

影像預處理

Image Preprocessing

使用說明 Manual

[GitHub 連結](#)

目錄

主選單.....	1
File 選單	4
Edit 選單	6
Colors 選單.....	7
Spatial 選單.....	9

主視窗

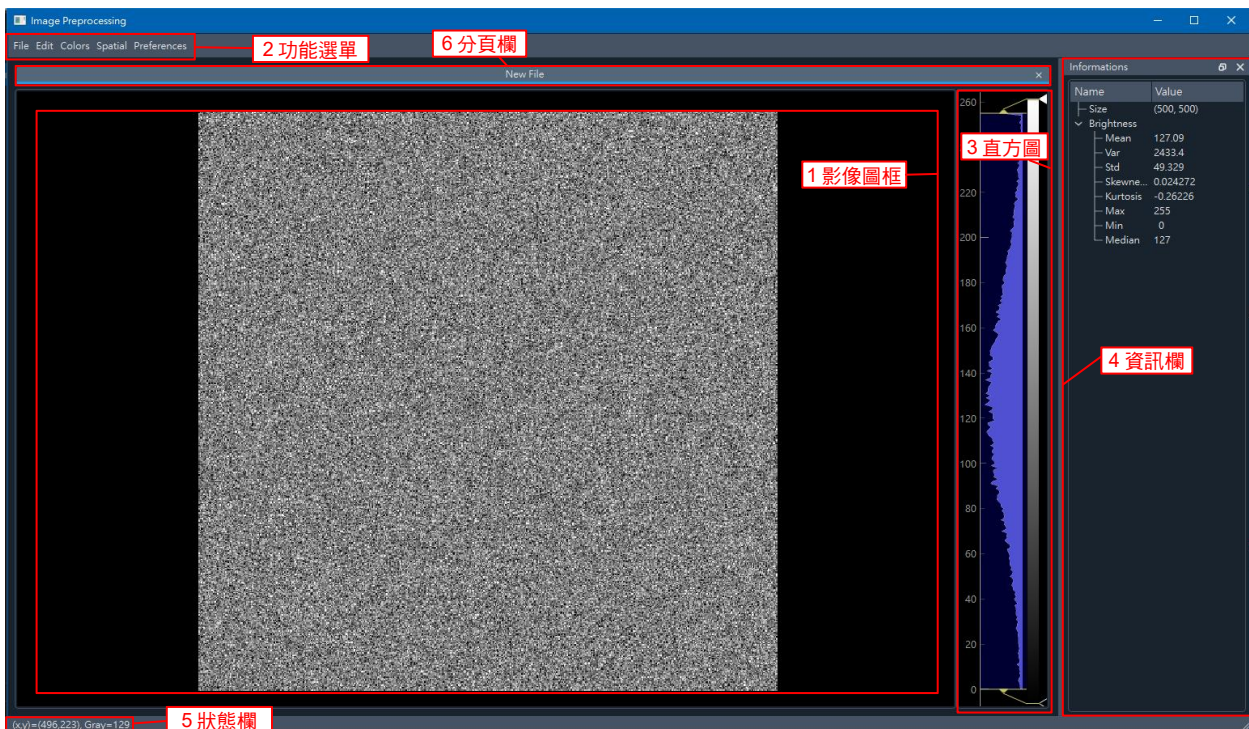


圖 1 主視窗區域介紹

1. 影像圖框(self.__image_view)。未讀取影像時展示一張高 480 寬 640 之高斯雜訊，程式碼於圖 2。
__image_view 以及 __histogram 之設定位於 __create_main_image_view。

2. 功能選單(self.__menu)。

3. 直方圖(self.__histogram)。彩色影像分別顯示 RGB，如圖 3（上）。程式碼如圖 4。

4. 資訊欄(self.__info_frame, self.__info_model)，影像亮度之統計資訊。

Preferences>Stats 可關閉/開啟（快捷鍵 F12）。更新之程式碼如圖 5、圖 6。

5. 狀態欄(self.__status_bar)，顯示鼠標所在之影像座標及數值，於 mainwindow.py 之 mouse_moved 中更新。彩色影像如圖 3（下）所示。

6. 分頁欄(self.__tab_bar)，未讀取影像時隱藏。

```
402 def update_all(self, img: Optional[Union[IMG_8U, Image]] = None):
403     """Update image view and information tree view."""
404     if img is None:
405         if not self.__img_objects:
406             img = np.random.normal(127.5, 50, (500, 500)).astype(np.uint8)
407         else:
408             img = self.__img_objects[self.__tab_index]
409     self.__update_information_tree(img)
410     if isinstance(img, Image):
411         img = img.image
412     self.__update_view(img)
```

圖 2 預設影像程式碼(第 406 行)，於 mainwindow.py。

預設放置圖片用以展示直方圖以及資訊欄，若刪除需額外撰寫程式碼，在關閉所有分頁時調整圖框、直方圖及資訊欄。

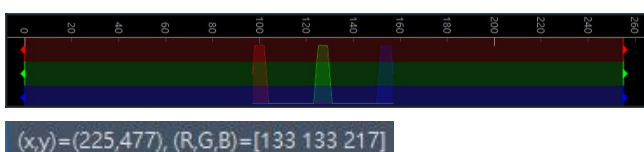


圖 3 彩色影像顯示之直方圖（上）與狀態欄（下）。

主視窗

```
413 def __update_view(self, img: IMG_8U):
414     self.__image_view.setImage(img, levels=(0, 255))
415     # setImage will update levelMode before updating image. Hence,
416     # setImage(img, levels=(0, 255), levelMode="rgba") raise error when
417     # substituting color image for grayscale image.
418     # 'setLevelMode' should be called individually.
419     if img.ndim == 2:
420         self.__histogram.setLevelMode("mono")
421     else:
422         self.__histogram.setLevelMode("rgba")
423     self.__histogram.vb.autoRange()
```

