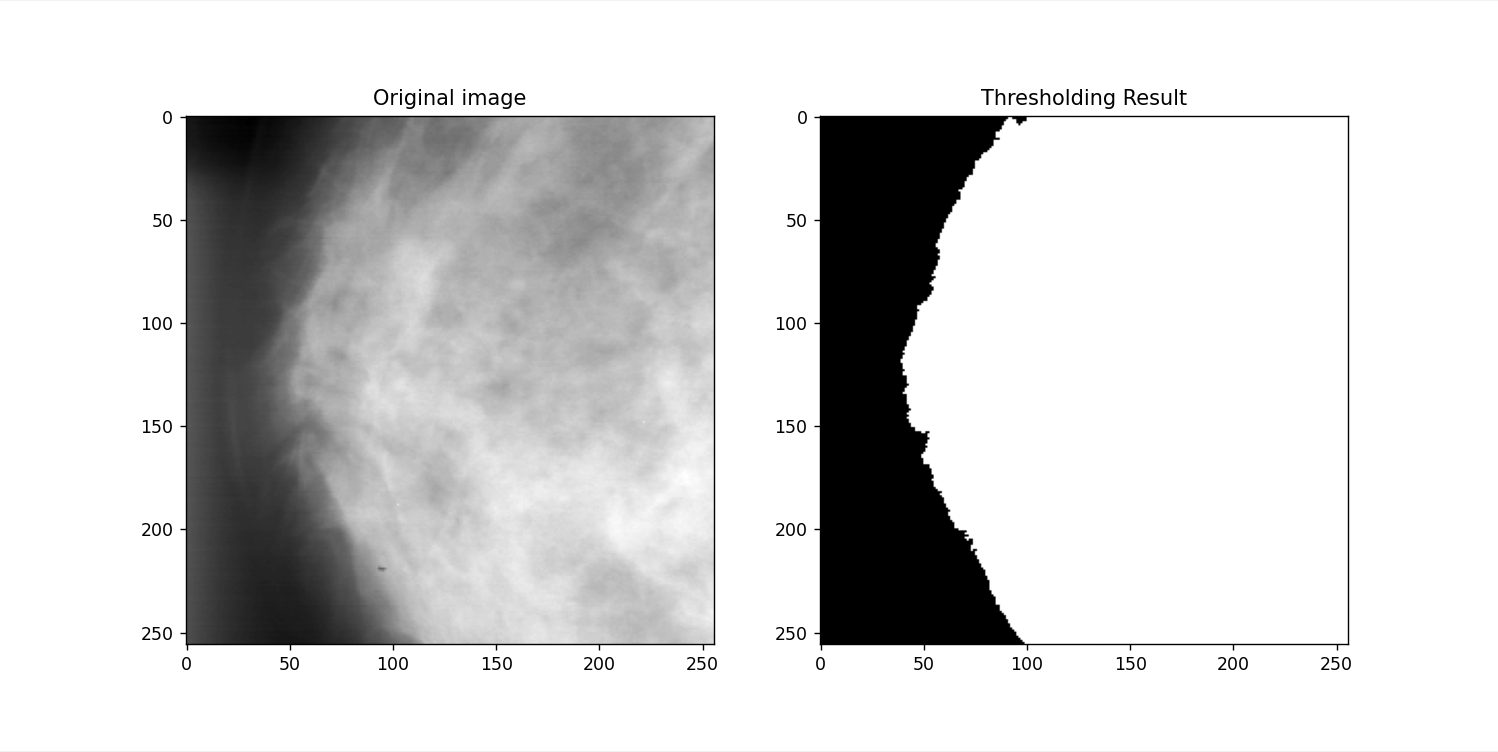
HOMEWORK 3

1. Bài 1

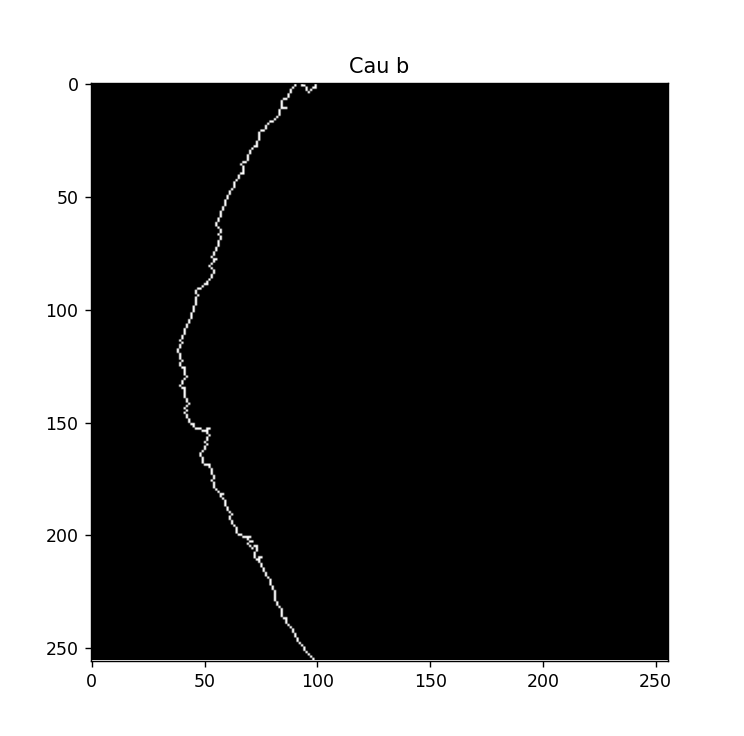
* Code:

#Họ và tên: Trần Thị Thúy  
#MSSV: N20DCCN154  
  
import numpy as np  
import matplotlib.pyplot as plt  
from matplotlib.colors import NoNorm  
  
  
def read\_file(file\_name, size):  
 try:  
 with open(file\_name, 'rb') as file:  
 data = np.fromfile(file, dtype=np.uint8, count=size\*size)  
 except FileNotFoundError:  
 print(f"\nLoi: Khong the mo file {file\_name}!")  
 return None  
 return data  
  
  
def plot\_a(data1, data2):  
 plt.figure(figsize=(12, 6))  
 plt.subplot(1, 2, 1)  
 plt.imshow(data1, cmap='gray', norm=NoNorm())  
 plt.title('Original image')  
  
 plt.subplot(1, 2, 2)  
 plt.imshow(data2, cmap='gray', norm=NoNorm())  
 plt.title('Thresholding Result')  
 plt.axis('image')  
  
  
def plot\_b(data, file\_name):  
 plt.figure(figsize=(6, 6))  
 plt.subplot(1, 1, 1)  
 plt.imshow(data, cmap='gray', norm=NoNorm())  
 plt.axis('image')  
 plt.title('Cau b', fontsize =12)  
 plt.savefig(file\_name, format='eps')  
