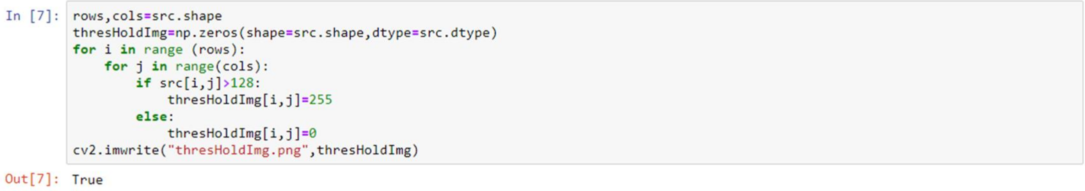
Computer Vision HW#3

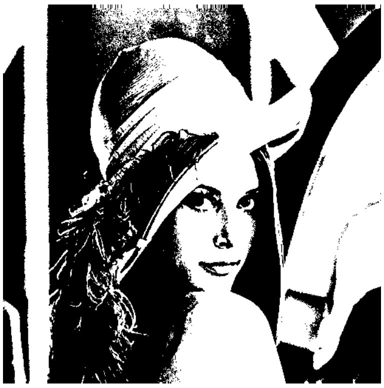
R08942125 廖克允

Part 1.

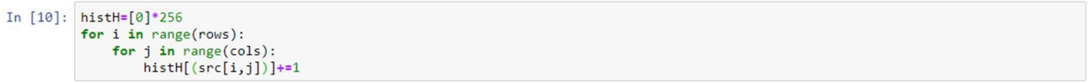
(a) a binary image (threshold at 128)



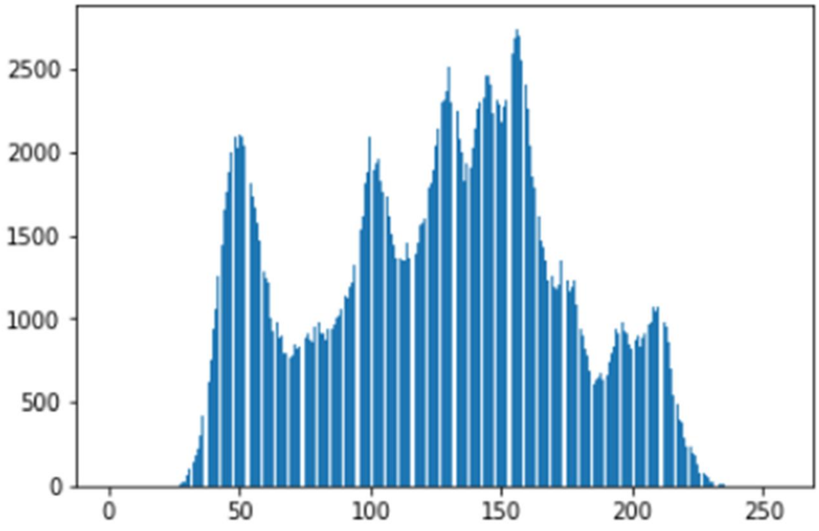
```
In [7]: rows,cols=src.shape
        thresHoldImg=np.zeros(shape=src.shape,dtype=src.dtype)
        for i in range (rows):
            for j in range(cols):
                if src[i,j]>128:
                    thresHoldImg[i,j]=255
                else:
                    thresHoldImg[i,j]=0
        cv2.imwrite("thresHoldImg.png",thresHoldImg)

Out[7]: True
```



(b) a histogram



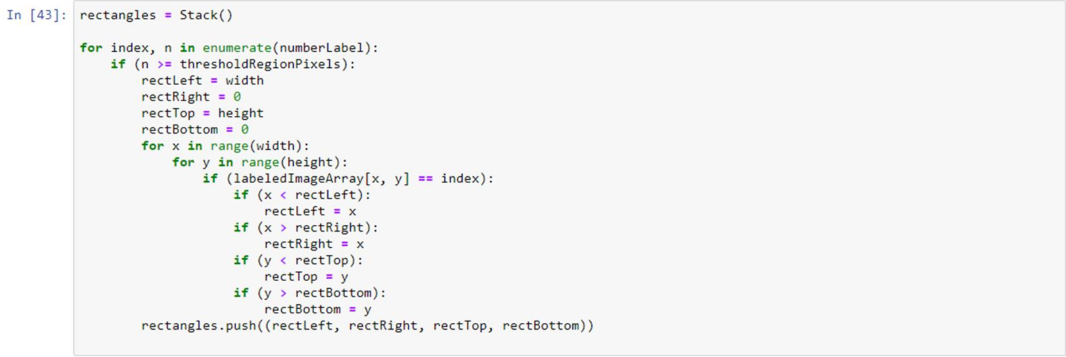
```
In [10]: histH=[0]*256
         for i in range(rows):
             for j in range(cols):
                 histH[(src[i,j])]+=1
```



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

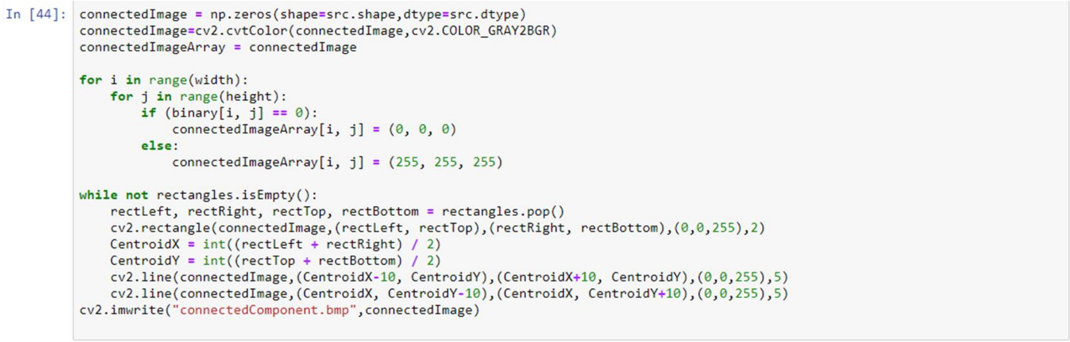
By using 8-connected, set threshold to 500, push rectangle's info into stack. And then draw the rectangles and crosses on new image

Push rectangle's info into stack



```
In [43]: rectangles = Stack()

         for index, n in enumerate(numberLabel):
             if (n >= thresholdRegionPixels):
                 rectLeft = width
                 rectRight = 0
                 rectTop = height
                 rectBottom = 0
                 for x in range(width):
                     for y in range(height):
                         if (labeledImageArray[x, y] == index):
                             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))
```

Draw the bounding box and centroids



```
In [44]: connectedImage = np.zeros(shape=src.shape,dtype=src.dtype)
         connectedImage=cv2.cvtColor(connectedImage,cv2.COLOR_GRAY2BGR)
         connectedImageArray = connectedImage

         for i in range(width):
             for j in range(height):
                 if (binary[i, j] == 0):
                     connectedImageArray[i, j] = (0, 0, 0)
                 else:
                     connectedImageArray[i, j] = (255, 255, 255)

         while not rectangles.isEmpty():
             rectLeft, rectRight, rectTop, rectBottom = rectangles.pop()
             cv2.rectangle(connectedImage,(rectLeft, rectTop),(rectRight, rectBottom),(0,0,255),2)
             CentroidX = int((rectLeft + rectRight) / 2)
             CentroidY = int((rectTop + rectBottom) / 2)
             cv2.line(connectedImage,(CentroidX-10, CentroidY),(CentroidX+10, CentroidY),(0,0,255),5)
             cv2.line(connectedImage,(CentroidX, CentroidY-10),(CentroidX, CentroidY+10),(0,0,255),5)
         cv2.imwrite("connectedComponent.bmp",connectedImage)
```