**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC CÔNG NGHỆ TP.HCM**

**BÁO CÁO THỰC HÀNH XỬ LÝ ẢNH TRONG Y TẾ**

SV : Trần Trọng Tấn

Lớp : 18DYSA1

www.hutech.edu.vn

**BÁO CÁO THỰC HÀNH TIN HỌC KỸ THUẬT**

Ấn bản 2019

MỤC LỤC

[MỤC LỤC 1](#_Toc16679666)

[BÀI 1: CƠ SỞ VỀ XỬ LÝ ẢNH 2](#_Toc16679667)

[BÀI 2: XỬ LÝ ẢNH NÂNG CAO 10](#_Toc16679668)

# CƠ SỞ VỀ XỬ LÝ ẢNH

**Bài 1.1** Đọc ảnh:

1. Code

table = imread(['image1.jpg']);

imshow(table)

1. Kết quả

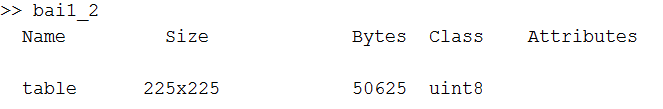


**Bài 1.2**

1. Code

Whos

1. Kết quả



**Bài 1.3**

1. Code

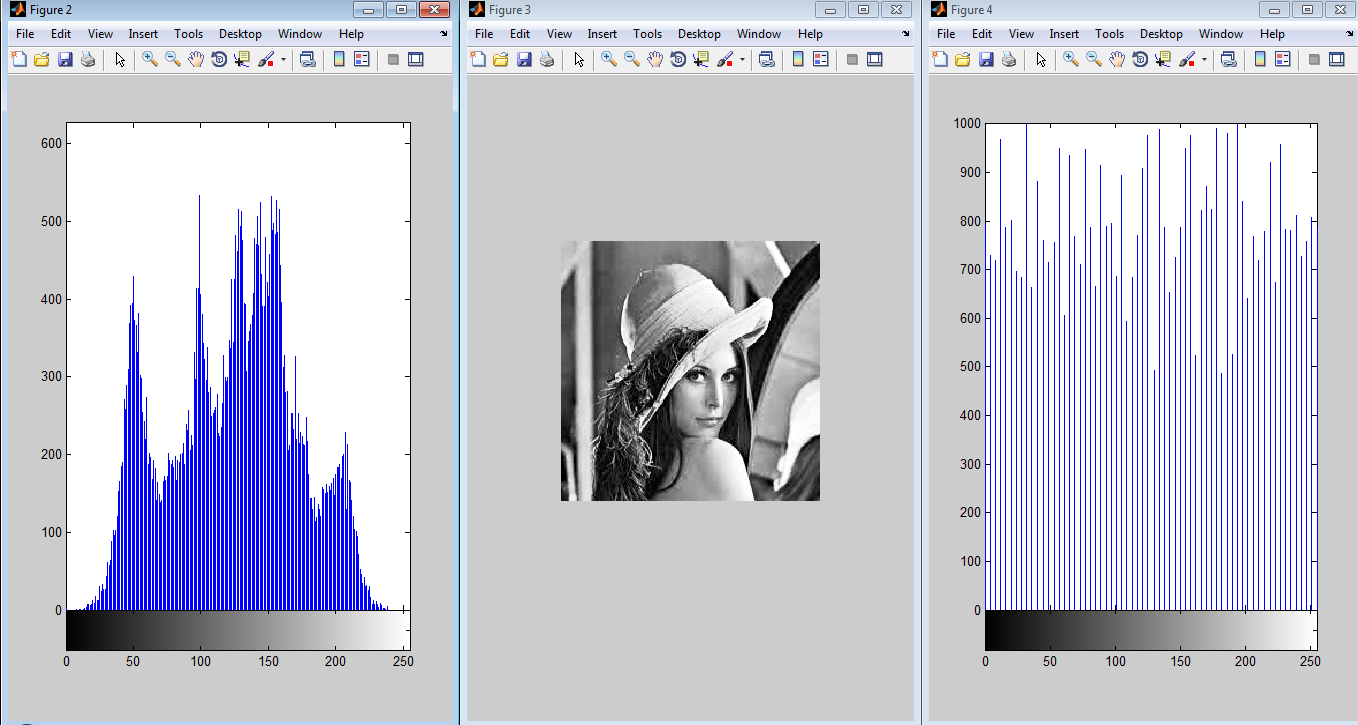
figure, imhist(table)

table2=histeq(table);

figure, imshow(table2)

figure, imhist(table2)

1. Kết quả

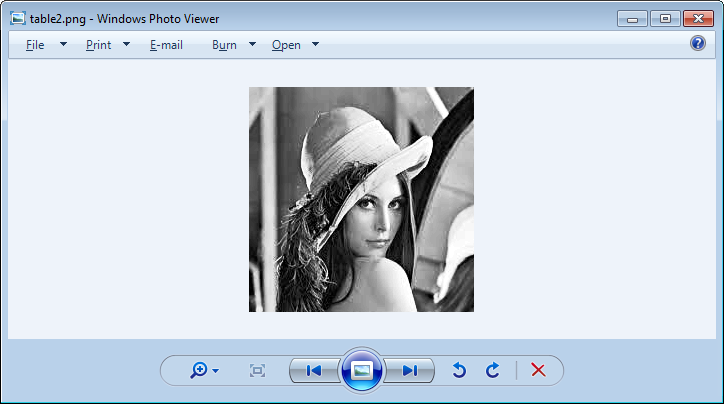


**Bài 1.4**

1. Code

imwrite(table2, 'table2.png');

1. Kết quả



**Bài 1.5**

1. Code

imfinfo('table2.png')

1. Kết quả

ans =

Filename: 'C:\Users\Administrator\Documents\MATLAB\table2.png'

FileModDate: '12-Dec-2020 14:46:30'

FileSize: 24512

Format: 'png'

FormatVersion: []

Width: 225

Height: 225

BitDepth: 8

ColorType: 'grayscale'

FormatSignature: [137 80 78 71 13 10 26 10]

Colormap: []

Histogram: []

InterlaceType: 'none'

Transparency: 'none'

SimpleTransparencyData: []

BackgroundColor: []

RenderingIntent: []

Chromaticities: []

Gamma: []

XResolution: []

YResolution: []

ResolutionUnit: []

XOffset: []

YOffset: []

OffsetUnit: []

SignificantBits: []

ImageModTime: '12 Dec 2020 07:46:30 +0000'

Title: []

Author: []

Description: []

Copyright: []

CreationTime: []

Software: []

Disclaimer: []

Warning: []

Source: []

Comment: []

OtherText: []

# XỬ LÝ ẢNH NÂNG CAO

**Bài 2.1**

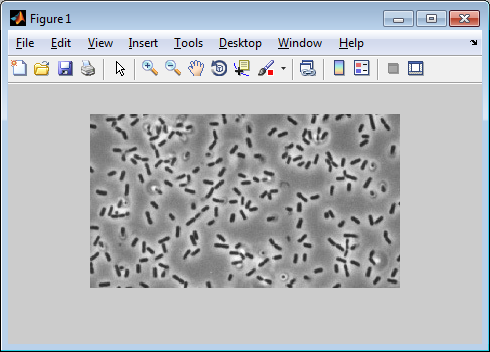
1. Code

clear, close all

i=imread('santa.jpg');

imshow(i)

1. Kết quả



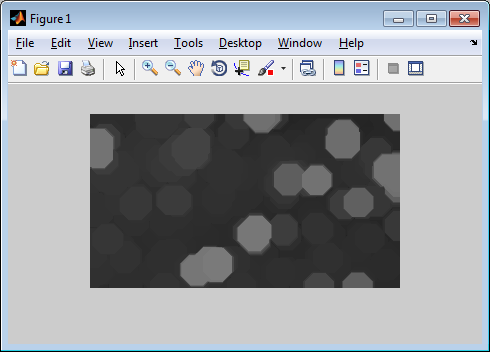
**Bài 2.2**

a.Code

bg=imopen(i,strel('disk',15));

imshow(bg)

b.Kết quả



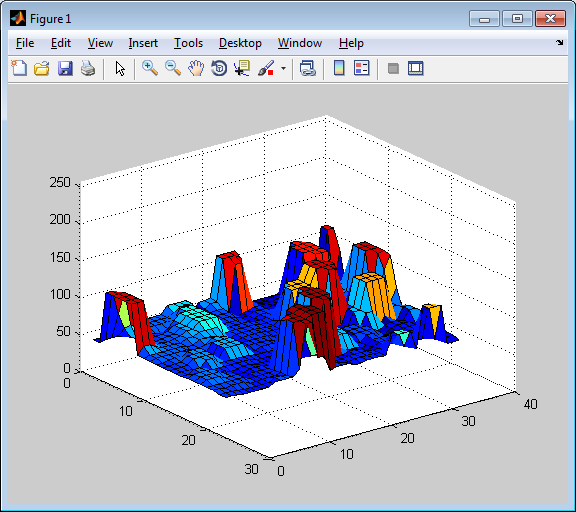
**Bài 2.3**

a.Code

figure, surf(double(bg(1:8:end, 1:8:end))),zlim([0 255]);

set(gca, 'ydir', 'reverse');

b.Kết quả



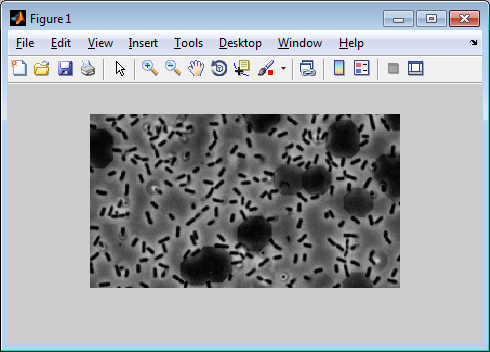
**Bài 2.4**

a.Code

i2=imsubtract(i,bg);

figure, imshow(i2)

b.Kết quả



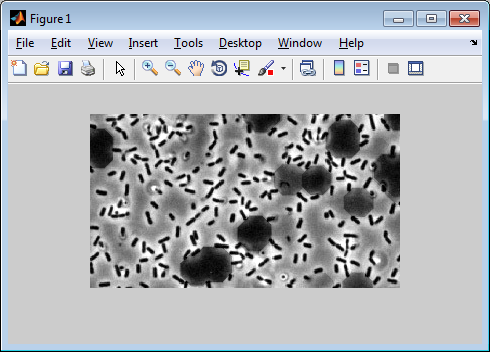
**Bài 2.5**

a.Code

i3= imadjust(i2, stretchlim(i2), [0 1]);

figure, imshow(i3)

b.Kết quả



**Bài 2.6**

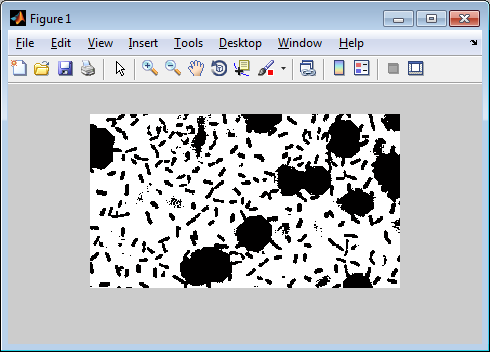
a.Code

lv=graythresh(i3);

bw=im2bw(i3,lv);

