Homework 2

R05922002 廖建棋

撰寫程式所使用的程式語言是python 3.5，需先安裝scipy, numpy, Pillow(PIL), matplotlib套件。scipy+Pillow是負責讀寫圖片；numpy是做影像陣列的資料結構；matplotlib是繪製圖表。

Threshold at 128

Threshold在128作法只需要判斷每個pixel是不是超過threshold t=128，如果是，則將此pixel改為白色(255)，不是的話，就設為黑色(新開的空白陣列全為0，因此不需做此步)：

```
if image[y, x] >= t:
    thresholded_img[y, x] = 255
```

結果

Histogram

首先，需開一個陣列做pixel intensity的統計，intensity是從0~255，給每一個intensity一個bin統計落入的pixel數：

```
statistic = np.zeros(256)
```

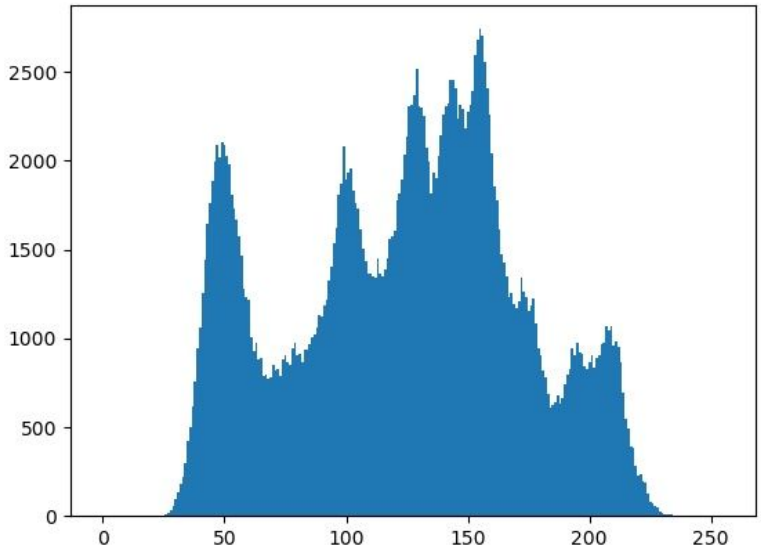
接著，掃過所有pixels，並以其intensity當作index選擇bin，向上計數：

statistic[image[y, x]] += 1

最後用matplotlib的bar函數繪製長條圖(不能用hist函數，因為它會自動幫我們做像素統計，因此即失去這題的意義)，並設定直條寬度為1：

plt.bar(range(0, 256), statistic, width=1)

結果



Connected Components

本題是參考上課教的iterative algorithm實作的，透過參數可以選擇是4連通或是8連通，由此參數選擇適當的找鄰近label函數，由於forward pass和backword pass的鄰近點不同，因此分成兩個函數：

forward_neighbor_4_labels, backward_neighbor_4_labels

forward_neighbor_8_labels, backward_neighbor_8_labels

在做掃描填label的核心程式碼是這段，和投影片的pseudocode雷同：

```
if yx_label > 0:
    labels = forward_neighbor_labels(image, y, x)
    labels.append(yx_label)
    min_label = min(labels)
    if min_label != yx_label:
        image[y, x] = min_label
        changed = True
```

標記結束後，將標記後的影像回傳，並進行畫框和標中心點，是box_components(img, labeled_img)這個函數。這支函數一開始會用一個table紀錄有那些components，其相關資訊會用一個ComponentInfo的物件紀錄下來，其資訊分為：component的邊界x, y座標(min_x, min_y, max_x, max_y)、pixel數量。這些資訊透過update這個方法掃描component時更新的：

```
con_comps.setdefault(labeled_img[y, x], ComponentInfo()).update(y, x)
```

接著，依題目要求，要把少於500 pixels的component捨棄，因此，透過filter函數先過濾掉這些不合格的component：

```
components = filter(lambda comp: comp.pixel_num >= 500, con_comps.values())
```

有了component邊界座標資訊後，迭代每個component，計算中心點畫十字。畫十字時，是將原影像重疊到的像素由白變黑由黑變白，這樣看得比較明顯：

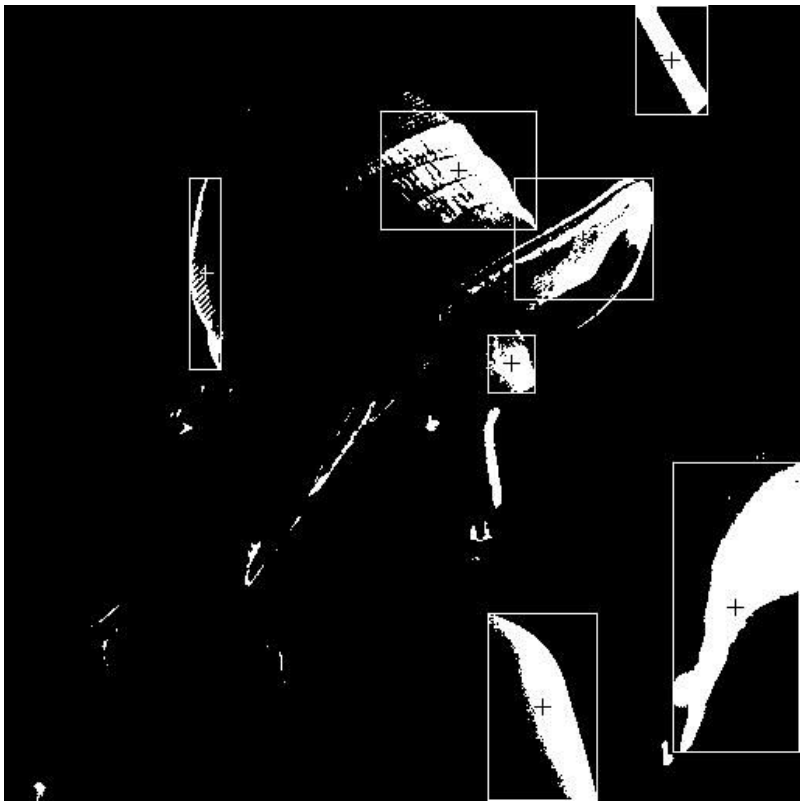
```
for y in range(cent_y - CROSS_HALF_SIZE, cent_y + CROSS_HALF_SIZE + 1):
    if y >= 0 and y < img.shape[0]:
        boxed_img[y, cent_x] = 0 if img[y, cent_x] > 0 else 255

for x in range(cent_x - CROSS_HALF_SIZE, cent_x + CROSS_HALF_SIZE + 1):
    if x >= 0 and x < img.shape[1]:
        boxed_img[cent_y, x] = 0 if img[cent_y, x] > 0 else 255
```

接著是畫框框，框框即根據邊界座標位置，繪製四條白線即可：

```
    for y in range(comp.min_y, comp.max_y + 1):
        boxed_img[y, comp.max_x] = 255
    for y in range(comp.min_y, comp.max_y + 1):
        boxed_img[y, comp.min_x] = 255
    for x in range(comp.min_x, comp.max_x + 1):
        boxed_img[comp.max_y, x] = 255
    for x in range(comp.min_x, comp.max_x + 1):
        boxed_img[comp.min_y, x] = 255
```

結果

執行時大約要花個數秒的迭代處理，下圖是以8連通、threshold為200二值化的lena圖：



4連通(和8連通相差不大):