 plt.savefig(file\_name[:-4]+'.pdf', format='pdf')  
  
  
def main():  
 #CAU A  
 xsize = 256  
 thresh = 95  
  
 file\_name = 'Mammogram\_256bin.sec'  
 x\_original = read\_file(file\_name, xsize)  
 if x\_original is None:  
 return  
 x\_original = x\_original.reshape(xsize, xsize)  
  
 x = 255 \* (x\_original >= thresh)  
 plot\_a(x\_original, x)  
  
 #CAU B  
 y = np.zeros((xsize, xsize), dtype=np.uint8)  
 for row in range(xsize):  
 for col in range(xsize):  
 if x[row, col] == 255:  
 y[row, col] = 0  
 else:  
 if((col > 0 and x[row, col-1] == 255) or  
 (col <xsize - 1 and x[row, col + 1] == 255) or  
 (row > 0 and x[row-1, col] == 255) or  
 (row < xsize -1 and x[row + 1, col] == 255)):  
 y[row, col] = 255  
 else:  
 y[row, col] = 0  
 plot\_b(y, 'Approximate Contour Image Generation')  
 plt.show()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

* Kết quả câu a:



* Kết quả câu b:



* Câu c: Một mã chuỗi có thể được dùng để biểu diễn đường viền chính trong hình ảnh đường viền hay không? Tại sao hoặc tại sao không?

