資工所碩一 R10922123 周昱豪

Code

Part a:

抓出每個 pixel value，如果 < 128 就設為 0，否則設為 255。

```
def binary_image_threshold(img, threshold):
    for c in range(img.width):
        for r in range(img.height):
            values = img.getpixel((c, r))
            if(values < threshold):
                values = 0
            else:
                values = 255
            img.putpixel((c, r), values)
    img.save("lena_binary.bmp")
    return img
```

Part b:

建一個 size = 256 的 list，index 表示 pixel value，list[index]表示這張 image 有幾個這個 pixel 的個數

```
def image_histogram(img):
    hist = [0 for _ in range(256)]
    for c in range(img.width):
        for r in range(img.height):
            values = img.getpixel((c, r))
            hist[values] += 1

    histFile = open('lena_hist.csv', "w")
    wri = csv.writer(histFile)
    wri.writerow(hist)
    histFile.close()

    x = np.arange(len(hist))
    plt.bar(x, hist)
    plt.xlim(0, 256)
    plt.savefig('lena_hist.png')
    plt.show()
```

Part C: (8-connected)

利用 DFS，創一個 visited list 來儲存已被拜訪過的 pixel，如果沒被 visit 過就 push 進去 stack，反正就是 DFS 然後 stack 空的時候代表那些剛剛 visit 的位置都是連通的。

```
64      for c in range(width):
65          for r in range(height):
66              if binary_img.getpixel((c, r)) == 0:
67                  visited[c, r] = 1
68              elif visited[c, r] == 0:
69                  stack = Stack()
70                  stack.push((c, r))
71                  while not stack.isEmpty():
72                      col, row = stack.pop()
73
74                      if visited[col, row] == 1:
75                          continue
76                      visited[col, row] = 1
77                      label_image_array[col, row] = region_cnt
78
79                      number_of_label[region_cnt] = number_of_label[region_cnt] + 1
80
81                      for x in [col - 1, col, col + 1]:
82                          for y in [row - 1, row, row + 1]:
83                              if (0 <= x < width) and (0 <= y < height):
84                                  if (binary_img.getpixel((x, y)) != 0) and (visited[x, y] == 0):
85                                      stack.push((x, y))
86                  region_cnt += 1
87
```

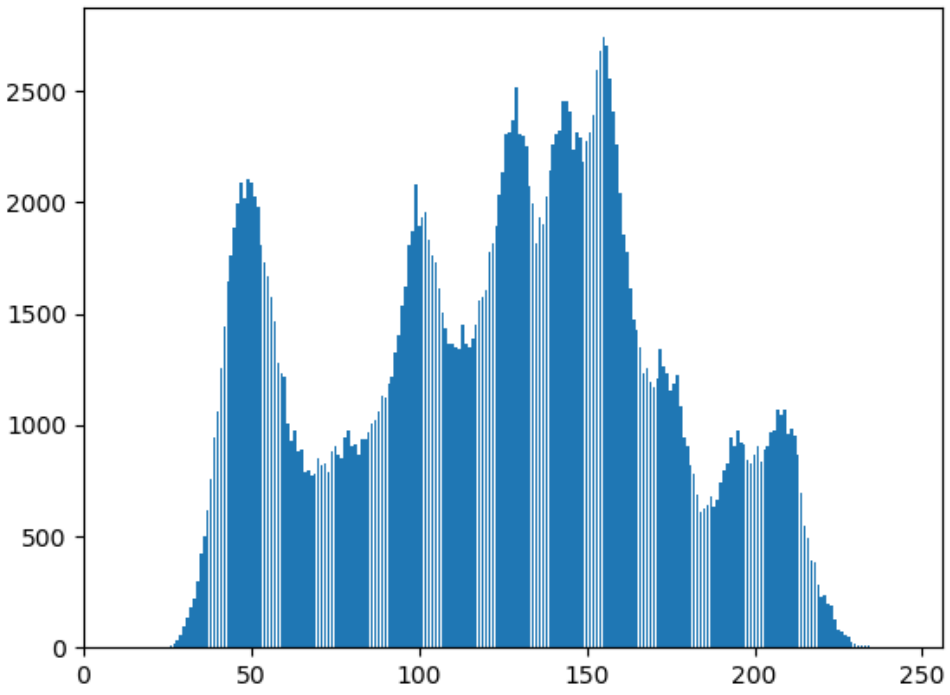
Bounding box 就是去找每個上面找出來的 region 的邊界，那在 line99~112 找，順便算 centroid 需要的 value

```
90      for regionID, n in enumerate(number_of_label):
91          if n >= region_threshold:
92              rect_left = width
93              rect_right = 0
94              rect_top = height
95              rect_bottom = 0
96              sum_c = 0
97              sum_r = 0
98              cnt = 0
99              for c in range(width):
100                 for r in range(height):
101                     if (label_image_array[c, r] == regionID):
102                         sum_c += c
103                         sum_r += r
104                         cnt += 1
105                         if (c < rect_left):
106                             rect_left = c
107                         if (c > rect_right):
108                             rect_right = c
109                         if (r < rect_top):
110                             rect_top = r
111                         if (r > rect_bottom):
112                             rect_bottom = r
113             # Push rectangle's information to stack.
114             rect.append([rect_left, rect_right, rect_top, rect_bottom])
115             centroid.append([int(sum_c/cnt), int(sum_r/cnt)])
116
```

Result:

Part a:

Part b:



Part C: