電腦視覺作業二

網媒所碩一 呂佳晏 R11944045

---

part (a) a binary image (threshold at 128)

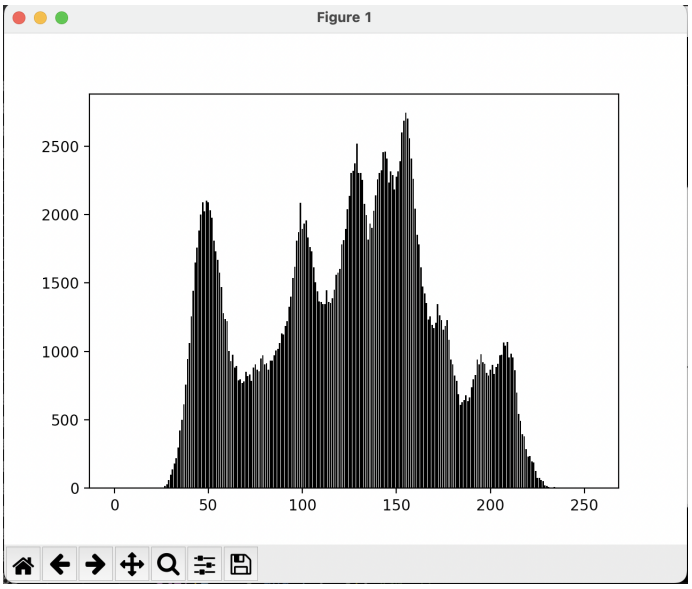
我的作法:檢查每個 pixel ，若大於等於閾值的話設成白色，否則設成黑色

```
#(a) a binary image (threshold at 128)
def image_bin(image_to_process, value):
    row = image_to_process.shape[0] #image 高度
    col = image_to_process.shape[1] #image 寬度
    for i in range (row):
        for j in range(col):
            for k in range(3):
                if image_to_process[i, j, k] >= value:
                    image_to_process[i, j, k] = 255 #設成白色
                else:
                    image_to_process[i, j, k] = 0 #設成黑色
    return image_to_process
```

part (b) a histogram

我的作法, 計算每個點的值存在 list 裏面，再轉換成 histogram

```
#(b) a histogram
def hist(image_to_process):
    hist = [0] * 256
    row = image_to_process.shape[0] #image 高度
    col = image_to_process.shape[1] #image 寬度
    for i in range (row):
        for j in range(col):
            hist[image_to_process[i, j, 0]] += 1
    #print(hist)
    plt.bar(range(0, 256), hist, color = 'black')
    plt.show()
    return
```



** 下一頁還有

part (c) connected components (regions with + at centroid, bounding box)

我的作法:

第一步 row by row 先將二值化後的圖建立連通部位，並幫他們標記 id

我用的是四聯通的方式

```
#(c) connected components use (a) result
image3 = image_bin
row = image3.shape[0] #image 高度
col = image3.shape[1] #image 寬度

label_id = np.zeros((row, col), dtype = int) #紀錄label id
label_cnt = 1
for i in range(row):
    for j in range(col):
        if image3[i, j, 0] != 0:
            if(i == 0 and j == 0): #原點
                label_id[i, j] = label_cnt
                label_cnt += 1
            elif(i == 0 and j != 0): #第一行
                if(label_id[i, j - 1]) != 0:
                    label_id[i, j] = label_id[i, j - 1]
                else:
                    label_id[i, j] = label_cnt
                    label_cnt += 1
            elif(i != 0 and j == 0): #第一列
                if(label_id[i - 1, j]) != 0:
                    label_id[i, j] = label_id[i - 1, j]
                else:
                    label_id[i, j] = label_cnt
                    label_cnt += 1
            else: #其他點
                if(label_id[i - 1, j] == 0 and label_id[i, j - 1] != 0):#左邊不為0
                    label_id[i, j] = label_id[i, j - 1]
                elif(label_id[i - 1, j] != 0 and label_id[i, j - 1] == 0):#上面不為0
                    label_id[i, j] = label_id[i - 1, j]
                elif(label_id[i - 1, j] != 0 and label_id[i, j - 1] != 0): #上左都不為零
                    if(label_id[i - 1, j] == label_id[i, j - 1]): #上左相同
                        label_id[i, j] = label_id[i, j - 1]
                    else: #上左不相同
                        color_left = label_id[i, j - 1]
                        color_up = label_id[i - 1, j]
                        label_id[i, j] = color_left #此點先設成跟左邊一樣
                        #更新讓上面的點和左邊的點相同
                        for a in range(row):
                            for b in range(col):
                                if(label_id[a, b] == color_up):
                                    label_id[a, b] = color_left
                else: #上左都沒有相鄰的點
                    label_id[i, j] = label_cnt
                    label_cnt += 1
```

第二步是找出面積大於 500 的連通部位來建立 bounding box 和 畫重心

```
threshold = 500
#計算大於 500 的面積有哪些 label_cnt
area_cnt = np.zeros(row * col, dtype = int)
for i in range(row):
    for j in range(col):
          area_cnt[label_id[i, j]] += 1
area = [] #有五個 label_id 數量超過 500
for i in range(1, row * col):
    if(area_cnt[i] > 500):
        area.append(i)

for i in area:
    drew_retangle(i)
```

最後一步就是計算 bounding box 的邊界還有 計算相同 label id 之重心

```
#(c) connected components use (a) result
def drew_retangle(id):
    down = 0
    right = 0
    top = row
    left = col
    cen_i = []
    cen_j = []
    for i in range(row):
        for j in range(col):
            if(label_id[i, j] == id):
                cen_i.append(i)
                cen_j.append(j)
                if(i < top):
                    top = i
                elif(i > down):
                    down = i
                if(j < left):
                    left = j
                elif(j > right):
                    right = j
    cv.rectangle(image3, (left, top), (right, down), (0, 255, 0), 2) #draw retangle
    #畫重心
    center_i = sum(cen_i) / len(cen_i)
    center_j = sum(cen_j) / len(cen_j)
    center_i = int(center_i)
    center_j = int(center_j)
    cv.line(image3, (center_j - 8, center_i), (center_j + 8, center_i), (0, 255, 0), 2) #橫線
    cv.line(image3, (center_j, center_i - 8), (center_j, center_i + 8), (0, 255, 0), 2) #直線
```