圖 4 影像更新程式碼，於 mainwindow.py。若不想將 RGB 分開顯示，則可刪除紅框範圍。

```
39 stats_labels = (
40     "Mean", "Var", "Std", "Skewness", "Kurtosis",
41     "Max", "Min", "Median"
42 )
43
44 def get_statistics(
45     img: ARR_8U2D
46 ) -> Tuple[MEAN, VAR, STD, SKEW, KURTOSIS, MAX, MIN, MEDIAN]:
47     # Order statistic
48     max_, min_ = stats.max_min(img)
49     median_ = stats.median(img)
50     # Moments
51     mean_, var_, skew, kurtosis = stats.moments(img)
52     std_ = np.sqrt(var_)
53     return mean_, var_, std_, skew, kurtosis, max_, min_, median_
```

圖 5 資訊欄標籤與統計資訊，於 image_object.py。stats_labels 可更改標籤順序，需同時更改 get_statistics 回傳順序。

```
422 def __update_information_tree(self, img: Union[IMG_8U, Image]):
423     if isinstance(img, Image):
424         # The Image object will compute statistics when Image.image be
425         # updated.
426         vals = img.brightness_statistics
427         img = img.image
428         self.__info_model["size"].setText(f"{img.shape}")
429     else:
430         self.__info_model["size"].setText(f"{img.shape}")
431         if img.ndim == 3:
432             img: ARR_8U2D = cv2.cvtColor(img, cv2.COLOR_RGB2GRAY)
433             vals = Image.get_statistics(img)
434
435     labels = BasicOperators.stats_labels
436     # Standardize float value length.
437     vals = [
438         # Max, Min, and Median are uint8 numbers.
439         f"{v:.5g}" if index < 5 else str(v)
440         for index, v in enumerate(vals)
441     ]
442     for label, val in zip(labels, vals):
443         self.__info_model[label].setText(val)
```

圖 6 資訊欄更新程式碼，於 mainwindow.py。紅框處為顯示之字串格式。彩色影像則轉換為灰階計算。

File 選單

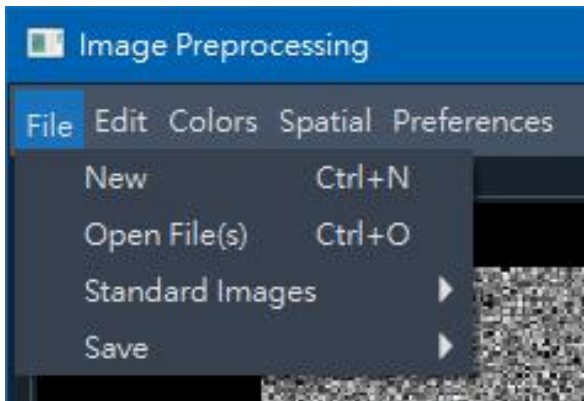


圖 7 File 選單功能。

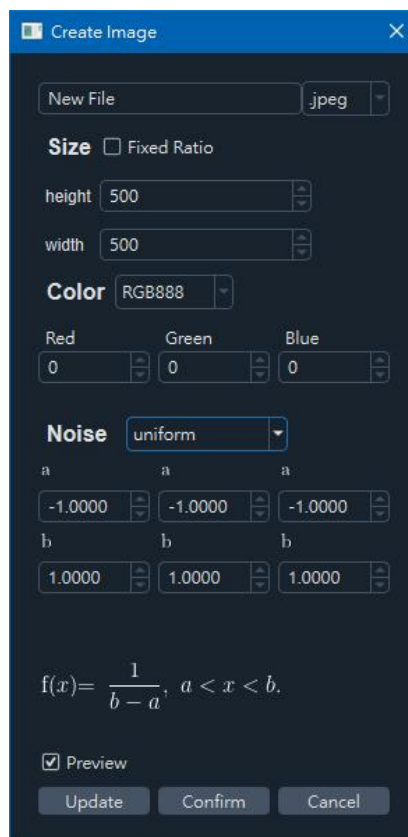
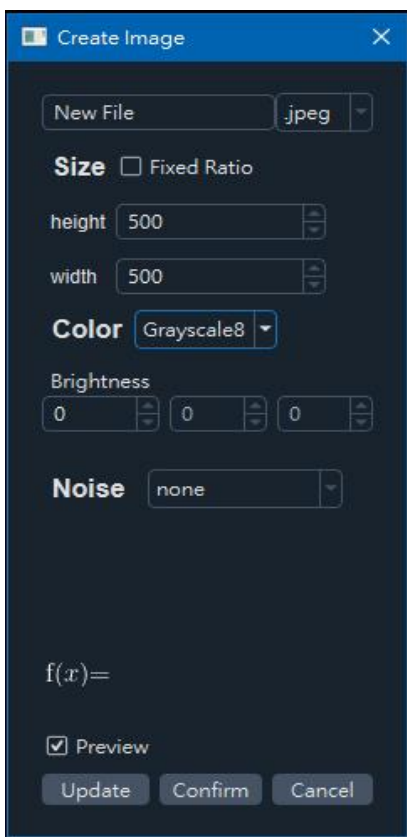


圖 8 新增影像(New)，快捷鍵 Ctrl+N。

Fixed Ratio：固定寬高比。

Color：彩色(RGB888)、灰階(Grayscale8)。皆為 uin8 陣列。預設為 RGB。

Noise：增加雜訊。支援之雜訊分布有無雜訊、uniform, gaussian, rayleigh, gamma, exponential, salt-and-pepper, beta。預設為無雜訊(none)。

下方有雜訊分布公式，使用 matplotlib 的 Tex 繪製。

File 選單

```
565 support_file_extensions = (  
566     "*.jpeg", "*.jpg", "*.jpe", # JPEG  
567     "*.jp2", # JPEG 2000  
568     "*.png", # PNG (Portable Network Graphics)  
569     # "*.tiff", "*.tif", # TIFF  
570     "*.webp", # WebP  
571     # "*.avif", # AVIF  
572     # "*.pbm", "*.pgm", "*.ppm", "*.pxm", "*.pnm", # Portable image format  
573     # "*.pfm", # PFM  
574     # "*.hdr", "*.pic" # Radiance HDR  
575 )
```

