (1) Laplacian Variance：計算圖像銳利度，變異數越小越模糊
  - i. 分為 Severe、Moderate 和 Slight 等級
- (2) 邊緣密度：計算邊緣像素比例，低於閾值即判定模糊
  - i. 分為 Severe、Moderate 和 Slight 等級
- (3) 運動模糊：使用 Sobel Filter 分別檢測水平方向與垂直方向的邊緣，統計邊緣像素數量，比較兩個方向的邊緣總數，若某一方向的邊緣顯著多於另一方向，判定圖像出現有方向性的運動模糊。
  - i. 分為 horizontal 和 vertical 2 種
- (4) 檢測結果儲存：在 detected\_blur 中存入模糊種類與等級
  - i. 'Laplacian-based blur detection (severe)'
  - ii. 'Laplacian-based blur detection (moderate)'
  - iii. 'Laplacian-based blur detection (slight)'
  - iv. 'Edge-based blur detection (severe)'
  - v. 'Edge-based blur detection (moderate)'
  - vi. 'Edge-based blur detection (slight)'
  - vii. 'Motion blur detected (vertical motion)'
  - viii. 'Motion blur detected (horizontal motion)'

#### 4. 光線問題

- (1) 過度曝光：統計直方圖中亮度值大於 230 的像素比例，若亮像素佔比  $> 20\%$ ，或亮像素佔比  $> 10\%$  且平均亮度  $> 200$ ，視為過曝。
  - i. 分為 Severe、Moderate 和 Slight 等級
- (2) 曝光不足：統計亮度值小於 25 的像素比例，若暗像素佔比  $> 20\%$ ，或佔比  $> 10\%$  且平均亮度  $< 50$ ，視為曝光不足。
  - i. 分為 Severe、Moderate 和 Slight 等級
- (3) 對比度不足：使用灰階範圍 General Contrast 和標準化對比度 Michelson Contrast，若 General Contrast  $< 100$  且 Michelson Contrast  $< 0.4$ ，判定為對比度不足。
- (4) 光照不均：將圖像分割為  $4 \times 4$  區塊，計算每塊的平均亮度，評估區塊亮度範圍、變異數 (block\_var)、相鄰區塊差 (avg\_adjacent\_diff)，若亮度變異大，再根據 avg\_adjacent\_diff  $> 25$  或 block\_var  $> 800$  判定為光照不均
  - i. 分為 Severe 和 Moderate 等級
- (5) 檢測結果儲存：在 lighting\_issues 中存入光線問題種類與等級，percentage 是用來判斷曝光的百分比(根據異常 pixel 比例)
  - i. 'Over exposure(severe, percentage% bright pixels)'
  - ii. 'Over exposure(moderate, percentage% bright pixels)'
  - iii. 'Over exposure(slight, percentage% bright pixels)'
  - iv. 'Under exposure(severe, percentage% dark pixels)'
  - v. 'Under exposure(moderate, percentage% dark pixels)'
  - vi. 'Under exposure(slight, percentage% dark pixels)'
  - vii. 'Low contrast'
  - viii. 'Severe non-uniform lighting'

ix. 'Moderate non-uniform lighting'

## 5. 色彩問題

- (1) 色彩單調：分析 RGB 各通道的標準差，若所有通道標準差  $< 30$  且平均變異  $< 20$ ，視為色彩單調、缺乏變化
- (2) 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
  - i. 分為 Severe、Moderate 和 Slight 等級
  - ii. 指出最為色偏的通道(彩色/灰階)
- (3) 色彩飽和度異常：使用 HSV 模型中的飽和度通道的計算公式，根據平均與標準差進行飽和度異常判定
  - i. 分為過高/低飽和
  - ii. 分為 Severe、Moderate 和 Slight 等級
- (4) 色彩噪聲：對 R/G/B 三通道進行中值濾波，計算原圖與濾波後的差異，若三通道變異大且與中值圖平均差異  $> 8$ ，視為色彩噪聲
  - i. 分為 Severe、Moderate 和 Slight 等級
- (5) 檢測結果儲存：在 detected\_color 中存入色彩問題種類與等級，average\_saturation 是整張圖像的平均飽和度值(估算)
  - i. 'Low color variation (flat colors)'
  - ii. 'Red color cast (severe)'
  - iii. 'Red color cast (moderate)'
  - iv. 'Red color cast (slight)'
  - v. 'Green color cast (severe)'
  - vi. 'Green color cast (moderate)'

- vii. 'Green color cast (slight)'
- viii. 'Blue color cast (severe)'
- ix. 'Blue color cast (moderate)'
- x. 'Blue color cast (slight)'
- xi. 'Low saturation (severe, average\_saturation)'
- xii. 'Low saturation (moderate, average\_saturation)'
- xiii. 'Low saturation (slight, average\_saturation)'
- xiv. 'High saturation (severe, average\_saturation)'
- xv. 'High saturation (moderate, average\_saturation)'
- xvi. 'High saturation (slight, average\_saturation)'
- xvii. 'Color noise detected (severe)'
- xviii. 'Color noise detected (moderate)'
- xix. 'Color noise detected (slight)'