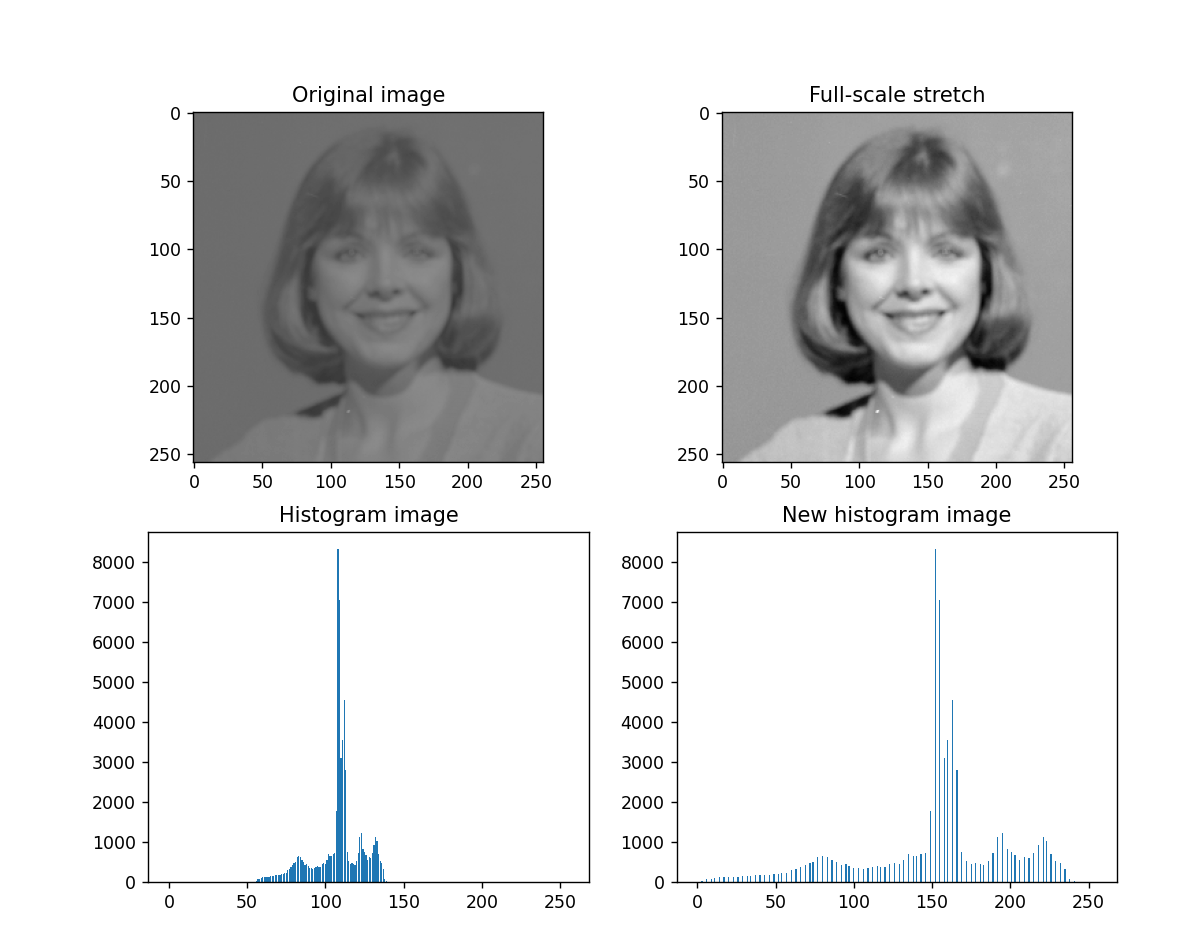
Trả lời: Mã chuỗi không thể được sử dụng để biểu diễn đường viền chính bởi vì thuật toán tạo đường viền chỉ mang tính xấp xỉ.

1. Bài 2

* Code:

#Họ và tên: Trần Thị Thúy  
#MSSV: N20DCCN154  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def read\_file(file\_name, xsize):  
 try:  
 with open(file\_name, 'rb') as file:  
 data = np.fromfile(file, dtype=np.uint8, count=xsize \* xsize)  
 if data.size != xsize \* xsize:  
 raise Exception(f"Doc file {file\_name} that bai")  
 return data.reshape(256, 256)  
 except FileNotFoundError:  
 raise Exception(f'Khong the mo file {file\_name}')  
  
  
def stretch(a):  
 min\_val = np.min(a)  
 max\_val = np.max(a)  
 x = 255.0 / (max\_val - min\_val)  
 stretched = np.round((a - min\_val) \* x)  
 return stretched.astype(np.uint8)  
  
  
def plot2(data1, data2, data3, data4):  
 plt.figure(figsize=(10, 8))  
 plt.subplot(2, 2, 1)  
 plt.imshow(data1, cmap='gray', vmin=0, vmax=255)  
 plt.title('Original image')  
  
 plt.subplot(2, 2, 2)  
 plt.imshow(data2, cmap='gray')  
 plt.title('Full-scale stretch')  
  
 plt.subplot(2, 2, 3)  
 plt.bar(range(256), data3)  
 plt.title('Histogram image')  
  
 plt.subplot(2, 2, 4)  
 plt.bar(range(256), data4)  
 plt.title('New histogram image')  
 plt.show()  
  
  
def main():  
 xsize = 256  
 x = read\_file('ladybin.sec', xsize)  
 histogram\_x = np.histogram(x, bins= np.arange(257), range=(0, 256))[0]  
 y = stretch(x)  
 histogram\_y = np.histogram(y, bins= np.arange(257), range=(0, 256))[0]  
 plot2(x, y, histogram\_x, histogram\_y)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

* Kết quả:

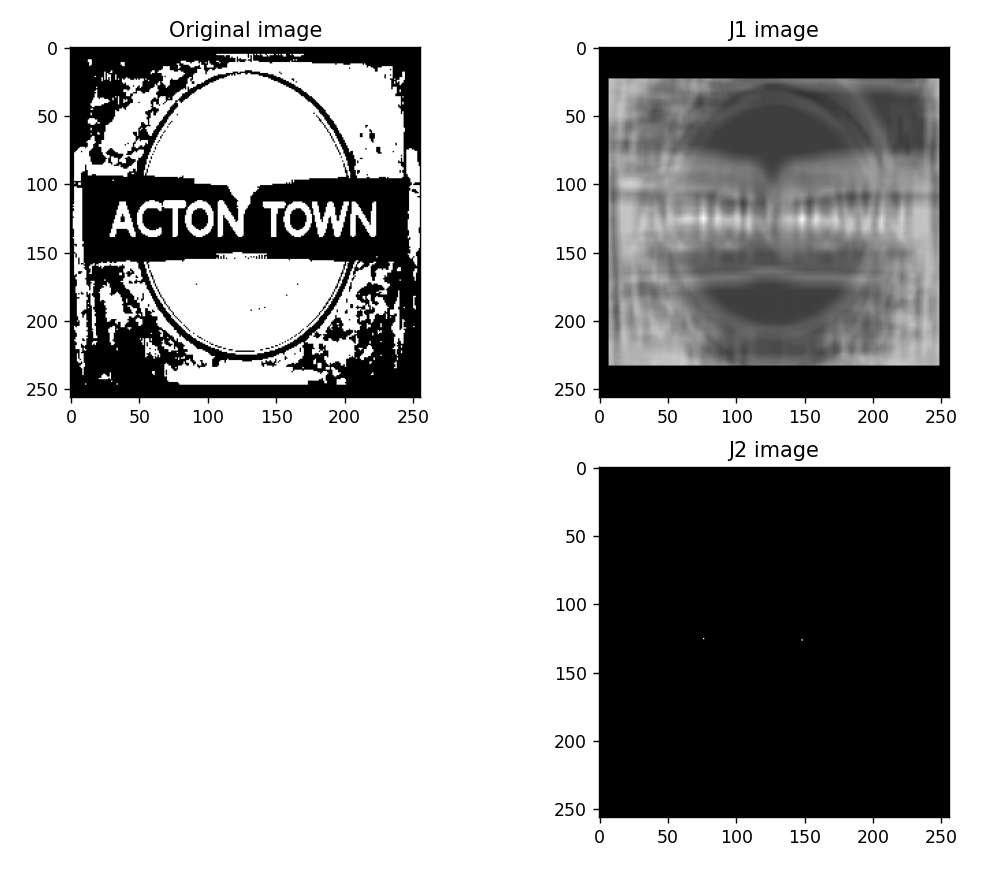


1. Bài 3

* Code:

#Họ và tên: Trần Thị Thúy  
#MSSV: N20DCCN154  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def read\_file(file\_name, size):  
 try:  
 with open(file\_name, 'rb') as file:  
 data = np.fromfile(file, dtype=np.uint8, count=size\*size)  
 except FileNotFoundError:  
 print(f"\nLoi: Khong the mo file {file\_name}!")  
 return None  
 return data  
  
  
def plot2(data1, data2, data3):  
 plt.figure(figsize=(10, 8))  
 plt.subplot(2, 2, 1)  
 plt.imshow(data1, cmap='gray')  
 plt.title('Original image')  
  
 plt.subplot(2, 2, 2)  
 plt.imshow(data2, cmap='gray', vmin=0, vmax=255)  
 plt.title('J1 image')  
  
 plt.subplot(2, 2, 4)  
 plt.imshow(data3, cmap='gray', vmin=0, vmax=255)  
 plt.title('J2 image')  
  
 plt.show()  
  
