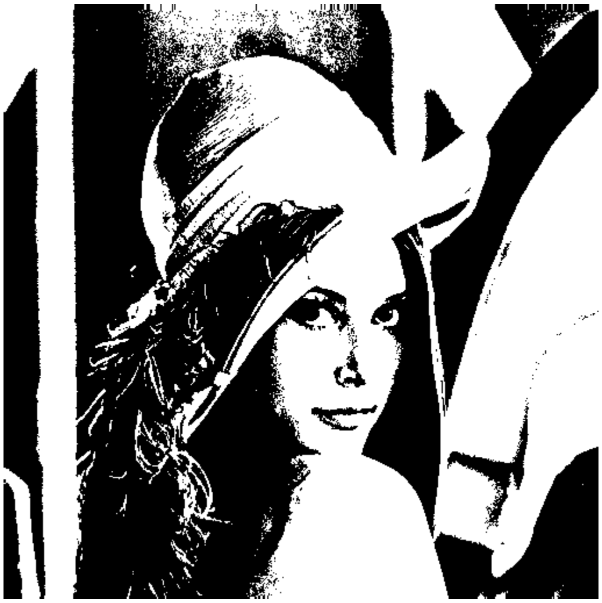
Computer Vision HW2

R09922093 楊子萱

In this homework, I use python3 and from PIL import Image and ImageDraw to read and write images and to draw the bounding blocks and the + on the picture. Matplotlib.pyplot to draw the historgram. To run my code, use command line and enter python3 hw2.py (http://hw2.py) .

(a)a binary image (threshold at 128)
Searching for the whole lena.bmp, if pixel value >= 128, then set the pixel to value 255. if pixel value < 128 then set the pixel to value 0.

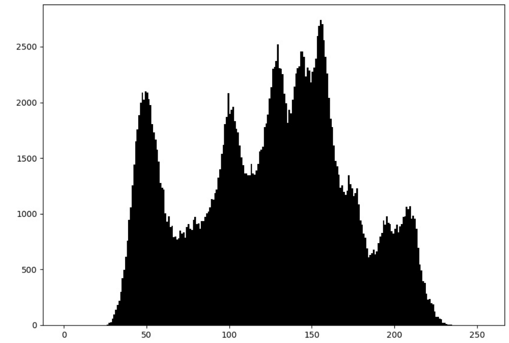
```
def binarize(threshold):
        for i in range(width):
                for j in range(height):
                        pixel = im.getpixel((i, j))
                        if pixel < threshold:
                                new.putpixel((i, j), 0)
                        else:
                                new.putpixel((i, j), 255)
        new.save("binarize.bmp")



threshold = 128
binarize(threshold)
```



(b)a histogram

Fist, get all the pixel values for the whole lena.bmp and convert them into a list. For plotting the histogram, use the plt.hist function to set the list range from 0 to 255 and plot.

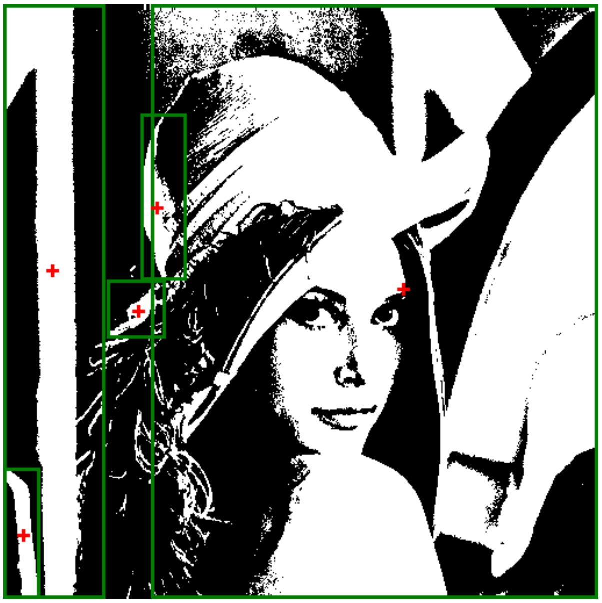
```
def histogram(im):
        width, height = im.size
        pixel_values = list(im.getdata())
        plt.figure(figsize=(10,7))
        plt.hist(pixel_values,range(0,255), color = "black")
        plt.savefig("plt.jpg")



histogram(im)
```



(c)connected components

In this part, I use the algorithm shown in 2.3.3 of the class ppt and the “4 connectivity” method to find the connoected component. First, initialization of each pixel to a unique label then, iteration of top-down followed by bottom-up passes until no change.

(For the page concerned, I put the result here first, and the function and code are on next page)



```
def connect(i, j):
        pixel_values = new.getpixel((i, j))
        if(i -1 < 0) :
                if(j - 1 < 0):
                        return -1
                elif(new.getpixel((i, j-1))== pixel_values):
                        return label[i][j-1]
                else:
                        return -1
        elif(j - 1 < 0):
                if (new.getpixel((i-1, j))== pixel_values):
                        return label[i-1][j]
                else:
                        return -1
        else:
                if(new.getpixel((i-1, j))!= pixel_values and new.getpixel((i, j-1))
                        return -1
                elif(new.getpixel((i-1, j)) != pixel_values
                     and new.getpixel((i, j-1)) == pixel_values):
                        return label[i][j-1]
                elif(new.getpixel((i-1, j)) == pixel_values
                     and new.getpixel((i, j-1)) != pixel_values):
                        return label[i-1][j]
                else:
                        maxx = max(label[i][j-1],label[i-1][j])
                        minn = min(label[i][j-1],label[i-1][j])
                        if maxx == minn:
                                return minn
                        else:
                                pairs[maxx] = minn
                                return minn
```

```
def check(i, j):
        pixel_values = new.getpixel((i, j))
        if(i + 1 > width-1 and j + 1 < height
            and new.getpixel((i, j+1)) == pixel_values):
                label[i][j] = label[i][j+1]
        elif(i + 1 < width and j + 1 > height-1 and new.getpixel((i+1, j)) == pixel
                label[i][j] = label[i + 1][j]
        elif(i+1 < width and j+1 < height):
                if (new.getpixel((i, j+1))!= pixel_values
                    and new.getpixel((i+1, j)) == pixel_values):
                        label[i][j] = label[i+1][j]
                elif (new.getpixel((i, j+1)) == pixel_values
                    and new.getpixel((i+1, j)) != pixel_values):
                        label[i][j] = label[i][j+1]
                elif (new.getpixel((i, j+1)) == pixel_values and new.getpixel((i+1,
                        maxx = max(label[i+1][j],label[i][j+1])
                        minn = min(label[i+1][j],label[i][j+1])
                        label[i][j] = minn
                        if maxx!=minn:
                                pairs[maxx] = minn



for j in range(height):
        for i in range(width):
                if(new.getpixel((i, j))== 255):
                        connected = connect(i, j)
                        if(connected == -1):
                                current += 1
                                label[i][j] = current
                        else:
                                label[i][j] = connected
        if(len(pairs) > 0):
                for i in range(width):
                        if (pairs. __contains__(label[i][j])):
                                label[i][j] = pairs.get(label[i][j])
        pairs.clear()

for j in range(height):
        for i in range(width):
                if(new.getpixel((width-i-1, height-j-1))) == 255:
                        check(width-i-1, height-j-1)
        if (len(pairs) > 0):
                for i in range(width):
                        if (pairs. __contains__(label[width-i-1][height-j-1])):
                                label[width-i-1][height-j-1] =
                                 pairs.get(label[width-i-1][height-j-1])
        pairs.clear()
```