## 5. 注意事項

- (1) 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
- (2) 該部分的文件格式不固定(依據圖片存在的問題)，可見於九、測試結果

## 六、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)，根據**程式碼順序**撰寫，其中，以 **R** 代表最終結果圖為前提，**簡略**寫出每種方法的函式，包含以下功能：

## 1. 高斯噪聲

- (1) 問題簡述：高斯分布的隨機亮度擾動
- (2) 處理方法：Gaussian Filter
- (3) 部分函式：R = imfilter(A, fspecial('gaussian', [kernel kernel], sigma), 'replicate')
  - i. kernel = 5
  - ii. sigma = 2

## 2. 松柏噪聲 + 斑點噪聲 + Localvar 噪聲

- (1) 問題簡述：
  - i. Speckle Noise：乘法性雜訊
  - ii. Poisson Noise：與亮度有關的機率性噪聲
  - iii. Localvar Noise：局部變異雜訊，每個像素雜訊程度不同
- (2) 處理方法：Combined Median and Gaussian Filter
- (3) 部分函式(按照順序)：
  - i. B = medfilt2(A, [kernel1 kernel1])
    - (i) kernel1 = 3
  - ii. R = imfilter(B, fspecial('gaussian', [kernel2 kernel2], sigma), 'replicate')
    - (i) kernel2 = 5
    - (ii) sigma = 1.2, 1.8, 1.5(根據標題噪聲順序)



### 3. 椒鹽噪聲

(1) 問題簡述：隨機產生極黑(0)或極白(255)像素點

(2) 處理方法：Adaptive Median Filter(多層次)

(3) 部分函式(按照順序)：

i.  $R = \text{medfilt2}(A, [\text{kernel1 } \text{kernel1}])$

(i)  $\text{kernel1} = 5$

ii. if (極端像素比例) > 限值

$R = \text{medfilt2}(A, [\text{kernel2 } \text{kernel2}])$

(i)  $\text{kernel2} = 3$

### 4. Laplacian Variance

(1) 問題簡述：檢測到 Laplacian-based 模糊(嚴重/中度/輕微)

(2) 處理方法：

i. 嚴重或中度模糊：使用高提升濾波(High Boost Filter)，增強邊緣強度。

ii. 輕微模糊：使用基本 Laplacian 銳化濾波器

(3) 部分函式：

i.  $R = \text{imfilter}(A, \text{kernel}, 'replicate')$

$A = A + 0.5 * B$

(i)  $\text{kernel} = [-1 \ -1 \ -1; -1 \ 8 \ -1; -1 \ -1 \ -1]$

ii.  $R = \text{conv2}(A, \text{kernel}, 'same')$

(i)  $\text{kernel} = [0 \ -1 \ 0; -1 \ 5 \ -1; 0 \ -1 \ 0]$

## 5. Edge Density

(1) 問題簡述：當檢測到 Edge-based 模糊(嚴重/中度/輕微)

(2) 處理方法：

- i. 嚴重/中度模糊：使用 Sobel Filter 計算邊緣梯度，並以梯度為權重進行增強
- ii. 輕微模糊：使用低強度高提升濾波器改善邊緣

(3) 部分函式：

- i.  $B = \text{imfilter}(A, \text{sobel\_x or sobel\_y})$   
 $R = B + \alpha * B .* \text{normalized\_gradient}$ 
  - (i)  $\alpha = 0.3$ , if 中度
  - (ii)  $\alpha = 0.5$ , if 嚴重
- ii.  $B = \text{imfilter}(A, \text{kernel}, 'replicate')$   
 $C = B / \max(\text{abs}(B(:))) * 50$   
 $R = B + C$ 
  - (i)  $\text{kernel} = [-1 \ -1 \ -1; -1 \ 8 \ -1; -1 \ -1 \ -1]$

## 6. Motion Blur Directional Sharpening

(1) 問題簡述：當檢測到 Motion Blur(水平或垂直)

(2) 處理方法：根據模糊方向使用對應的方向性銳化濾波器，濾波後結果進行正規化並加至原圖

(3) 部分函式：

- i.  $B = \text{imfilter}(A, \text{kernel}, 'replicate')$   
 $C = B / \max(\text{abs}(B(:))) * 40$   
 $R = B + C$

- (i)  $\text{kernel} = [-1 \ -1 \ -1; 2 \ 2 \ 2; -1 \ -1 \ -1]$
- ii.  $B = \text{imfilter}(A, \text{kernel}, 'replicate')$
- $C = B / \max(\text{abs}(B(:))) * 40$
- $R = B + C$
- (i)  $\text{kernel} = [-1 \ 2 \ -1; -1 \ 2 \ -1; -1 \ 2 \ -1]$

