

a. Binary image (threshold = 128)

先將照片利用 ndarray 讀取

設閾值為 128，判斷像素是否大於這閾值，回傳布林值

布林值 True 為 1，Fales 為 0，乘以 255 會以黑(0)白(255)顯示

```
def binarize(img):  
    im = np.array(img)  
    th = 128  
    im_bin_128 = (im > th) * 255  
    result = Image.fromarray(np.uint8(im_bin_128))  
    result.show()  
    return result
```

結果



b. Histogram

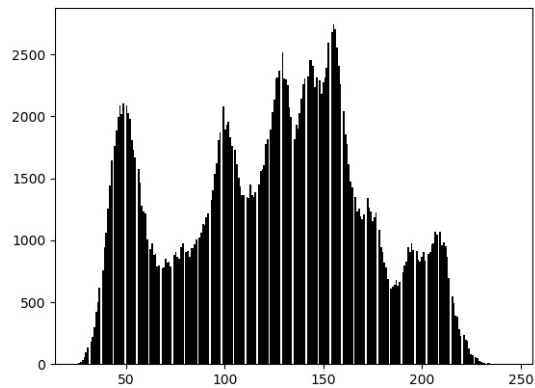
先將照片利用 ndarray 讀取

再利用 flatten 的方法將該 ndarray 變為一維陣列

最後利用 matplotlib.pyplot 繪製出直方圖

```
def histogram(img):  
    im = np.array(img)  
    arr = im.flatten()  
  
    plt.hist(arr, bins=256, facecolor='black')  
    plt.savefig("lena_histogram.jpg")  
    plt.show()  
    return 0
```

結果



c. Connected component (regions with + at centroid, bounding box)

利用 Two-pass 演算法進行四連通去串接物件

演算法詳細過程利用 common 寫在程式碼中

最後得到每個 label 的陣列，其中包含每個像素(x,y)

```
def connected_components(img_bin):
    labels = []
    pixels_label = [[-1] * img_bin.width for h in range(img_bin.height)]

    for y in range(img_bin.height):
        for x in range(img_bin.width):

            if img_bin.getpixel((x, y)) != 255:
                continue

            result_label = -1
            #1 先看候選像素點左方 (y,x-1)
            if x > 0 and pixels_label[y][x-1] != -1:
                result_label = pixels_label[y][x-1]

            #2 再看候選像素點上方 (y-1, x)
            if y > 0 and pixels_label[y-1][x] != -1:
                result_up = pixels_label[y-1][x]

            if result_label != -1 and result_label != result_up: #若上述符合執行合併
                for _x, _y in labels[result_label]:
                    pixels_label[_y][_x] = result_up #將左邊那個變為最小值
                labels[result_up] += labels[result_label] #合併為同個list
                labels[result_label] = None #刪除label較大的list

            result_label = result_up

            #3 看候選像素點是否有標記數字
            if result_label == -1:
                result_label = len(labels) #從0開始 賦予一個新的標記數字
                labels.append([(x, y)])

            else:
                labels[result_label].append((x, y)) #同個標記數字放同個list

            pixels_label[y][x] = result_label

    return labels
```

要過濾 pixels 少於 500 的 label

接著利用 OpenCV 的函示畫出矩形和重心

```
img_bin = Image.open("lena_binarize.bmp") #讀入th = 128 的檔案
img_new = cv2.imread("lena_binarize.bmp") #用cv2畫出矩形和重心
for component in connected_components(img_bin):
    if type(component) != list: #有些為None
        continue
    if len(component) < 500: # 只取面積 > 500
        continue

    (left, top), (right, bottom) = component[0], component[0]
    centroid_x = 0
    centroid_y = 0

    for x, y in component: #找到左上和右下的點
        if x < left:
            left = x
        if x > right:
            right = x
        if y < top:
            top = y
        if y > bottom:
            bottom = y

        centroid_x += x
        centroid_y += y

    C_x = centroid_x / len(component)
    C_y = centroid_y / len(component)
    cv2.rectangle(img_new, (left, top), (right, bottom), (0,255,0))
    cv2.circle(img_new, (int(C_x), int(C_y)), 5, (0,0,255), -1)
```

結果