figure, imshow(bw)

b.Kết quả



**Bài 2.7**

a.Code

[labeled,numObjects]=bwlabel(bw,4);

numObjects

b.Kết quả

numObjects =

24

**Bài 2.8**

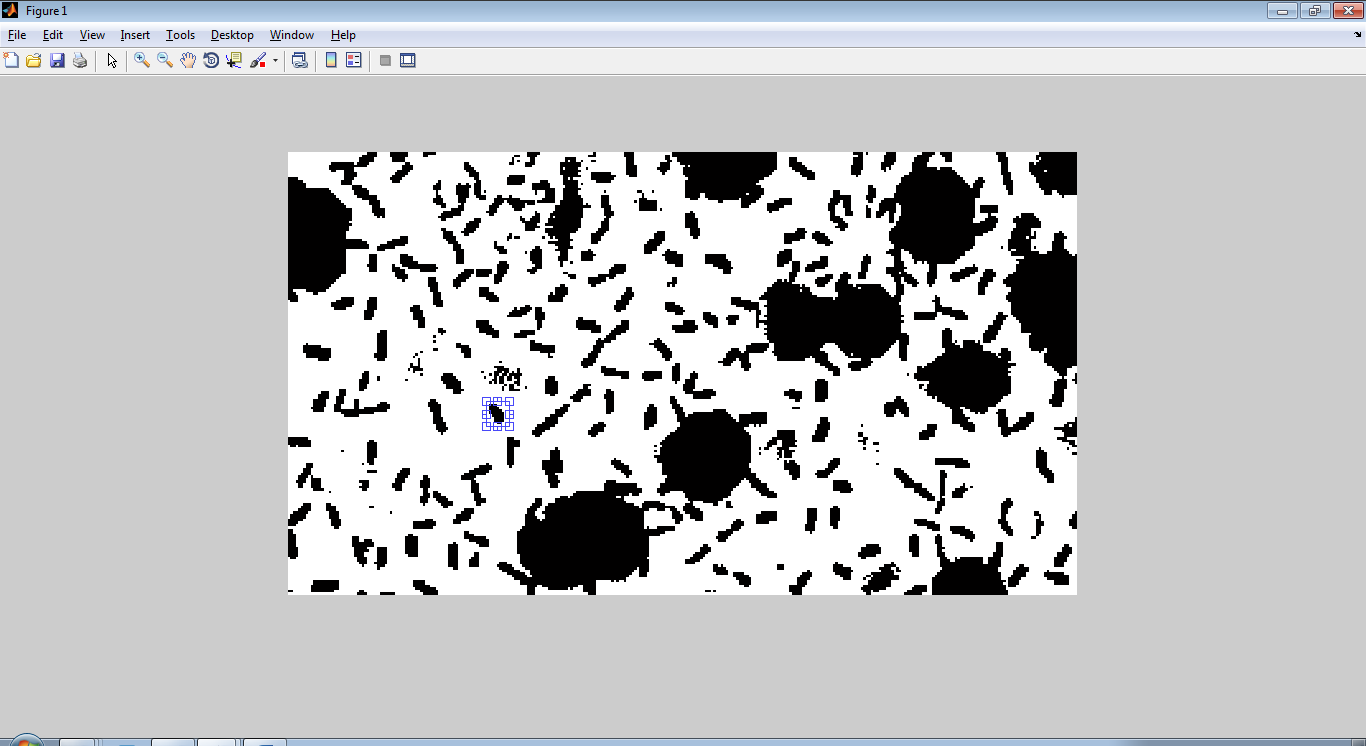
a.Code

grain=imcrop(labeled)

RGB\_label=label2rgb(labeled, @spring,'c','shuffle');

imshow(RGB\_label);

b.Kết quả



**grain =**

**1 1 1 1 1 1 1 1 1**

**1 0 0 0 1 1 1 1 1**

**1 0 0 0 0 1 1 1 1**

**1 0 0 0 0 0 1 1 1**

**1 0 0 0 0 0 0 1 1**

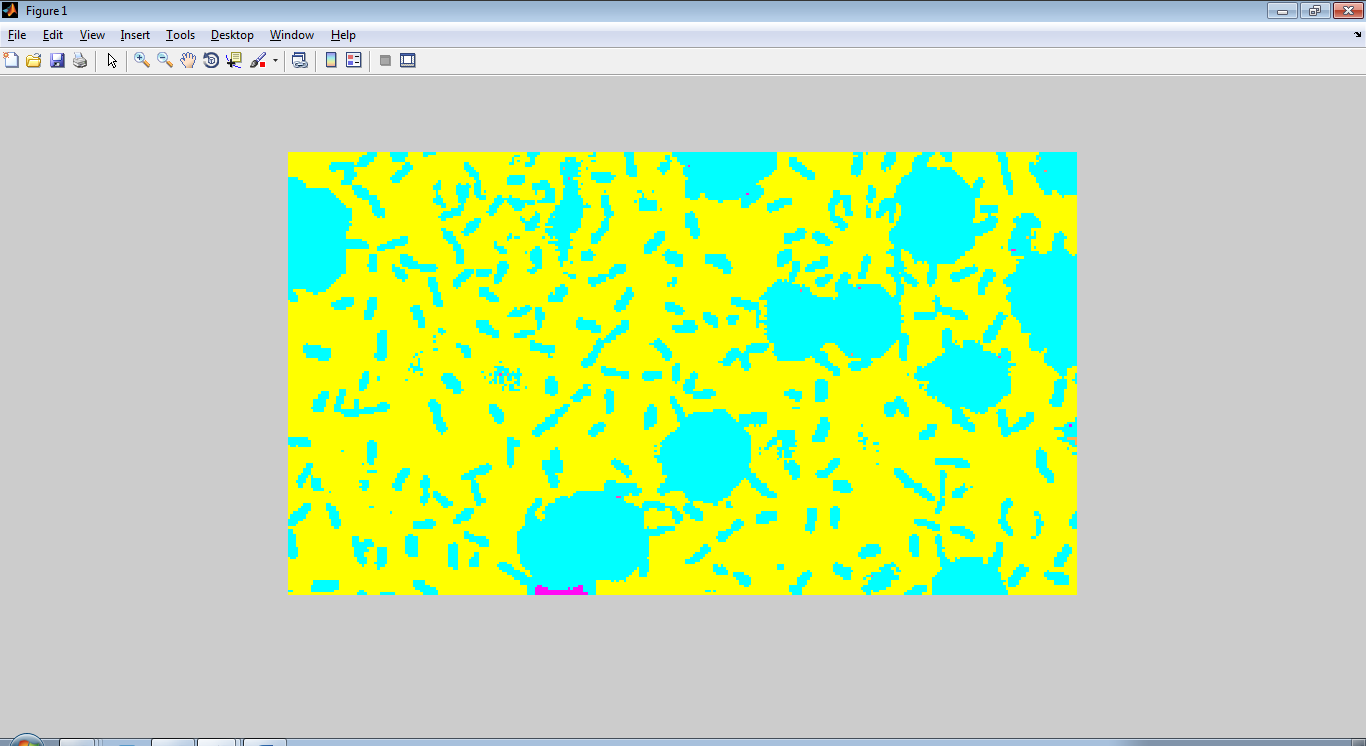
**1 1 0 0 0 0 0 1 1**

**1 1 1 0 0 0 0 1 1**

**1 1 1 0 0 0 0 1 1**

**1 1 1 1 0 0 1 1 1**

**1 1 1 1 1 1 1 1 1**



**Bài 2.9**

a.Code

graindata=regionprops(labeled,'basic')

graindata(21).Area

graindata(21).BoundingBox, graindata(21).Centroid

allgrains=[graindata.Area];

whos allgrains

b.Kết quả

graindata =

24x1 struct array with fields:

Area

Centroid

BoundingBox

ans =

1

ans =

Columns 1 through 3

295.5000 2.5000 1.0000

Column 4

1.0000

ans =

296 3

Name Size Bytes Class Attributes

allgrains 1x24 192 double

**Bài 2.10**

a.Code

max(allgrains)

biggrain=find(allgrains==36971)

mean(allgrains)

hist(allgrains,20)

b.Kết quả

ans =

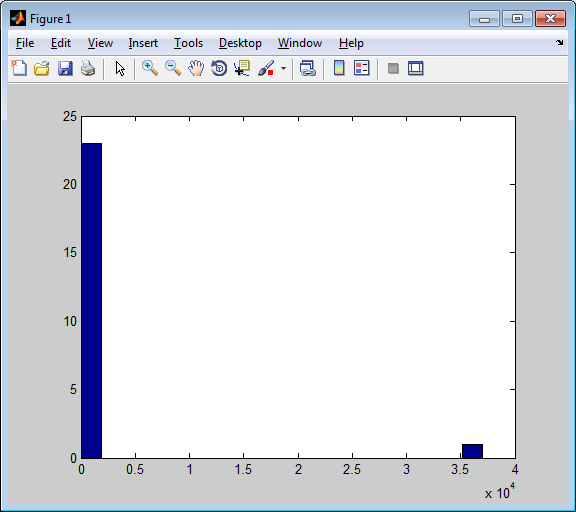
36971

biggrain =

1

ans =

1.5439e+03



# XỬ LÝ ẢNH Y TẾ - PHẦN 1

Phần 1, cung cấp các kiến thức:

(1) load MRI data into MATLAB

(2) display three anatomically oriented slices of the head MRI

(3) apply a binary mask to extract the brain tissue

(4) display three brain slices

(5) implement an edge detector to grab and display all edges of a slice

Nhiệm vụ của SV, thực hiện theo các bước hướng dẫn, ghi nhận báo cáo kết quả và nhận xét, sau đó làm các bài tập theo yêu cầu ở cuối phần 1

% load mat file containing the original mri

load raw\_mri

% see what's in memory

whos

% create a subplot window

subplot(2,2,1)

% grab the coronal slice corresponding to y=128 and store it in a new

% variable

coronal\_slice = mri(:,128,:);

% NOTE: colon operator denotes "all" elements in the given dimension

% NOTE: semicolon after line suppresses output

% get the size of the slice

size(coronal\_slice)

% NOTE: no semicolon as we want to display the current size of the variable

% "coronal\_slice"

% remove "singleton" dimension

coronal\_slice = squeeze(coronal\_slice);

% observe the size of the array after squeezing

size(coronal\_slice)

% NOTE: "coronal\_slice" is now a 2-dimensional array

% display slice

imagesc(coronal\_slice);

% NOTE: intensity of each pixel corresponds to tissue type

% change to a grayscale colormap

colormap(gray)

% add a title to our subplot

title('Coronal Slice')

% display the coronal slice right side up

% create a second panel

subplot(2,2,2)

% display the transposed matrix

imagesc(coronal\_slice');

% NOTE: in MATLAB, the prime operator ' denotes matrix transposition

% make the axes Cartesian

axis xy

% insert new title

title('Coronal Slice Right Side Up')

% display the sagittal slice

subplot(2,2,3)

% grab a slice corresponding to the y-z plane

sagittal\_slice = mri(128,:,:);

sagittal\_slice = squeeze(sagittal\_slice);

imagesc(sagittal\_slice');

axis xy

title('Sagittal Slice')

% display the axial slice

subplot(2,2,4)

% grab a slice corresponding to the x-y plane

axial\_slice = mri(:,:,95);

axial\_slice = squeeze(axial\_slice);

imagesc(axial\_slice');

axis xy

title('Axial Slice')