圖 9 讀取/儲存支援之副檔名，隱藏部分亦為 OpenCV 支援之格式。於 image_object.py 的 Image class 中。

```
577 support_database = (  
578     "astronaut", "brick", "camera", "cat", "cell", "checkerboard",  
579     "clock", "coffee", "coins", "colorwheel", "grass",  
580     "gravel", "horse", "hubble_deep_field", "immunohistochemistry",  
581     "logo", "microaneurysms", "moon", "page", "retina", "rocket",  
582     "shepp_logan_phantom", "text"  
583 )
```

圖 10 Standard Images 內容，部分的 scikit-image 資料。於 image_object.py 的 Image class 中。

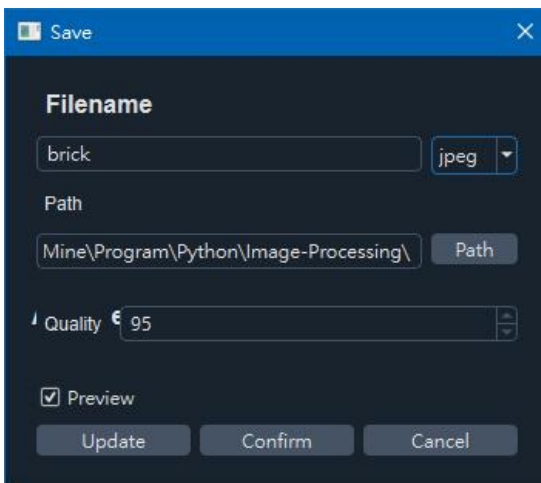


圖 11 儲存影像，預設 Ctrl+S。

Edit 選單

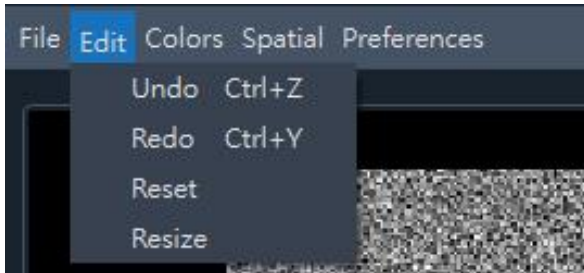


圖 12 Edit 選單。

預設記錄長度為 20 張圖，可於 Image class 中修改 (image_object.py)。

Image 額外保存原始圖片，Reset 會「更新」至原始圖片(加入 history 中)。

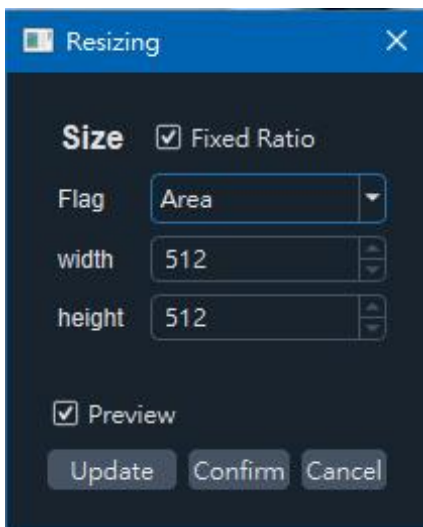


圖 13 影像縮放。

Flag：縮放之演算法。OpenCV 提供之方法: Nearest, Linear, Cubic, Area, Lanczos4。

Colors 選單

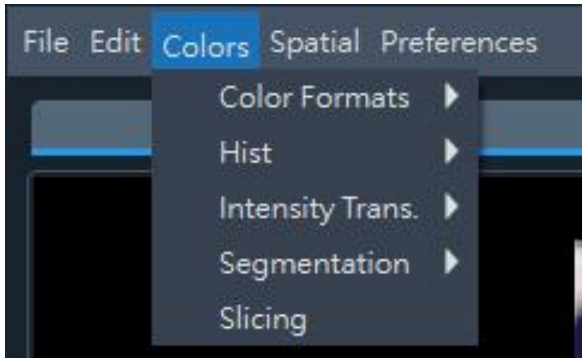


圖 14 Colors 選單。

Color Formats：可轉換影像為黑白或彩色。

Hist：直方圖方法，包含 CLAHE、Histogram Equalization 以及 Histogram Matching。

Intensity Trans.：強度轉換，彩色影像為 RGB 個別使用。包含 Auto Gamma Correction、Function Trans.以及 Piecewise Linear Trans.。

Auto Gamma Correction：選取 γ 使得亮度平均值轉換後為 127.5。

Segmentation：影像分割，包含 k-means。

Slicing：切割出指定顏色及半徑內之色彩，距離計算方法有 L1（絕對值相加）、L2(歐氏距離)以及 sup-norm（最大的絕對值）。對黑白影像，與 Function Trans 的 Level Slicing Type II(Inv)相同（圖 15）。

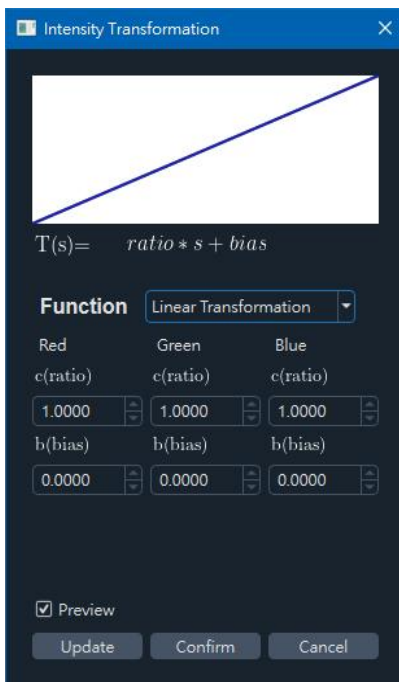


圖 15 Function Trans，透過特定函數轉換強度。各轉換函數 $f(s)$ 之公式如下

Linear Transformation：

$$ratio * s + bias$$

Gamma Correction：

$$\begin{cases} s, \gamma = 1; \\ 255^{1-\gamma} * s^\gamma + bias, \gamma \neq 1 \text{ and } ratio = 0; \\ ratio * s^\gamma + bias, \gamma \neq 1 \text{ and } ratio \neq 0; \end{cases}$$

Log Transformation：

$$\begin{cases} \frac{255 \ln(1+s)}{\ln(256)} + bias, ratio = 0; \\ ratio * \ln(1 + s) + bias, otherwise. \end{cases}$$

