

# Computer Vision

## Part 1: Generate a binary image (threshold at 128)

一開始讀入lena.bmp時，就以灰階圖的方式讀入，再把陣列中大於等於128的值改成255，小於128的部分則改成0。

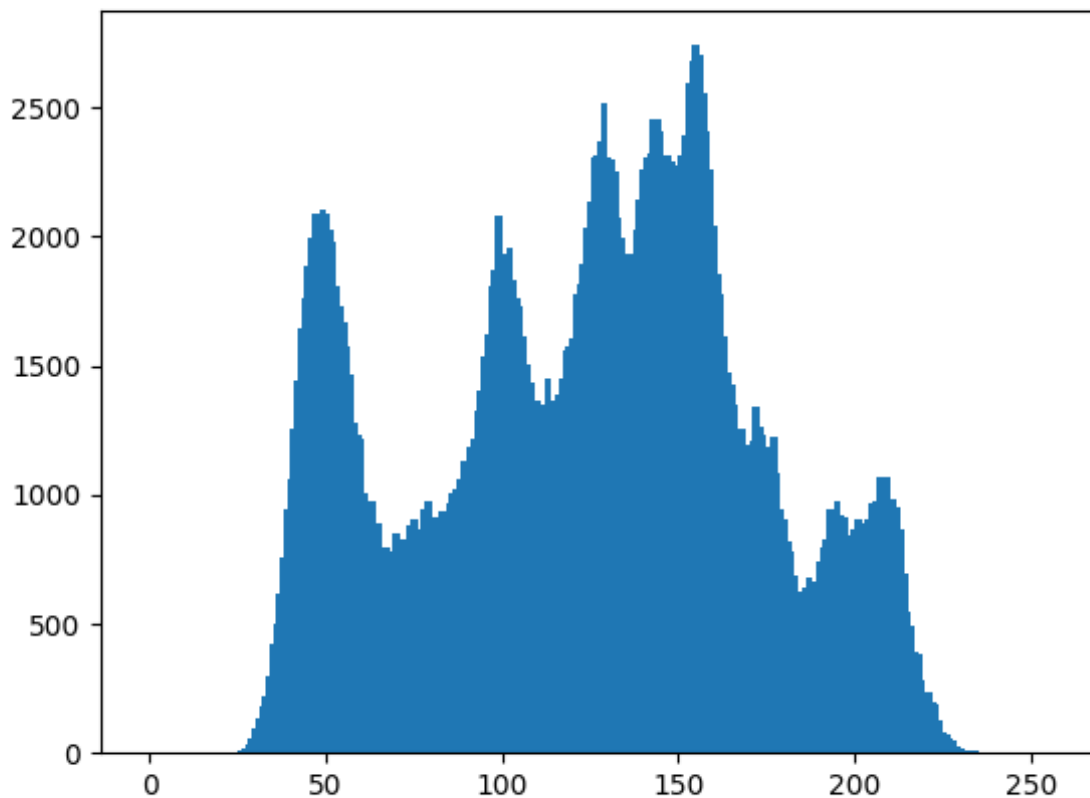
```
def threshold(img):  
    ans = np.zeros((img.shape), np.int)  
    for x in range(img.shape[0]):  
        for y in range(img.shape[1]):  
            if img[x,y] >= 128:  
                ans[x,y] = 255  
            else:  
                ans[x,y] = 0  
    return ans
```



## Part 2: Generate a histogram

創造一個256的一維陣列儲存grayscale value 0~255個別的數量，再利用兩層for迴圈計算數量並存於一維陣列當中，最後使用matplotlib中的plt.bar畫出圖像。

```
def histogram(img):  
    ans = np.zeros(256, np.int)  
    for x in range(img.shape[1]):  
        for y in range(img.shape[0]):  
            ans[img[x,y]] += 1  
    plt.bar(range(len(ans)), ans, width = 2.0)  
    plt.savefig("histogram.png")
```



### Part 3: Connected components

這個部分我是使用4-connected來判斷是否相鄰。用一個stack把與其相連的pixel放進去，直到連再一起的部分全部處理好（同時更新左界右界上界下界），再去看另一塊。每當有超過500pixel連在一起的部分時，我再用cv2.rectangle去把邊框畫出來，然後再用cv2.circle畫出質心。函式會把放兩張圖進去的原因是，原本是用灰階去讀入影像，但是這樣就不能畫出有顏色的邊框及質心。

```

def connect(img, img2):
    binary = threshold(img)
    final = img2.copy().astype(np.int32)
    visit = np.zeros(img.shape, np.int)
    for x in range(img.shape[0]):
        for y in range(img.shape[1]):
            cnt, cntc = 0, 0
            L, R, U, B, cnt = maxi, -maxi, maxi, -maxi, 0
            st = [(x, y)]
            while st:
                r, c = st.pop()
                if 512 > r >= 0 and 512 > c >= 0 and visit[r, c] == 0 and binary[r, c] != 0:
                    visit[r, c] = 1
                    cnt += 1
                    cntc += c
                    L = min(L, c)
                    R = max(R, c)
                    U = min(U, r)
                    B = max(B, r)
                    st.extend([(r, c + 1), (r, c - 1), (r + 1, c), (r - 1, c)])
            if cnt >= 500:
                cv2.rectangle(final, (L, B), (R, U), (255, 0, 0), 2)
                cv2.circle(final, (int(cntc / cnt), int(cnt / cnt)), 4, (0, 0, 255), -1)
    return final

```

