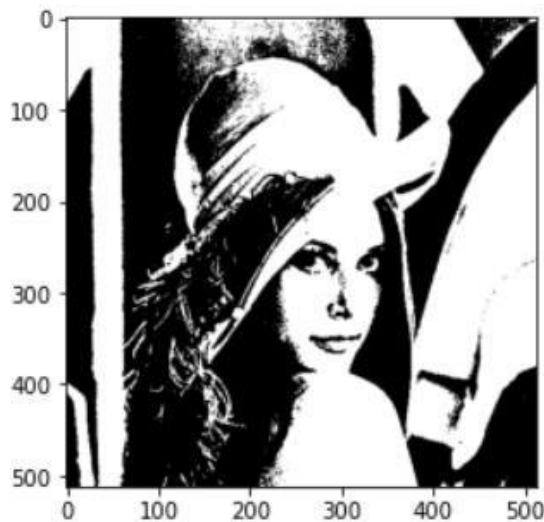


Computer Vision HW2

(a) a binary image (threshold at 128)

x,y 分別為圖片的長寬，之後就可以從 0 開始走當他大於等於 128 的時候就把它換成 255(白)，小於時就換成 0(黑)

```
for i in range(0, x, 1):
    for j in range(0, y, 1):
        if (lena[i][j] >= 128):
            lena_binary[i][j] = 255
        else:
            lena_binary[i][j] = 0
```

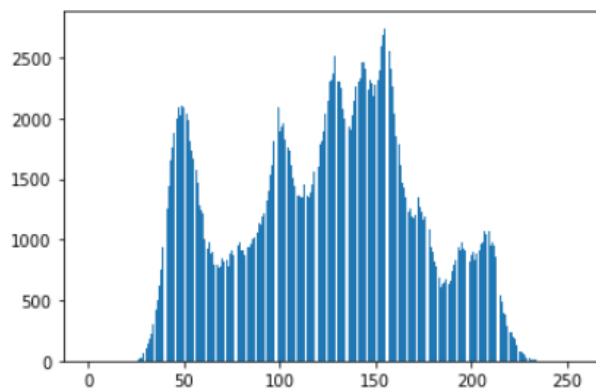


(b) a histogram

把每一個值放進histogram 的陣列裡，再做成直方圖即可

```
for i in range(0, x, 1):
    for j in range(0, y, 1):
        histogram[lena[i][j]] += 1

cv2.imwrite('histogram.jpg', histogram)
plt.bar(range(len(histogram)), histogram)
```



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

```
#threshold 500
threshold = 500

#file
lena = cv2.imread('C:/Users/kaiyuchou/test_data/lena.bmp', cv2.IMREAD_GRAYSCALE)
lena_binary = cv2.imread('lena_binary.bmp', cv2.IMREAD_GRAYSCALE)

#size
sum1 = lena.shape[0] #寬尺寸
sum2 = lena.shape[1] #長尺寸

num = np.zeros(lena.shape, int)
labelArray = np.zeros(lena.shape, int)
Count = 1
#紀錄有多少pixel
numberLabel = np.zeros(x * y)

for i in range(sum1):
    for j in range(sum2):
        #如過pixel是0的話就標視為到過
        if lena_binary.getpixel((i, j)) == 0:
            num[i, j] = 1
        elif num[i, j] == 0:
            stack = Stack()
            #把位置放進去
            stack.push((i, j))
            while not stack.isEmpty():
                #如過不是空的就把他pop出來
                col, row = stack.pop()

                if num[col, row] == 1:
                    continue
                num[col, row] = 1
                labelArray[col, row] = Count
                #看有多少pixel在這個標示裡
                numberLabel[Count] = numberLabel[Count] + 1
                #確認周圍的
                for a in [col - 1, col, col + 1]:
                    for b in [row - 1, row, row + 1]:
                        #當她在Lena裡
                        if (0 <= a < sum1) and (0 <= b < sum2):
                            #此位置不是0且沒訪問過
                            if (lena_binary.getpixel((a, b)) != 0) and (num[a, b] == 0):
                                stack.push((a, b))

            Count += 1
```

```

rectangles = Stack()
#處理pixel500以上的區域
for regionID, n in enumerate(numberLabel):
    if (n >= threshold):
        rectLeft = sum1
        rectRight = 0
        rectTop = sum2
        rectBottom = 0
        for x in range(sum1):
            for y in range(sum2):
                #從陣列裡搜尋最小的
                if (labeledArray[x, y] == regionID):
                    if (x < rectLeft):
                        rectLeft = x
                    if (x > rectRight):
                        rectRight = x
                    if (y < rectTop):
                        rectTop = y
                    if (y > rectBottom):
                        rectBottom = y

        rectangles.push((rectLeft, rectRight, rectTop, rectBottom))
#Create 新的圖但是是RGB
connectedImage = Image.new('RGB', lena.size)
connectedImageArray = connectedImage.load()

#一個pixel一個pixel將binary image 轉為 RGB
for c in range(width):
    for r in range(height):
        if (lena_binary.getpixel((c, r)) == 0):
            connectedImageArray[c, r] = (0, 0, 0)
        else:
            connectedImageArray[c, r] = (255, 255, 255)

#劃出範圍跟十字
while not rectangles.isEmpty():
    #要標出的範圍的資訊
    rectLeft, rectRight, rectTop, rectBottom = rectangles.pop()
    #畫的圖
    draw = ImageDraw.Draw(connectedImage)
    draw.rectangle((rectLeft, rectTop), (rectRight, rectBottom)), outline = 'white')
    #標出的範圍的中心點
    rectCenterX = (rectLeft + rectRight) / 2
    rectCenterY = (rectTop + rectBottom) / 2
    #垂直及水平的十字線

```