Arctan Transformation：

$$c * \arctan(g * s - b) + d, \text{ where } g = \frac{\gamma}{100}, b = center * g, c = \frac{255}{\arctan(255 * g - b) + \arctan(b)}, d = c * \arctan(b)$$

Logistic Correction：

$$\frac{255(P(s) - P(0))}{P(255) - P(0)}, P(s) = \frac{1}{1 + e^{-(s-center)/\sigma^2}}$$

Beta Correction：

$$\frac{255\Gamma(a)\Gamma(b)}{\Gamma(a+b)} \int_0^{s/255} t^{a-1}(1-t)^{b-1} dt$$

Level Slicing Type I：指定強度範圍為前景，其餘為背景。

$$\begin{cases} fg, low < s < high; \\ bg, otherwise. \end{cases}$$

Level Slicing Type II：凸顯指定強度範圍，其餘保留。

$$\begin{cases} level, low < s < high; \\ s, otherwise. \end{cases}$$

Level Slicing Type II(Inv)：保留指定強度範圍，凸顯其餘。

$$\begin{cases} s, low < s < high; \\ level, otherwise. \end{cases}$$

Colors 選單

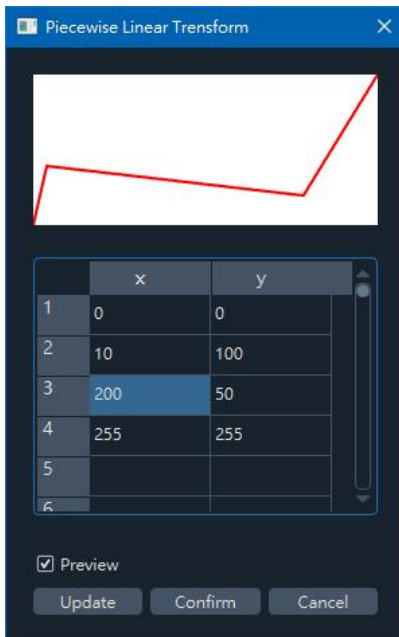


圖 16 Piecewise Linear Trans，構造 Piecewise Linear function 進行強度轉換。輸入(x,y)後，會依照 x 大小進行排序。若 x,y 僅輸入一項，不會進行更新。

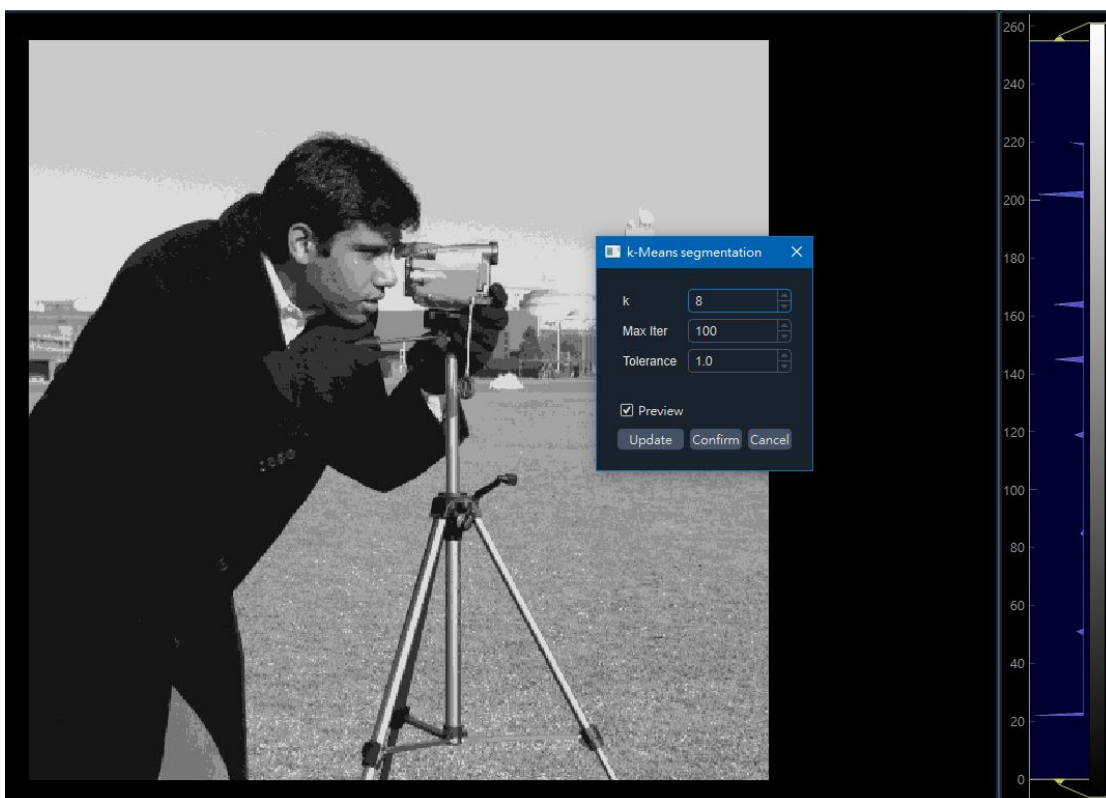


圖 17 k-means 色彩分割。

k：分割的顏色總數。

Max Iter：最大迭代次數。

Tolerance：兩次迭代變化小於此值，則停止迭代。

Spatial 選單

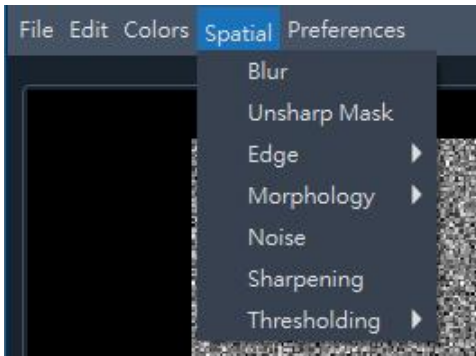


圖 18 Spatial 選單。

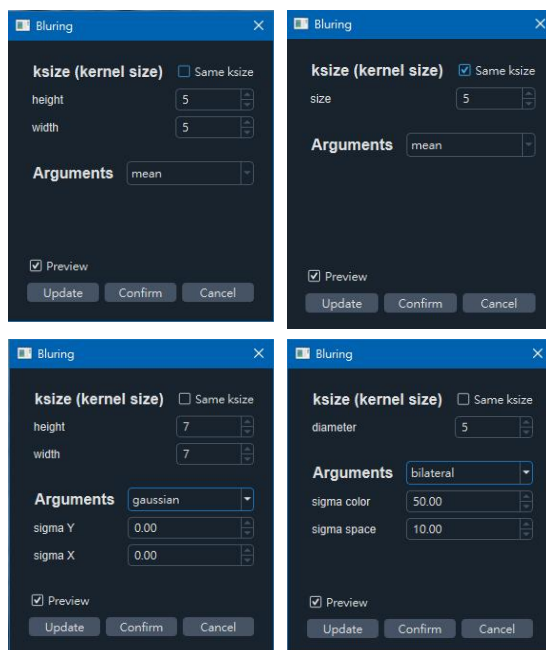


圖 19 模糊化(Blur)，包含 mean blur、gaussian blur、median blur 以及 bilateral blur。

Spatial_filter.py 中，另外還有 max blur、min blur、midpoint blur、geometric mean blur、harmonic mean blur、contraharmonic mean blur、alpha-trimmed mean、adaptive mean blur 以及 adaptive median blur。

Same ksize：mean blur 或 gaussian blur 的 kernel 長寬相等。

sigma Y、sigma X：高斯核的分散程度。

sigma color、sigma space：鄰近點的色彩（或空間）距離的影響力。詳細可參考 OpenCV 之 [bilateral blur](#) 條目。

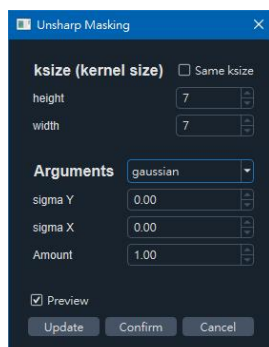


圖 20 非銳化遮罩(Unsharp masking, USM)，透過影像與其模糊化之差找出邊緣，並以此強化影像邊緣：

$$USM = img + amount * (img - blur(img))$$

提供的模糊化方法於圖 19 註解。

amount：邊緣增幅的程度。通常在 1 與 4.5 之間。

Spatial 選單

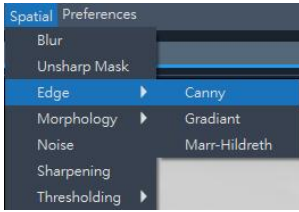


圖 21 邊緣檢測方法，包含 Canny、Marr-Hildreth 以及測邊算子等。



圖 22 Canny 算子。

Blur Args：高斯模糊參數。

Threshold、Threshold 2：低門檻及高門檻。低於低門檻者非邊緣，大於高門檻者為邊緣，兩者之間則取決於此點有無與大於高門檻的點連接。

Aperture Size：Sobel 算子大小，僅能輸入 3、5、7。

L2 Gradient：梯度計算為 L1 或 L2。

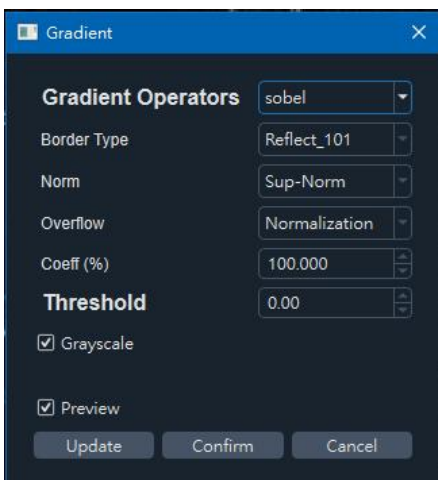


圖 23 常見的測邊算子，包含 laplacians, laplacians_all, laplacian2, laplacian4, sobel, prewitt, robinson, roberts, kirsch, scharr, LoG。拉普拉斯算子(laplacians, laplacians_all, laplacian2, laplacian4)之差異見圖 24。

Norm：除 laplacian2、laplacian4 及 LoG 外，其餘皆包含多個方向的檢測，以 norm 的方式合併多張邊緣。包含 sup-norm（絕對值得最大值）、Abs. Sum（即 L1）、Euclidean Norm 以及 Abs. Mean（L1 除以算子數量）。

Overflow：經過 norm 合併後，範圍可能超出[0, 255]，須處理至此範圍中。方法包含 clip(大於 255 的值修剪為 255)、透過下列公式進行 Normalization：

$$f(s) = \begin{cases} 255, & s > p * max + (1 - p) * min; \\ \frac{255(s-min)}{p(max-min)}, & \text{elsewise.} \end{cases}$$

其中 $p = \text{coeff}/100$ 。直觀上來說，可視為將大於 $p * max + (1 - p) * min$ 的值拉至 255，min 拉至 0，中間以直線連接。亦或視為 min-max normalization 縮放至[0, 255]後，乘上一個範圍在[1, ∞)的常數再 clip 至[0, 255]。因 norm 處理後的範圍可能很大，直接以 min-max normalization 縮放至[0, 255]可能使大部分明顯的邊緣落在中明度，乘上一個大於 1 的常數可使這些邊緣更明顯。

Threshold：在處理 overflow 之後進行，將低於門檻的值設為 0。

Norm 之公式：

Sup-norm	$\max_i \text{grad}_i $
Abs. sum	$\sum_i \text{grad}_i$
Euclidean norm	$\sum_i \text{grad}_i^2$
Abs. mean	$\frac{1}{\max_i} \sum_i \text{grad}_i $

Spatial 選單

```

961 # Laplacian operator
962 if grad_name == "laplacians":
963     kernel = [
964         # Y
965         [[ 1], [-2], [ 1]],
966         # X
967         [[1, -2,  1]],
968     ]
969 elif grad_name == "laplacians_all":
970     kernel = [
971         # Y
972         [[ 1], [-2], [ 1]],
973         # X
974         [[1, -2,  1]],
975         # NW
976         [[1,  0,  0],
977          [0, -2,  0],
978          [0,  0,  1]],
979         # NE
980         [[0,  0,  1],
981          [0, -2,  0],
982          [1,  0,  0]],
983     ]

```

```

984 elif grad_name == "laplacian2": # X,Y Direction
985     kernel = [
986         [[0,  1,  0],
987          [1, -4,  1],
988          [0,  1,  0]]
989     ]
990 elif grad_name == "laplacian4": # All Direction
991     kernel = [
992         [[1,  1,  1],
993          [1, -8,  1],
994          [1,  1,  1]]
995     ]

```

圖 24 拉普拉斯算子差別。laplacians 及 laplacians_all 分別為 2 個和 4 個不同方向的算子；laplacian2 及 laplacian4 為一個算子。

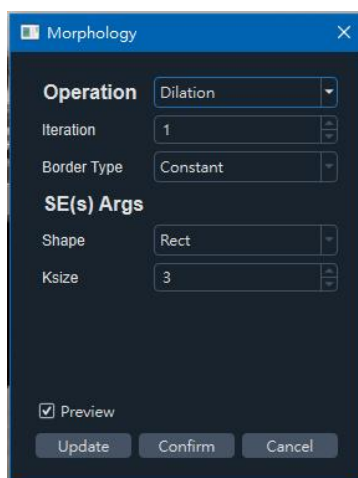


圖 25 形態學運算，非 OpenCV 提供之函數位於 morphology.py 中，包含

Dilation, Erosion, Opening, Closing, Gradient, Top Hat, Black Hat, Hit-or-Miss	由 OpenCV 直接提供。
Pruning	圖 26
Connect Components	圖 27
Thinning	圖 28
Thickening	
Skelton	
Region Filling	圖 29
Border Clean	圖 30
Boundary Extraction	圖 31
Opening by Reconstruction	
Closing by Reconstruction	

Skelton 僅可用於 binary image（僅有黑與白的影像）。

Spatial 選單



圖 26 Pruning (剪除) 範例，左圖為剪除前，右圖為剪除後。剪除線條上的突起。

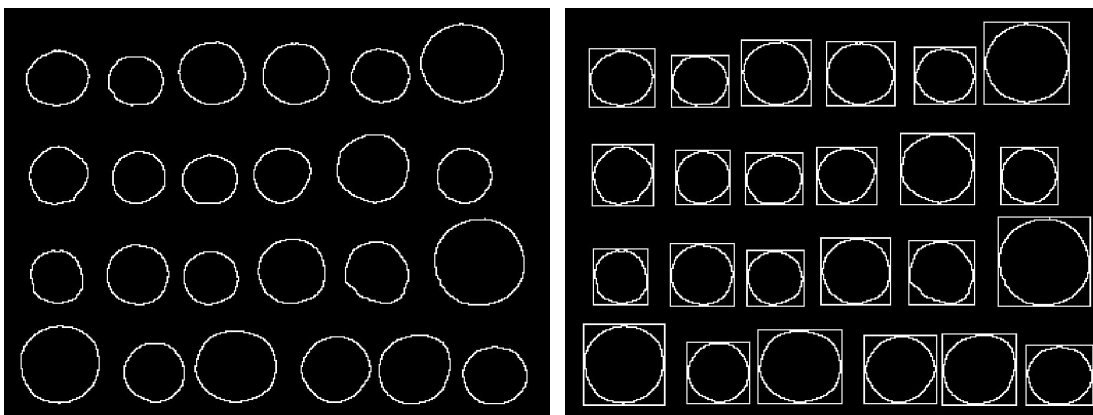


圖 27 Connect Components (連通成分) 範例，框取所有連通成分。

iteration：選取的點進行擴張(dilation)的次數。1 表示選取一個點並擴張一次，此範圍與原圖的交集，即視為一個連通成分。0 表示擴張至無法再擴張，如上圖右。

