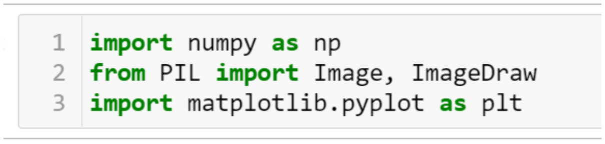
Computer Vision Homework 2

Name: 黃新予
Student ID: f08922136

I. Environment Setup

Language: Python 3 (on jupyter)
Library: numpy, PIL, matplotlib

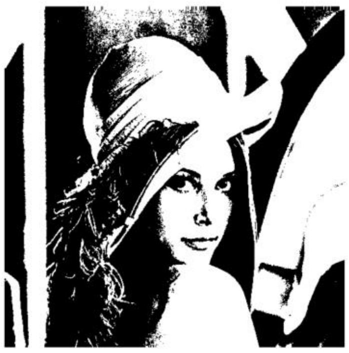


```python
import numpy as np
from PIL import Image, ImageDraw
import matplotlib.pyplot as plt
```

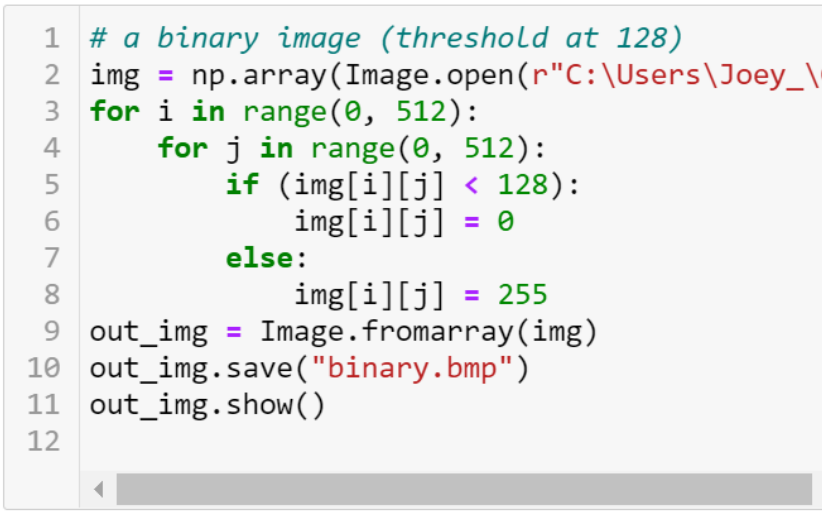
II. Q1: binary image (threshold at 128)

Step 1: Read image into np array
Step 2: traverse all the element in array and filter by threshold 128

output image:



code:

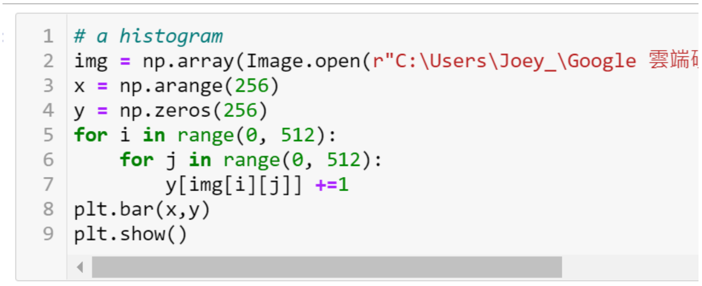


```python
# a binary image (threshold at 128)
img = np.array(Image.open(r"C:\Users\Joey_\
for i in range(0, 512):
    for j in range(0, 512):
        if (img[i][j] < 128):
            img[i][j] = 0
        else:
            img[i][j] = 255
out_img = Image.fromarray(img)
out_img.save("binary.bmp")
out_img.show()
```

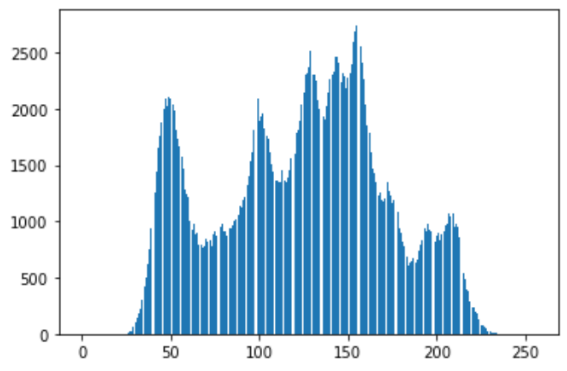
III. Q2: create a histogram

Step 1: read image as np array
Step 2: traverse all the element in np array and count the number
Step 3: use pyplot to draw histogram

output image and code:



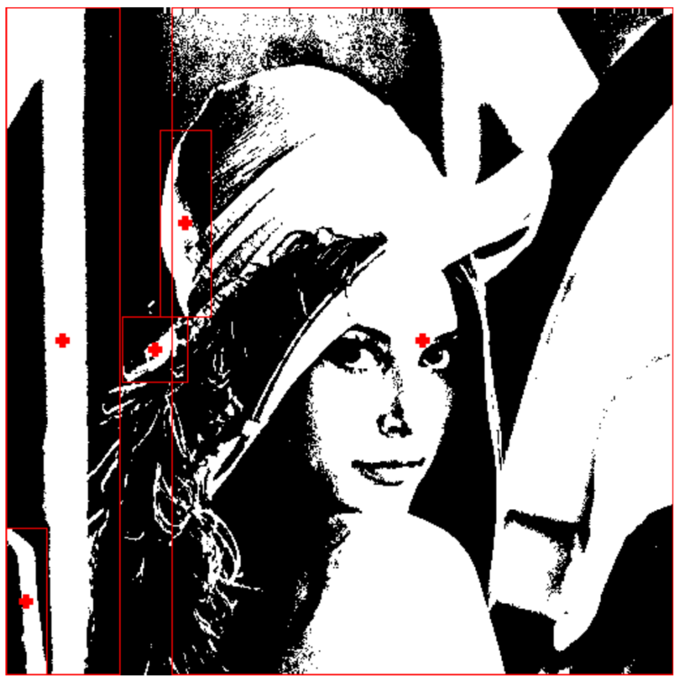
```python
# a histogram
img = np.array(Image.open(r"C:\Users\Joey_\Google 雲端
x = np.arange(256)
y = np.zeros(256)
for i in range(0, 512):
    for j in range(0, 512):
        y[img[i][j]] +=1
plt.bar(x,y)
plt.show()
```



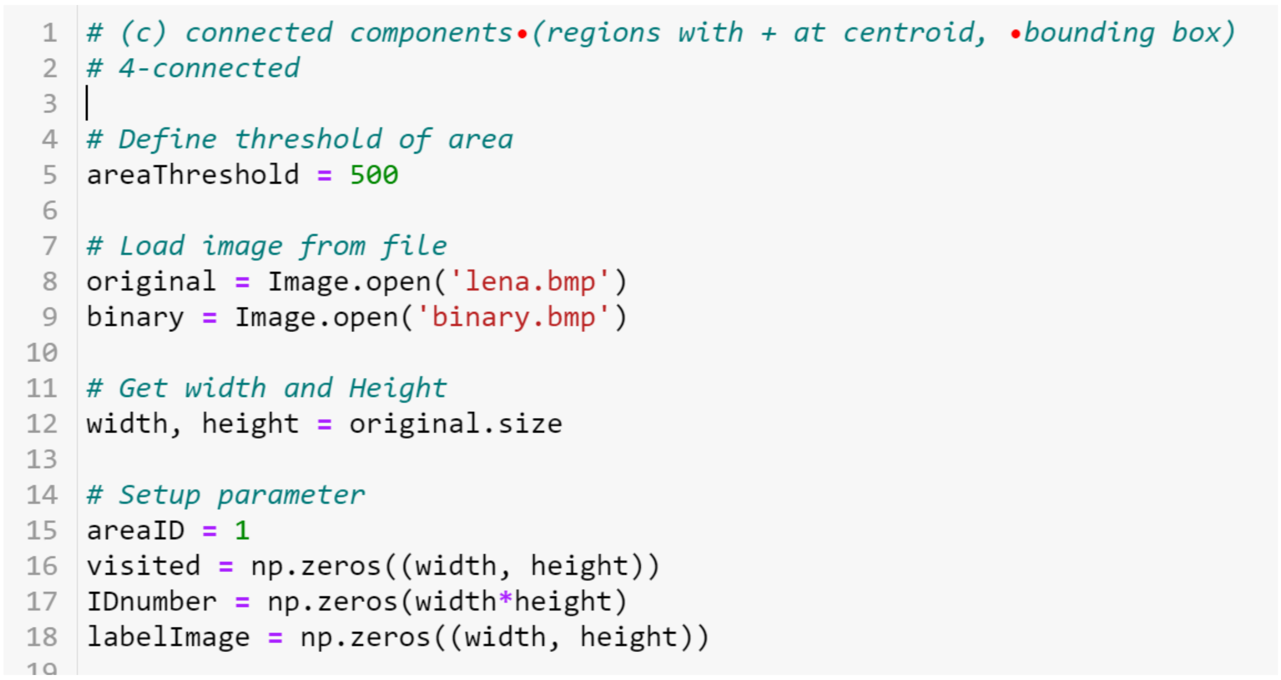
IV. Q3: connected components (regions with + at centroid, bounding box)

Ref: https://github.com/JasonYao81000/CV2017Fall

output image:



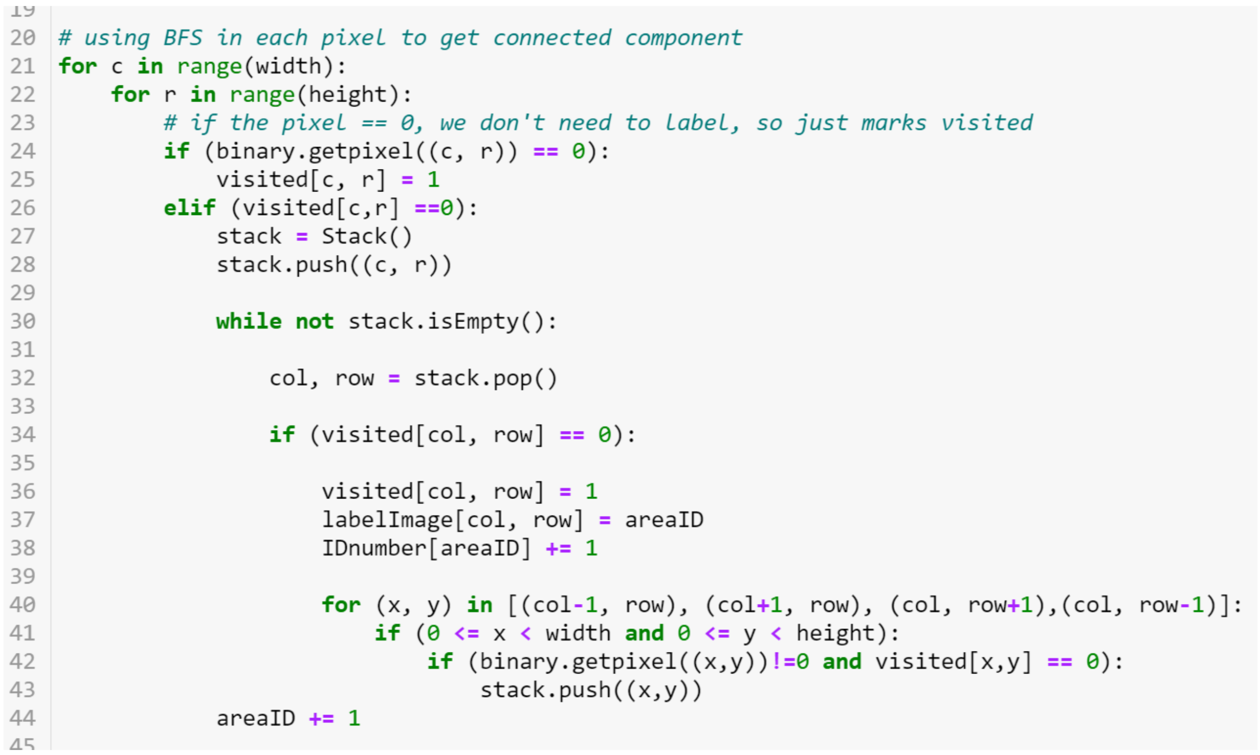
Step 1: read image and set up parameter



```python
# (c) connected components (regions with + at centroid, bounding box)
# 4-connected

# Define threshold of area
areaThreshold = 500

# Load image from file
original = Image.open('lena.bmp')
binary = Image.open('binary.bmp')

# Get width and Height
width, height = original.size

# Setup parameter
areaID = 1
visited = np.zeros((width, height))
IDnumber = np.zeros(width*height)
labelImage = np.zeros((width, height))
```

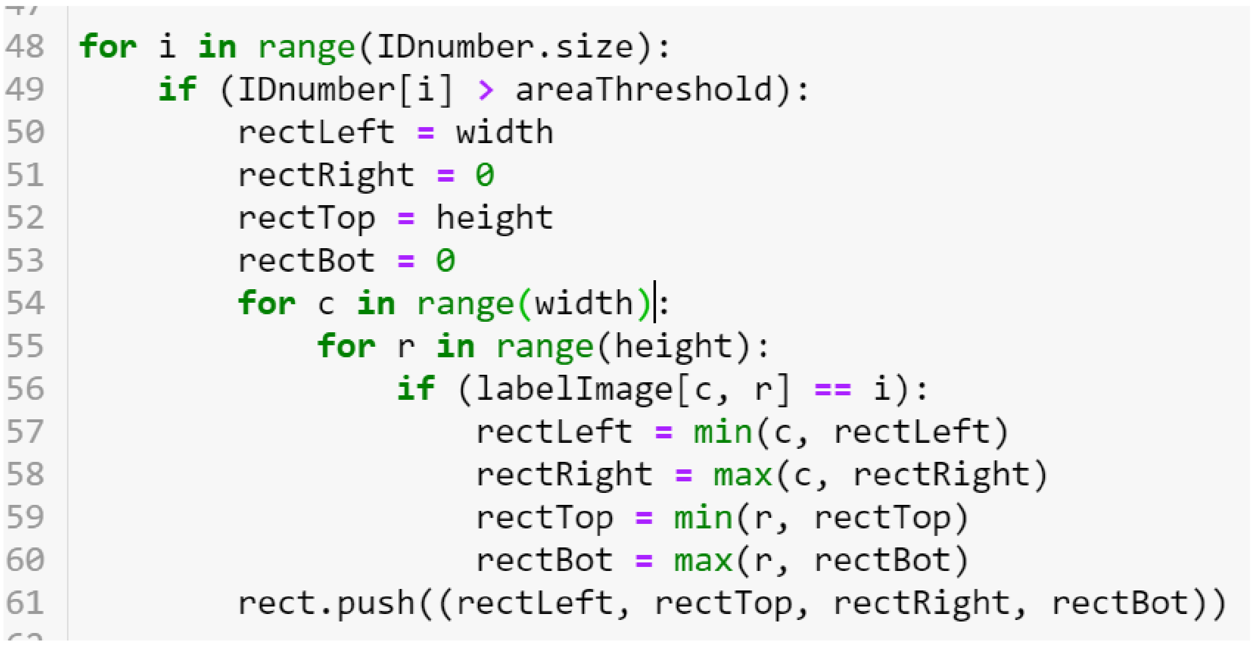
Step 2: using DFS in each pixel to get connected component

Step 3: compare neighbor and get 4-connected component



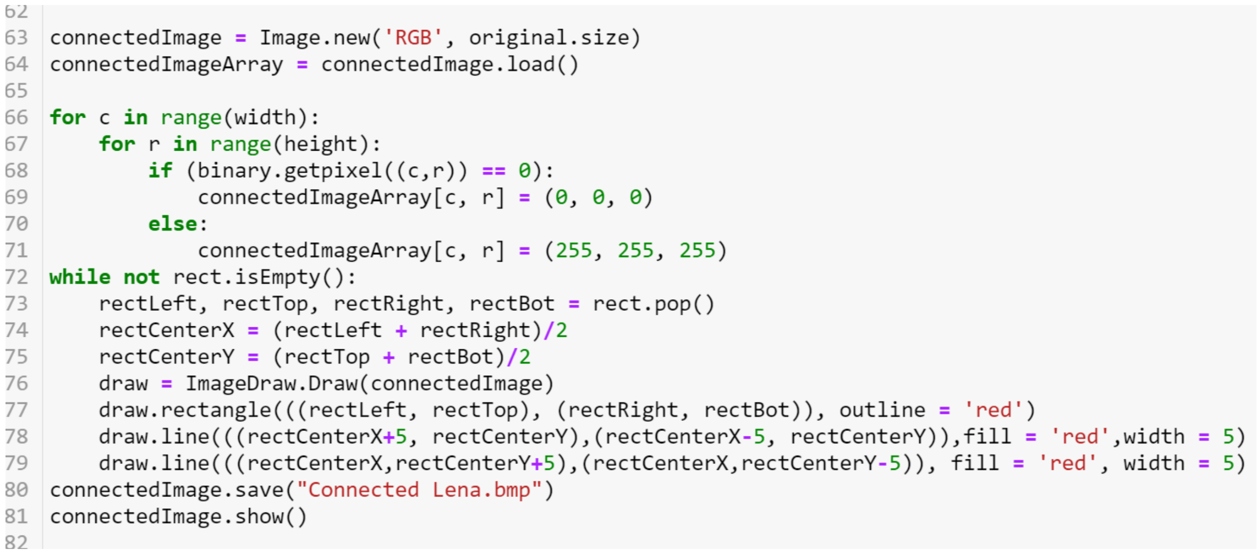
```python
# using BFS in each pixel to get connected component
for c in range(width):
    for r in range(height):
        # if the pixel == 0, we don't need to label, so just marks visited
        if (binary.getpixel((c, r)) == 0):
            visited[c, r] = 1
        elif (visited[c,r] ==0):
            stack = Stack()
            stack.push((c, r))

            while not stack.isEmpty():

                col, row = stack.pop()

                if (visited[col, row] == 0):

                    visited[col, row] = 1
                    labelImage[col, row] = areaID
                    IDnumber[areaID] += 1

                    for (x, y) in [(col-1, row), (col+1, row), (col, row+1),(col, row-1)]:
                        if (0 <= x < width and 0 <= y < height):
                            if (binary.getpixel((x,y))!=0 and visited[x,y] == 0):
                                stack.push((x,y))
            areaID += 1
```

Step 4: get left, right, top, bottom side of each connected component which area is larger than 500.



```python
for i in range(IDnumber.size):
    if (IDnumber[i] > areaThreshold):
        rectLeft = width
        rectRight = 0
        rectTop = height
        rectBot = 0
        for c in range(width):
            for r in range(height):
                if (labelImage[c, r] == i):
                    rectLeft = min(c, rectLeft)
                    rectRight = max(c, rectRight)
                    rectTop = min(r, rectTop)
                    rectBot = max(r, rectBot)
        rect.push((rectLeft, rectTop, rectRight, rectBot))
```

Step 5: draw rectangle and cross.



```python
connectedImage = Image.new('RGB', original.size)
connectedImageArray = connectedImage.load()

for c in range(width):
    for r in range(height):
        if (binary.getpixel((c,r)) == 0):
            connectedImageArray[c, r] = (0, 0, 0)
        else:
            connectedImageArray[c, r] = (255, 255, 255)
while not rect.isEmpty():
    rectLeft, rectTop, rectRight, rectBot = rect.pop()
    rectCenterX = (rectLeft + rectRight)/2
    rectCenterY = (rectTop + rectBot)/2
    draw = ImageDraw.Draw(connectedImage)
    draw.rectangle(((rectLeft, rectTop), (rectRight, rectBot)), outline = 'red')
    draw.line(((rectCenterX+5, rectCenterY),(rectCenterX-5, rectCenterY)),fill = 'red',width = 5)
    draw.line(((rectCenterX,rectCenterY+5),(rectCenterX,rectCenterY-5)), fill = 'red', width = 5)
connectedImage.save("Connected Lena.bmp")
connectedImage.show()
```