  
#Hàm tính toán đo lường M2  
def compute\_match\_measure(image, template):  
 J1 = np.zeros\_like(image, dtype=int)  
 TemplRowso2, TemplColso2 = template.shape[0]//2, template.shape[1]//2  
  
 for row in range(TemplRowso2, image.shape[0] - TemplRowso2):  
 for col in range(TemplColso2, image.shape[1] - TemplColso2):  
 window\_set = image[row - TemplRowso2:row + TemplRowso2 + 1, col - TemplColso2:col + TemplColso2 + 1]  
 J1[row, col] = np.sum(window\_set == template)  
   
 return J1  
  
def bult\_templ():  
 TemplRows = 47  
 TemplCols = 15  
 TemplRowso2 = TemplRows // 2  
 TemplColso2 = TemplCols // 2  
 Templ = np.zeros((TemplRows, TemplCols), dtype=np.uint8)  
  
 for row in range(10):  
 for col in range(TemplCols):  
 Templ[row, col] = 0  
  
 for row in range(10, 16):  
 for col in range(TemplCols):  
 Templ[row, col] = 255  
  
 for row in range(17, 37):  
 for col in range(6):  
 Templ[row, col] = 0  
 for col in range(6, 10):  
 Templ[row, col] = 255  
 for col in range(10, TemplCols):  
 Templ[row, col] = 0  
  
 for row in range(38, TemplRows):  
 for col in range(TemplCols):  
 Templ[row, col] = 0  
  
 return Templ  
  
  
def stretch(a):  
 min\_val = np.min(a)  
 max\_val = np.max(a)  
 x = 255.0 / (max\_val - min\_val)  
 stretched = np.round((a - min\_val) \* x)  
 return stretched.astype(np.uint8)  
  
  
def main():  
 xsize = 256  
 file\_name = 'actontBinbin.sec'  
 x = read\_file(file\_name, xsize)  
 if x is None:  
 return  
 x = x.reshape(xsize, xsize)  
 Templ = bult\_templ()  
  
 J1 = compute\_match\_measure(x, Templ)  
 J1 = stretch(J1)  
  
 count = np.sort(J1.ravel())[::-1]  
 threshold = count[1]  
 J2 = 255 \* (J1 >= threshold)  
 plot2(x, J1, J2)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

* Kết quả:



1. Câu 4

* Code:

#Họ và tên: Trần Thị Thúy  
#MSSV: N20DCCN154  
  
import numpy as np  
import matplotlib.pyplot as plt  
  
  
def read\_file(file\_name, size):  
 try:  
 with open(file\_name, 'rb') as file:  
 data = np.fromfile(file, dtype=np.uint8, count=size\*size)  
 except FileNotFoundError:  
 print(f"\nLoi: Khong the mo file {file\_name}!")  
 return None  
 return data  
  
  
def plot2(data1, data2, data3, data4):  
 plt.figure(figsize=(10, 8))  
 plt.subplot(2, 2, 1)  
 plt.imshow(data1, cmap='gray', vmin=0, vmax=255)  
 plt.title('Original image')  
  
 plt.subplot(2, 2, 2)  
 plt.bar(np.arange(256), data2)  
 plt.title('Original Histogram')  
 plt.axis([0, 256, 0, 2500])  
 plt.xticks(np.arange(0, 256, 16))  
 plt.yticks(np.arange(0, 2500, 250))  
  
 plt.subplot(2, 2, 3)  
 plt.imshow(data3, cmap='gray', vmin=0, vmax=255)  
 plt.title('Histogram Equalized Image')  
  
 plt.subplot(2, 2, 4)  
 plt.bar(np.arange(256), data4)  
 plt.title('Equalized Histogram')  
 plt.axis([0, 256, 0, 2500])  
 plt.xticks(np.arange(0, 256, 16))  
 plt.yticks(np.arange(0, 2500, 250))  
  
 plt.show()  
  
def stretch(a):  
 min\_val = np.min(a)  
 max\_val = np.max(a)  
 x = 255.0 / (max\_val - min\_val)  
 stretched = np.round((a - min\_val) \* x)  
 return stretched.astype(np.uint8)  
  
  
def hist\_equalization(image):  
 hist = np.histogram(image, bins=256, range=(0, 256))[0]  
 num\_pixels = image.size  
 x = hist/num\_pixels  
  
 y = np.cumsum(x)  
 J = y[image]  
  
 K = stretch(J)  
 return K  
  
  
def main():  
 xsize = 256  
 file\_name = 'johnnybin.sec'  
 x = read\_file(file\_name, xsize)  
 if x is None:  
 return  
 x = x.reshape(xsize, xsize)  
  
 hist\_x = np.histogram(x, bins=256, range=(0, 256))[0]  
  
 y = hist\_equalization(x)  
 hist\_y = np.histogram(y, bins=256, range=(0, 256))[0]  
  
 plot2(x, hist\_x, y, hist\_y)  
  
  
if \_\_name\_\_ == '\_\_main\_\_':  
 main()

* Kết quả:

