Computer Vision HW2 楊閔喻 R09921012

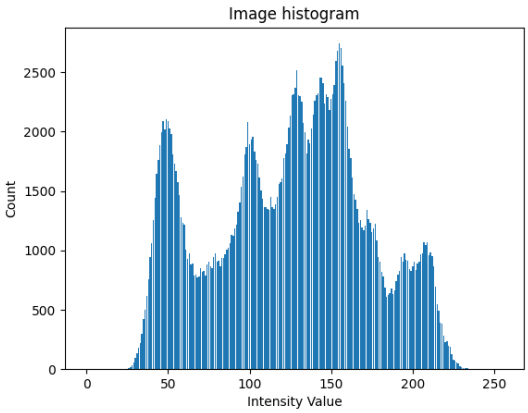
Write a program to generate:

(a) a binary image (threshold at 128)
(b) a histogram
(c) connected components(regions with + at centroid,bounding box)

(a)(b)

```
for i in range(img.shape[0]):
    for j in range(img.shape[1]):
        (b,g,r) = img[i,j]
        histogram[b] = histogram[b]+1
        if b >= 128 :
            img[i,j] = (255,255,255)
            img2[i,j] = 1
        else :
            img[i,j] = (0,0,0)
```

- 走訪每個 pixel，取得其 Intensity Value，如大於 threadhold 就將其設定為 255，反之則設定為 0
- 走訪每個 pixel，取得其 Intensity Value，紀錄於 histogram 變數中



(c)

```
count = 0
for i in range(img2.shape[0]):
    for j in range(img2.shape[1]):
        if img2[i][j] > 0:
            count = count+1
            img2[i][j] = count
```

- 初始化連通元件 array 中每個大於 threadhold pixel 的編碼值，以遞增的形式

```
while change:
    time = time+1
    print("TIME ",time)
    change = 0
    #up-down
    for i in range(img2.shape[0]):
        for j in range(img2.shape[1]):
            if img2[i,j] != 0 :
                smallest = img2[i,j]
                if i-1 >=0 :
                    if img2[i-1,j] != 0 and img2[i-1,j]<smallest :
                        smallest = img2[i-1,j]
                if j-1>=0 :
                    if img2[i,j-1] != 0 and img2[i,j-1]<smallest :
                        smallest = img2[i,j-1]

                if smallest != img2[i,j]:
                    img2[i,j] = smallest
                    change = True
    #down->up
    for i in range(img2.shape[0]-1,-1,-1):
        for j in range(img2.shape[1]-1,-1,-1):
            if img2[i,j] != 0 :  #以下略
```

- 利用 Iterative Algorithm 進行連通元件判斷，採用四連通為判斷基準
- 迭代直到某次迭代的結果並無發生任何改變時，lena 共進行 20 次迭代

```
area = np.zeros(count+1,dtype=int)
for i in range(img2.shape[0]) :
```

```
    for j in range(img2.shape[1]) :
        if img2[i,j] != 0 :
            area[img2[i,j]] = area[img2[i,j]]+1
```

- 計算各個編號所佔有的 pixel 數，計算重心使用

```
for c in range(area.shape[0]):
    if area[c] >= 500 :
        #connected_count = connected_count+1
        hight = -1
        low = img.shape[0]
        left = -1
        right = img.shape[1]
        center_row = 0
        center_col = 0
        for i in range(img2.shape[0]):
            for j in range(img2.shape[1]):
                if img2[i,j] == c:
                    center_row = center_row+i
                    center_col = center_col+j
                    if i>hight:
                        hight = i
                    if i<low:
                        low = i
                    if j>left:
                        left = j
                    if j<right:
                        right = j
```

- 存取 pixel 數大於 500 的編號，記錄其連通元件在上下左右的邊界極值

```
        center_row = int(center_row/area[c])
        center_col = int(center_col/area[c])
        cv.rectangle(img,(right,low),(left,hight),(255,0,0),2)
        cv.drawMarker(img,(center_col,center_row),(0,0,255),markerType=
cv.MARKER_CROSS,markerSize=20,thickness=2)
```

- Call library 將重心與 bounding box 繪出