Spatial 選單

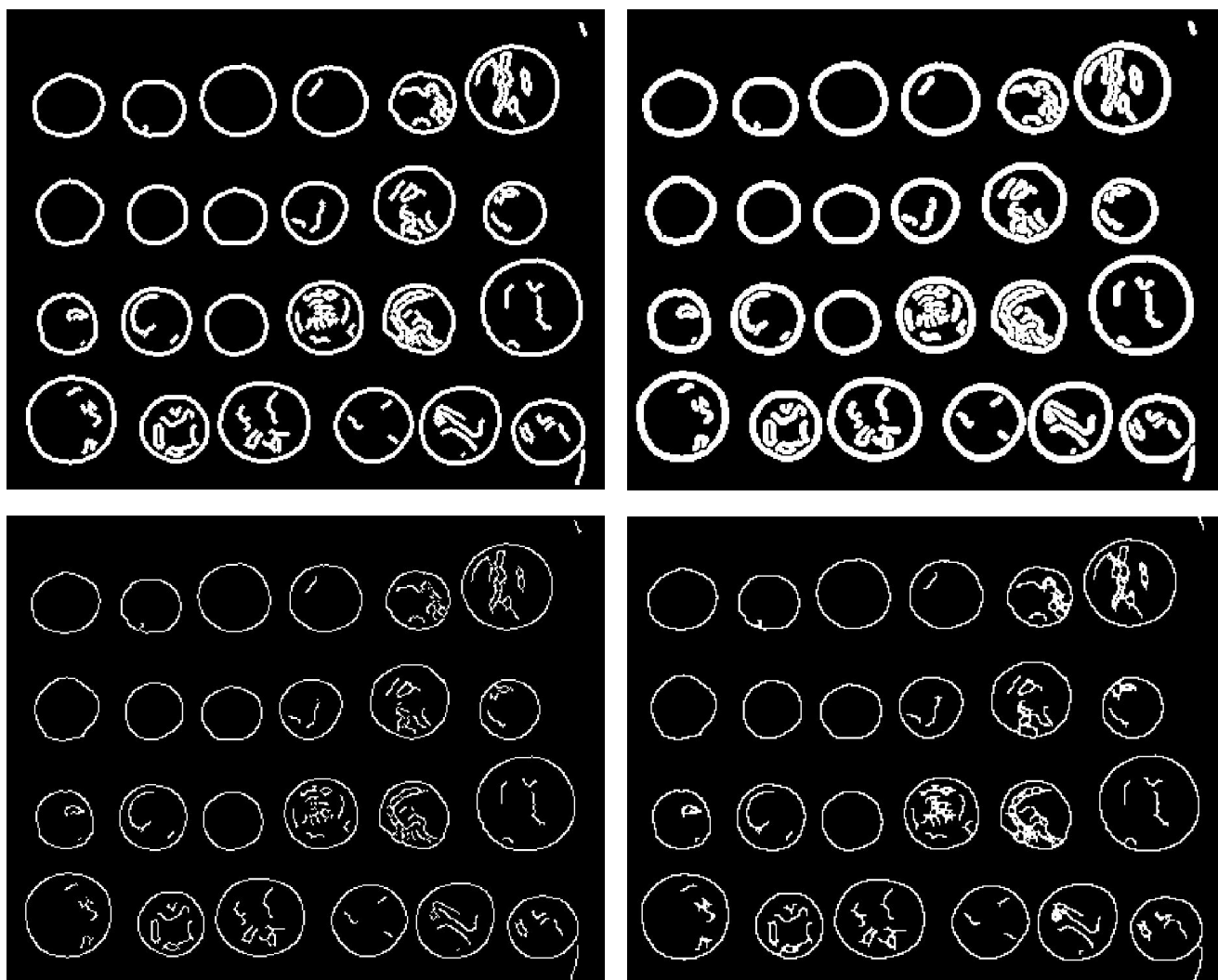


圖 28 左上：原圖。右上：Thickening(厚化)。左下：Thinning(細線化)。右下：Skelton (骨架化)。

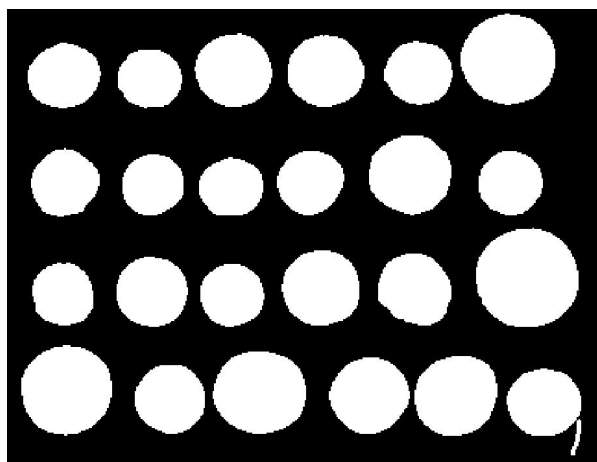


圖 29 Region Filling，類似油漆桶功能，自動填充所有封閉區域。判斷「封閉」與 SE 形狀有關，以圖 28 左下來說，因邊緣太薄，Rect 形狀的 SE 可能判斷某些圓圈不算封閉，但 Cross 形狀的 SE 則會判斷此區域為封閉。

Spatial 選單



圖 30 Border clean，清除接觸到圖片邊界(border)的連通物件。

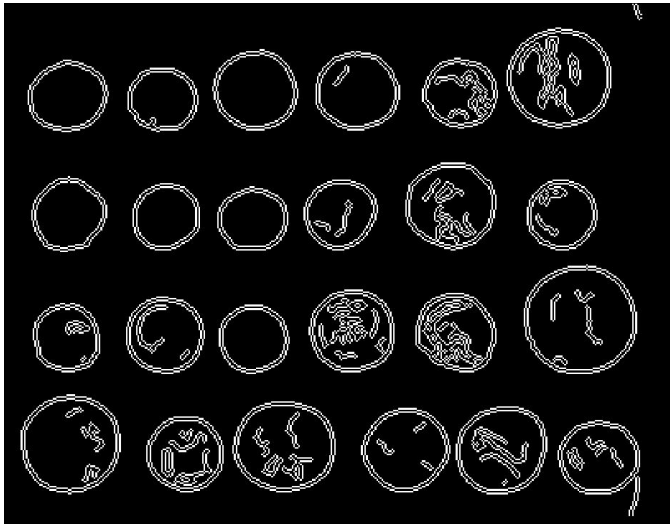


圖 31 Boundary extraction（邊界萃取），效果類似 gradient operator。

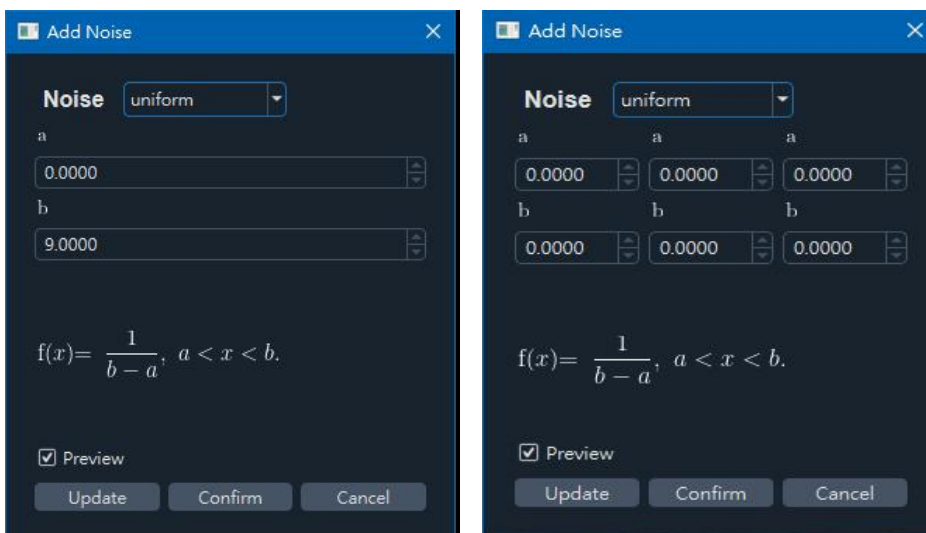


圖 31 增加雜訊。支援之雜訊分布有 uniform, gaussian, rayleigh, gamma, exponential, salt-and-pepper, beta。預設為 uniform。