## 7. 過度曝光 + 光照不均

- (1) 問題簡述：當檢測到過度曝光(嚴重/中度/輕微)與光照不均(嚴重/中度)
- (2) 處理方法：
  - i. 計算每個色彩通道的平均亮度
  - ii. 根據亮度高低選擇適當的  $\gamma$  值進行 power law 調整
    - (i) 若平均亮度  $> 200$ ，代表過度曝光， $\gamma = 1.5$ (降低亮部亮度)
    - (ii) 若平均亮度  $< 80$ ，代表偏暗， $\gamma = 0.7$ (提升亮部)
- (3) 部分函式： $R = A.^{\gamma}$ 
  - i. 過度曝光： $\gamma = 1.5$
  - ii. 偏暗： $\gamma = 0.7$
  - iii. 否則： $\gamma = 1$

## 8. 曝光不足

- (1) 問題簡述：當檢測到曝光不足(嚴重/中度/輕微)
- (2) 處理方法：

- i. 對整張影像進行對數轉換來強化暗部細節
  - ii. 根據曝光不足程度選擇增強係數  $c$
- (3) 部分函式： $R = c * \log(1 + A)$
- i. 嚴重： $c = 2$
  - ii. 中度： $c = 1.5$
  - iii. 輕微： $c = 1.2$

## 9. 對比度不足

- (1) 問題簡述：當檢測到對比度不足
- (2) 處理方法：
- i. 對每個色彩通道使用 stretchlim 計算上下限
  - ii. 將像素值線性映射至  $[0,1]$  範圍，拉伸對比
- (3) 部分函式： $R = (A - \min) / (\max - \min)$
- i.  $\max$  = 各通道的灰階上限
  - ii.  $\min$  = 各通道的灰階下限

## 10. 色彩單調

- (1) 問題簡述：當檢測到色彩單調
- (2) 處理方法：對比度調整
- i. 針對色彩變異性低(畫面顏色平坦、缺乏層次)情況，透過對整體像素值乘以係數  $\text{contrast}$ ，來拉開像素差異，使圖像色彩更有層次感
- (3) 部分函式： $R = A * \text{contrast}$

- i.  $\text{contrast} = 1.3$

## 11. 色偏

- (1) 問題簡述：當檢測到色偏(最為色偏的通道, 嚴重/中度/輕微)
- (2) 處理方法：Channel-Based Color Balancing
- (3) 部分函式：
  - i.  $\text{ratio}\{j\} = \text{avg\_gray} / \text{avg\_channel}\{j\};$
  - ii.  $\text{t1}\{j\} = \min(255, \max(0, \text{channel}\{j\} * \text{ratio}\{j\}));$
  - iii.  $R = \text{cat}(3, \text{t1}\{1\}, \text{t1}\{2\}, \text{t1}\{3\});$ 
    - (i)  $\text{ratio}(R, G, B)$ ：根據通道與灰度平均值自動計算(範圍限制 0.6~1.5)

## 12. 色彩飽和度異常

- (1) 飽和度過低
  - i. 問題簡述：當檢測到飽和度過低(嚴重/中度/輕微)
  - ii. 處理方法：
    - (i) 色差強化(亮度與色差分離增強法)
    - (ii) 將亮度與色差分離後，強化色差再加入亮度，提升整體飽和度
  - iii. 部分函式：
    - (i)  $r\_diff = r - \text{luminance};$
    - (ii)  $r\_new = \text{luminance} + r\_diff * \text{enhancement};$
    - (iii)  $R = \text{cat}(3, r\_new, g\_new, b\_new);$

(iv) enhancement = 2.0/1.7/1.5(嚴重/中度/輕微)

(2) 飽和度過高

- i. 問題簡述：當檢測到飽和度過高(嚴重/中度/輕微)
- ii. 處理方法：
  - (i) 色差減弱(亮度與色差分離減弱法)
  - (ii) 反向處理，透過減弱色差來降低過度鮮豔的色彩，達到自然效果
- iii. 部分函式：
  - (i)  $r\_diff = r - luminance$ ;
  - (ii)  $r\_new = luminance + r\_diff * reduction$ ;
  - (iii)  $R = cat(3, r\_new, g\_new, b\_new)$ ;
  - (iv) reduction = 0.5/0.6/0.7(嚴重/中度/輕微)

### 13. 色彩躁聲

- (1) 問題簡述：當檢測到色彩躁聲(嚴重/中度/輕微)
- (2) 處理方法：Gaussian Filter
  - i. 針對出現的色斑或不規則色點，使用 fspecial('gaussian')生成平滑濾波器進行通道分離降噪，保留結構、降低色彩雜訊
- (3) 部分函式： $R = imfilter(result, fspecial('gaussian', kernel sigma), 'replicate')$ 
  - i. kernel = 5, sigma = 1.5, if 中度
  - ii. kernel = 7, sigma = 1.8, if 嚴重

## 14. 注意事項

- (1) 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
- (2) 該部分的文件格式不固定(依據圖片存在的問題)，可見於九、測試結果，尤其是想要測試特定的處理，方法就寫在九-(5)，助教可前去查看，以節省時間
- (3) 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

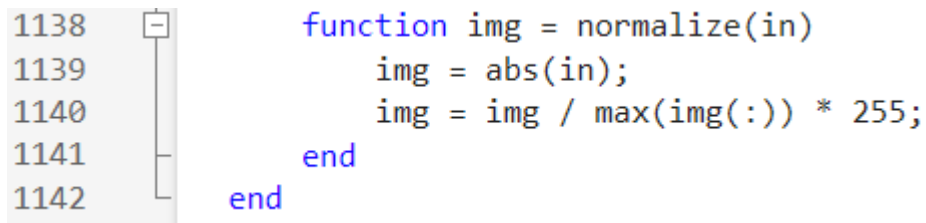
## 七、專案自製函式

1. function img = filtering(filter, in, kernel, mode)

```
1121 function img = filtering(filter, in, kernel, mode)
1122     times = size(in, 3);
1123     if filter == "medfilt2"
1124         for j = 1:times
1125             img(:,:,j) = medfilt2(in(:,:,j), kernel);
1126         end
1127     elseif filter == "imfilter"
1128         for j = 1:times
1129             img(:,:,j) = imfilter(in(:,:,j), kernel, mode);
1130         end
1131     elseif filter == "conv2"
1132         for j = 1:times
1133             img(:,:,j) = conv2(in(:,:,j), kernel, mode);
1134         end
1135     end
1136 end
```

- (1) 用來執行不同的濾波器，自適應彩色或灰階圖片(不需要轉換)
- (2) filter = 濾波器名稱字串
- (3) in = 輸入影像
- (4) kernel = 濾波視窗大小，如 fspecial('gaussian', [3 3])
- (5) mode = 濾波器模式，如'same'

2. function img = normalize(in)



```
function img = normalize(in)
    img = abs(in);
    img = img / max(img(:)) * 255;
end
```

(1) 用來正規化圖片

(2) in = 輸入影像

## 八、 程式碼撰寫規則

1. 所有變數都主要以**小寫字母**組成，例如 temp
2. 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()
3. 若要使用常數，應該先提前定義完成
4. 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

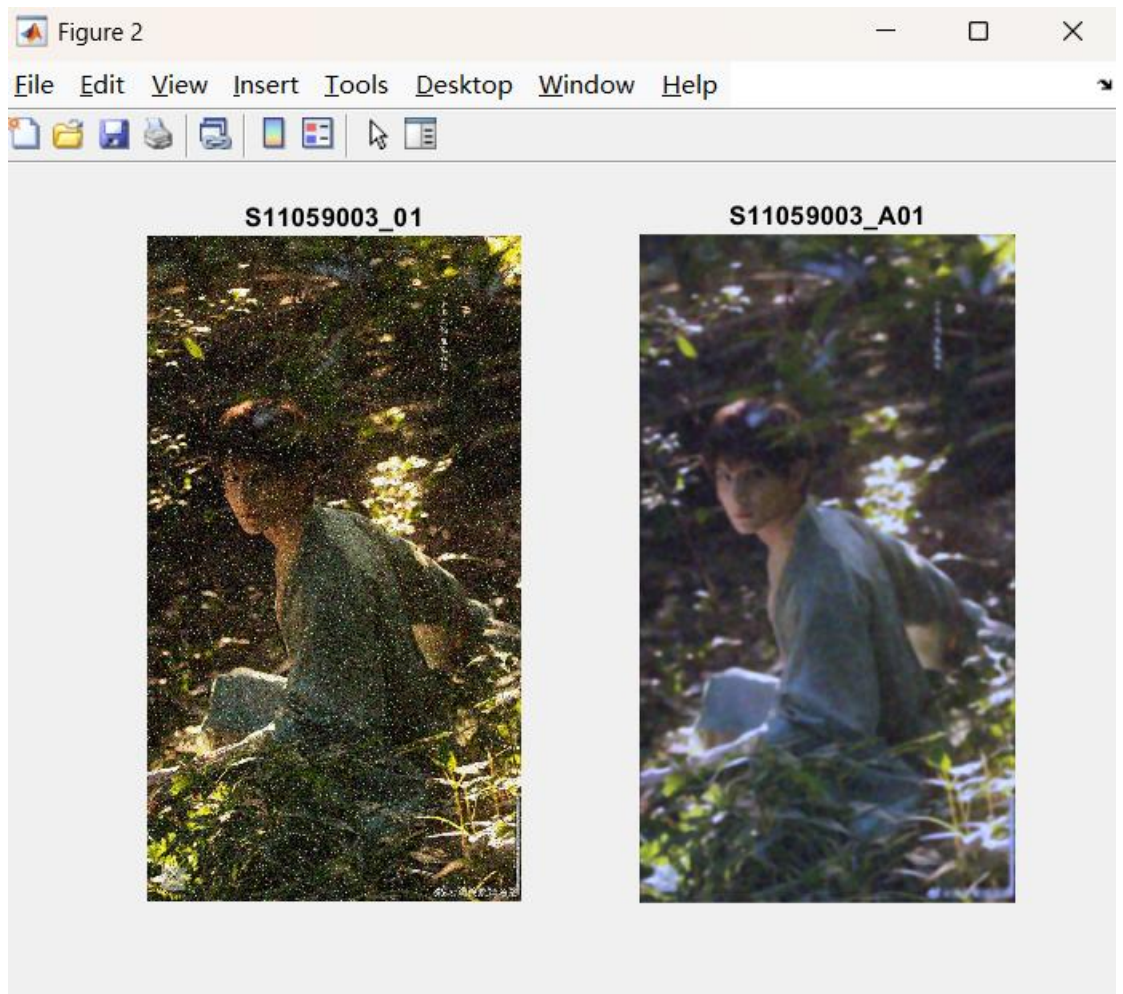
## 九、 測試結果(請注意第(5)點)

使用 3 張圖片(測試資料)輸入



(1) 01.jpg(原名 light.jpg)

i. 對比圖片



## ii. 文件內容

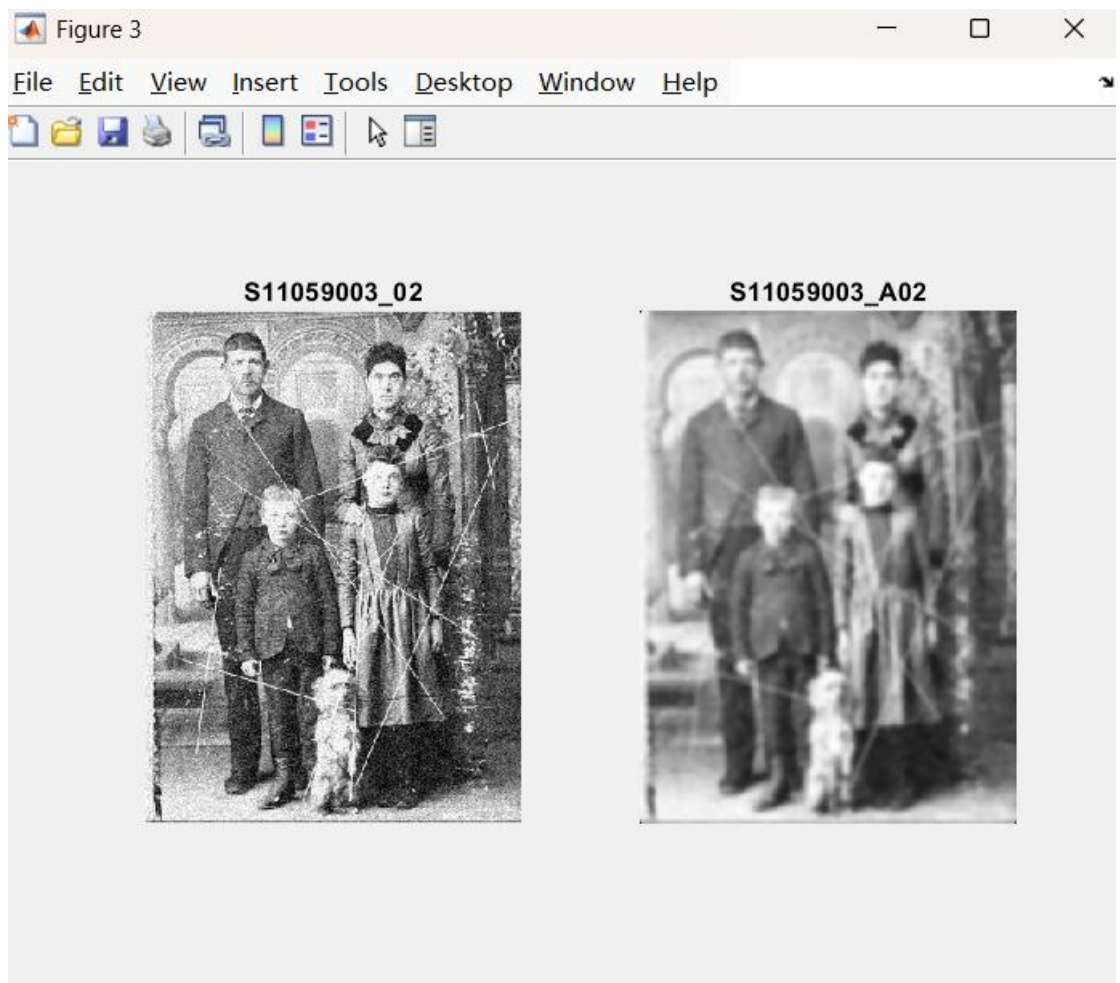
```

1 | Image analysis report - 01.jpg
2 |
3 | === Image Characteristics ===
4 | - Type: Color(RGB)
5 | - Resolution: 675 x 1200
6 | - Bit Depth: 24-bit
7 | - Contrast(ratio) Analysis:
8 |   1. General Contrast: 255.00 (General Contrast is very high; the image has extreme differences betwe
9 |   2. Luminance contrast: 1148417904979476480.00 (Luminance contrast is extremely high; very large di
10 |   3. Simple contrast: 1148417904979476480.00 (Simple contrast is extremely high; the brightest and d
11 |   4. Michelson Contrast: 1.00 (Michelson contrast is extremely high; suitable for high-contrast disp
12 | - Sharpness Analysis:
13 |   1. Laplacian Variance: 10547.29 (The image is extremely sharp with rich details and crisp edges.)
14 |   2. Edge Detection: 0.03 (Edge density is moderate; edges are clearly visible in the image.)
15 |
16 | === Image Problem Detection ===
17 | - Noise Detection: Detected 3 types:
18 |   1. Gaussian Noise
19 |   2. Salt & Pepper Noise
20 |   3. Speckle Noise
21 | - Blur Detection: Detected 0 types. No obvious blur detected.
22 | - Lighting Issues: Detected 1 types:
23 |   1. Under exposure( slight, 20.5% dark pixels)
24 | - Color Issues: Detected 2 types:
25 |   1. Red color cast ( slight)
26 |   2. Color noise detected ( severe)
27 |
28 | === Image Problem Solutions ===
29 | 1.處理方法: Gaussian Filter - 去除高斯噪聲
30 |   函式: imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
31 |   參數: A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差
32 |
33 | 2.處理方法: Combined Median and Gaussian Filter - 去除斑點噪聲
34 |   函式: medfilt2(A, [m n]) 和 imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
35 |   參數: A = 輸入影像, [m n] = 濾波視窗大小 (中值濾波)
36 |         A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差 (高斯濾波)
37 |
38 | 3.處理方法: Adaptive Median Filter - 去除椒鹽噪聲
39 |   函式: medfilt2(A, [m n])
40 |   參數: A = 輸入影像, [m n] = 濾波視窗大小
41 |
42 | 4.處理方法: Log Transformation - 增強曝光不足區域
43 |   函式: c * log(1 + A)
44 |   參數: A = 正規化輸入影像(0~1範圍), c = 增強係數
45 |
46 | 5.處理方法: Channel-Based Color Balancing - 校正色偏
47 |   函式: 根據各通道平均值自動平衡色彩偏差
48 |   參數: ratios(by channel(s)) = 0.84, 0.92, 1.38
49 |
50 | 6.處理方法: 高斯濾波 - 降低色彩噪聲
51 |   函式: imfilter(channel, fspecial('gaussian', kernel_size, sigma), 'replicate')
52 |   參數: kernel_size = [7 7]
53 |         sigma = 1.8
54 |
55 | Contrast is very high; the image has extreme differences between light and dark regions.)
56 | 1148417904979476480.00 (Luminance contrast is extremely high; very large differences between dark and bright regions.)
57 | 1148417904979476480.00 (Simple contrast is extremely high; the brightest and darkest regions differ greatly.)
58 | 1.00 (Michelson contrast is extremely high; suitable for high-contrast display.)
59 |
60 | image is extremely sharp with rich details and crisp edges.)
61 | is moderate; edges are clearly visible in the image.)

```

(2) 02.jpg(原名 old\_photo.jpg)

i. 對比圖片

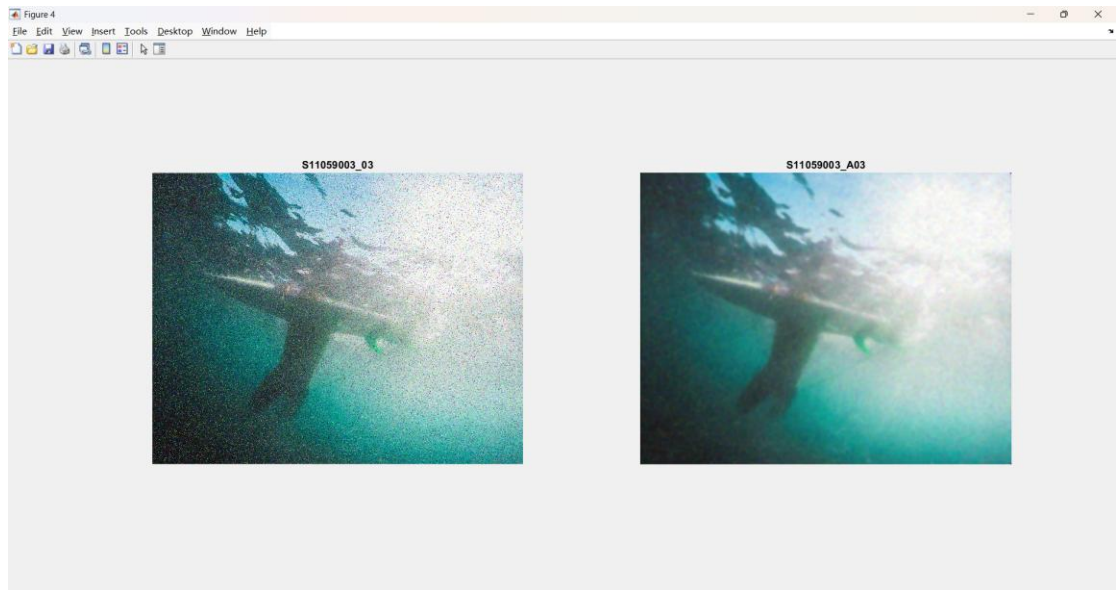


## ii. 文件內容

```
1 |Image analysis report - 02.jpg
2
3 |=== Image Characteristics ===
4 | - Type: Grayscale
5 | - Resolution: 294 x 400
6 | - Bit Depth: 8-bit
7 | - Contrast(ratio) Analysis:
8 |   1. General Contrast: 255.00 (General Contrast is very high; the image has extreme differences betw
9 |   2. Luminance contrast: 1148417904979476480.00 (Luminance contrast is extremely high; very large di
10 |   3. Simple contrast: 1148417904979476480.00 (Simple contrast is extremely high; the brightest and d
11 |   4. Michelson Contrast: 1.00 (Michelson contrast is extremely high; suitable for high-contrast disp
12 | - Sharpness Analysis:
13 |   1. Laplacian Variance: 7614.14 (The image is extremely sharp with rich details and crisp edges.)
14 |   2. Edge Detection: 0.05 (Edge density is moderate; edges are clearly visible in the image.)
15
16 |=== Image Problem Detection ===
17 | - Noise Detection: Detected 3 types:
18 |   1. Gaussian Noise
19 |   2. Salt & Pepper Noise
20 |   3. Speckle Noise
21 | - Blur Detection: Detected 0 types. No obvious blur detected.
22 | - Lighting Issues: Detected 1 types:
23 |   1. Severe non-uniform lighting
24 | - Color Issues: Not applicable (Grayscale image).
25
26 |=== Image Problem Solutions ===
27 | 1.處理方法: Gaussian Filter - 去除高斯噪聲
28 |   函式: imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
29 |   參數: A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差
30
31 | 2.處理方法: Combined Median and Gaussian Filter - 去除斑點噪聲
32 |   函式: medfilt2(A, [m n]) 和 imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
33
34 |   函式: medfilt2(A, [m n]) 和 imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
35 |   參數: A = 輸入影像, [m n] = 濾波視窗大小 (中值濾波)
36 |         A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差 (高斯濾波)
37
38 | 3.處理方法: Adaptive Median Filter - 去除椒鹽噪聲
39 |   函式: medfilt2(A, [m n])
40 |   參數: A = 輸入影像, [m n] = 濾波視窗大小
41
42 | 4.處理方法: Power Law Transformation - 修正光照不均
43 |   函式: A .^ gamma
44 |   參數: A = 輸入影像, gamma = power law factor
45
46 | 8 |contrast is very high; the image has extreme differences between light and dark regions.)
47 | 9 |76480.00 (Luminance contrast is extremely high; very large differences between dark and bright regions.)
48 |10 |30.00 (Simple contrast is extremely high; the brightest and darkest regions differ greatly.)
49 |11 |n contrast is extremely high; suitable for high-contrast display.)
50 |12 |
51 |13 |mage is extremely sharp with rich details and crisp edges.)
52 |14 |is moderate; edges are clearly visible in the image.)
```