Spatial 選單

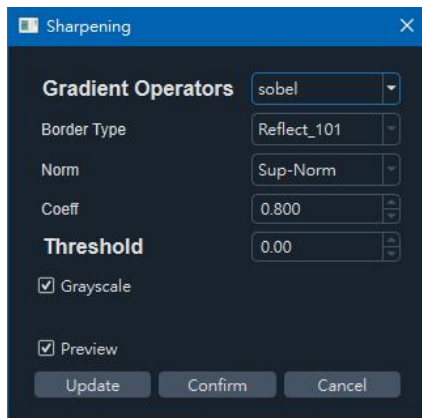


圖 32 Sharpening（銳利化），使用梯度算子進行銳利化：

$$\text{sharpened} = \text{img} + \text{coeff} * \text{grad}$$

norm：非數學上的 norm，僅以此詞稱呼。與 gradient(圖 23)中類似，合併多張梯度結果，但會保留正負。

Sup-norm	$\text{grad}_k, \text{ where } k = \arg \max_i \text{grad}_i $
Abs. sum	$\sum_i \text{grad}_i$
Euclidean norm	$\text{sign}(\text{grad}_k) \sum_i \text{grad}_i^2, \text{ where } k = \arg \max_i \text{grad}_i $
Abs. mean	$\frac{1}{\max_i} \sum_i \text{grad}_i$

coeff：邊緣增幅的程度。

Grayscale：按鈕僅在彩色圖片中出現。轉為灰階計算梯度，再分別對 RGB 三通道銳利化，因此結果仍是彩色。

Spatial 選單

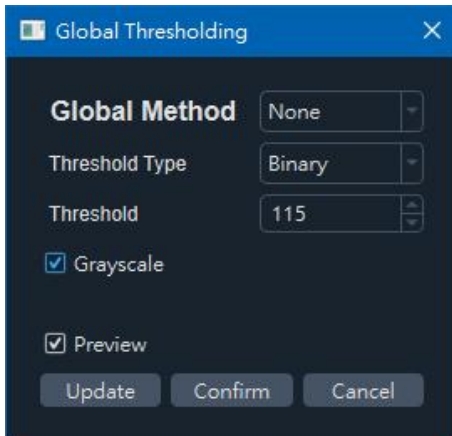


圖 33 Global thresholding（全域門檻化）。

Global method：自動選取門檻值的方法

名稱	公式或演算法（Threshold value, t） 符號： $\text{mean}_g = \text{mean}(\text{img})$ 、 p_k 為強度 k 出現的機率、 $S(k) = \sum_{i=0}^k p_i$ 、 $m_n(k) = \sum_{i=0}^k i^n \cdot p_i$ 、 $E(k) = -\sum_{i=0}^k p_i \log p_i$ 、 $Q(k) = \sum_{i=0}^k p_i^2$
None	手動選取
mean	$t = \text{mean}(\text{img})$
itermean	1.初始化 $k \in [0, 255]$ 2.計算 $c_1 = \text{mean}(\text{img}[\text{img} \leq k])$, $c_2 = \text{mean}(\text{img}[\text{img} > k])$ 3.若 $k = \frac{c_1 + c_2}{2}$ 則設 $t = k$ 。否則設 $k = \frac{c_1 + c_2}{2}$ ，並回到步驟 2。
otsu	$t = \underset{k}{\text{argmax}} \frac{[\text{mean}_g \cdot S(k) - m_1(k)]^2}{S(k)(1 - S(k))}$
max_entropy	$t = \underset{k}{\text{argmax}} \log [S(k)(1 - S(k))] + \frac{E(k)}{S(k)} + \frac{E(255) - E(k - 1)}{1 - S(k)}$
yen	$t = \underset{k}{\text{argmax}} 2 * \log [S(k)(1 - S(k))] - \log [Q(k)(Q(255) - Q(k))]$
moments	$X_1 = \frac{m_1(255)m_3(255) - [m_2(255)]^2}{m_2(255) - [m_1(255)]^2}$ $X_2 = \frac{m_1(255)m_2(255) - m_3(255)}{m_2(255) - [m_1(255)]^2}$ $X_0 = 0.5 - \frac{m_1(255) + X_2/2}{\sqrt{X_2^2 - 4 * X_1}}$ $t = \underset{k}{\text{argmin}} k - X_0 $

Spatial 選單

Threshold type :

名稱	公式 (Threshold value, t)
Binary	$f(s) = \begin{cases} 255, & s > t; \\ 0, & \text{elsewise.} \end{cases}$
Binary Inverse	$f(s) = \begin{cases} 0, & s > t; \\ 255, & \text{elsewise.} \end{cases}$
Trunc	$f(s) = \begin{cases} t, & s > t; \\ s, & \text{elsewise.} \end{cases}$
To Zero	$f(s) = \begin{cases} s, & s > t; \\ 0, & \text{elsewise.} \end{cases}$
To Zero Inverse	$f(s) = \begin{cases} 0, & s > t; \\ s, & \text{elsewise.} \end{cases}$

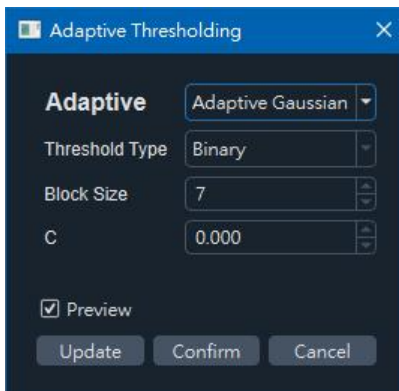


圖 34 Local thresholding。依據像素周圍的資訊進行門檻化支援的方法有 Adaptive Mean, Adaptive Gaussian, Adaptive Median, niBlack, Sauvola, Wolf, Nick。

Threshold type : 支援 Binary 和 Binary Inverse。

Block size : 進行參考的範圍大小。

C : 將計算出的門檻值位移, $t_{\text{final}} = t_{\text{local}} - C$ 。 t_{final} 為最終的門檻值、 t_{local} 為依據周圍資訊計算的門檻值。