### (3) 03.jpg(原名 water.jpg)

#### i. 對比圖片



#### ii. 文件內容

1	Image analysis report - 03.jpg	
2		
3	=== Image Characteristics ===	
4	- Type: Color(RGB)	
5	- Resolution: 996 x 781	
6	- Bit Depth: 24-bit	
7	- Contrast(ratio) Analysis:	
8	1. General Contrast: 255.00 (General Contrast is very high; the image has extreme differences betw	
9	2. Luminance contrast: 1148417904979476480.00 (Luminance contrast is extremely high; very large di	
10	3. Simple contrast: 1148417904979476480.00 (Simple contrast is extremely high; the brightest and d	
11	4. Michelson Contrast: 1.00 (Michelson contrast is extremely high; suitable for high-contrast disp	
12	- Sharpness Analysis:	
13	1. Laplacian Variance: 10099.49 (The image is extremely sharp with rich details and crisp edges.)	
14	2. Edge Detection: 0.04 (Edge density is moderate; edges are clearly visible in the image.)	
15		
16	=== Image Problem Detection ===	
17	- Noise Detection: Detected 3 types:	
18	1. Gaussian Noise	
19	2. Salt & Pepper Noise	
20	3. Speckle Noise	
21	- Blur Detection: Detected 0 types. No obvious blur detected.	
22	- Lighting Issues: Detected 1 types:	
23	1. Severe non-uniform lighting	
24	- Color Issues: Detected 1 types:	
25	1. Color noise detected (severe)	
26		
27	=== Image Problem Solutions ===	
28	1.處理方法: Gaussian Filter - 去除高斯噪聲	
29	函式: imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')	
30	參數: A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差	
31		
32	2.處理方法: Combined Median and Gaussian Filter - 去除斑點噪聲	



```

33 函式: medfilt2(A, [m n]) 和 imfilter(A, fspecial('gaussian', [m n], sigma), 'replicate')
34 參數: A = 輸入影像, [m n] = 濾波視窗大小 (中值濾波)
35       A = 輸入影像, [m n] = 濾波視窗大小, sigma = 標準差 (高斯濾波)
36
37 3.處理方法: Adaptive Median Filter - 去除椒鹽噪聲
38 函式: medfilt2(A, [m n])
39 參數: A = 輸入影像, [m n] = 濾波視窗大小
40
41 4.處理方法: Power Law Transformation - 修正光照不均
42 函式: A .^ gamma
43 參數: A = 輸入影像, gamma = power law factor
44
45 5.處理方法: 高斯濾波 - 降低色彩噪聲
46 函式: imfilter(channel, fspecial('gaussian', kernel_size, sigma), 'replicate')
47 參數: kernel_size = [7 7]
48       sigma = 1.8

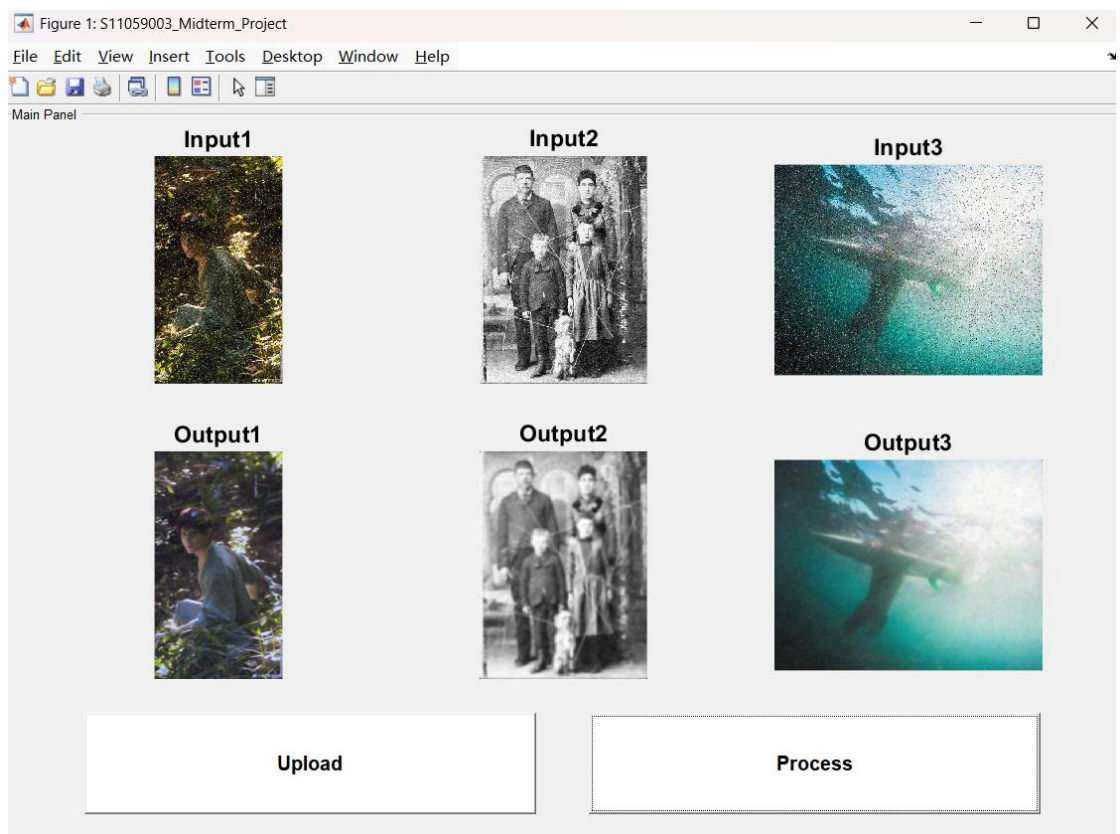
```

```

8 Contrast is very high; the image has extreme differences between light and dark regions.)
9 76480.00 (Luminance contrast is extremely high; very large differences between dark and bright regions.)
10 30.00 (Simple contrast is extremely high; the brightest and darkest regions differ greatly.)
11 n contrast is extremely high; suitable for high-contrast display.)
12
13 image is extremely sharp with rich details and crisp edges.)
14 is moderate; edges are clearly visible in the image.)

```

#### (4) 程式執行結果(按下 Process 之後)



(5) 測試說明

i. 建議使用在 685 行之後

ii. 躁聲處理：始於 686 行

(i) 特定處理使用方法：detected\_noises{end+1} = '躁聲名稱';

(ii) 模糊/光線/色彩處理可以依此類推

```
detected_noises{end+1} = 'Gaussian Noise';  
detected_noises{end+1} = 'Poisson Noise';  
if ismember('Gaussian Noise', detected_noises)  
    ...  
noise types = {'Poisson Noise', 'Speckle Noise', 'Localvar Noise'}
```

iii. 模糊處理：始於 766 行

(i) detected\_blur{end+1} = ";

```
detected_blur{end+1} = 'Laplacian-based blur detection (severe)';  
if ismember('Laplacian-based blur detection (severe)', detected_blur) || ismember('Laplaciar  
    ...
```

iv. 光線處理：始於 884 行

(i) lighting\_issues{end+1} = ";

```
lighting_issues{end+1} = 'Under exposure';  
if any(contains(lighting_issues, 'Under exposure'))
```

v. 色彩處理：始於 950 行

(i) detected\_color{end+1} = 'Red color cast';

```
% 色彩問題診斷 - 色偏  
detected_color{end+1} = 'Red color cast';  
if any(contains(detected_color, {'Red color cast', 'Green color cast', 'Blue c
```

## 十、 參考文獻

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