%%

% load mat file containing our data: mask of the brain

load brain\_mask

% display the contents of memory

whos

% NOTE: the mask is stored in a variable called "m"

% get the size of the mask m

size(m)

% get the type of data contained in the "m" variable

class(m)

% get the type of data contained in the "mri" variable

class(mri)

% since the data types don't match up, we must convert one of them to match

% the other if we want to do any processing

% convert "mri" to a floating-point array (as opposed to integer data)

mri\_new = double(mri);

% compare the size of "mri\_new" with that of "mri"

whos

% now observe the type of data in "mri\_new"

class(mri\_new)

% apply the mask to the mri image

masked\_mri = mri\_new.\*m;

% NOTE: .\* denotes element-by-element multiplication, which is not the same

% as matrix multiplication

% display three brain slices

% create a new figure window

figure

subplot(2,2,1)

masked\_coronal\_slice = masked\_mri(:,128,:);

masked\_coronal\_slice = squeeze(masked\_coronal\_slice);

imagesc(masked\_coronal\_slice');

axis xy

colormap(gray)

title('Masked Coronal Slice')

% display the sagittal slice

subplot(2,2,2)

masked\_sagittal\_slice = masked\_mri(128,:,:);

masked\_sagittal\_slice = squeeze(masked\_sagittal\_slice);

imagesc(masked\_sagittal\_slice');

axis xy

colormap(gray)

title('Masked Sagittal Slice')

% display the axial slice

subplot(2,2,3)

masked\_axial\_slice = masked\_mri(:,:,95);

masked\_axial\_slice = squeeze(masked\_axial\_slice);

imagesc(masked\_axial\_slice');

axis xy

colormap(gray)

title('Masked Axial Slice')

%%

% IN-CLASS ASSIGNMENT

%%

% TASK 1

% Your job is to use the "edge" function of the image processing toolbox to

% find and display all images in the axial slice of the brain mask. Once

% you have computed the edges, you should display the edgemap in the

% remaining subplot panel (bottom right) with the proper orientation.

%

% type "help edge" in the command window to view the function prototype and

% argument information for the edge function

% TASK 2

% In a new figure window, display a histogram of MR intensities for all non-zero voxels. To do

% this, you will first use the "find" function to identify the voxels that

% have a non-zero intensity. Next, create a new vector to store the intensities

% of all non-zero voxels. Finally, you will call the "hist" function with

% this vector as the argument to display the intensity histogram.

% TASK 3

%

% In a new figure window, iterate across the z-dimension (dimension 3 in

% the array mri\_new) and display the value of each x-y slice successively.

% This requires the use of a "for-loop" as well as the "drawnow" function.

% The pseudocode of the script is given below. You need only fill in the

% question marks with the appropriate matlab variables/commands.

Bài làm:

% Dùng load để load dữ liệu file mat gốc

load raw\_mri

% xem bộ nhớ

whos

% tạo cửa sở Subplot

subplot(2,2,1)

% tạo lớp cắt coronal slice với y=128 và lưu vào biến mới, ngoài ra dấu “:” để lấy tất cả dữ liệu trong file

coronal\_slice = mri(:,128,:);

% lấy kích thước coronal slice

size(coronal\_slice)

% loại bỏ không gian singleton bằng lệnh squeezing

coronal\_slice = squeeze(coronal\_slice);

% lấy kích thước coronal slice

size(coronal\_slice)

% coronal\_slice giờ là mảng 2 chiều

% hiển thị lát

imagesc(coronal\_slice);

% Đổi ảnh sang dạng ảnh xám

colormap(gray)

% thêm tiêu đề

title('Coronal Slice')

% Hiển thị Coronal Slice bên phải

% tạo bảng Subplot thứ 2

subplot(2,2,2)

% hiển thị ma trận chuyển vị

imagesc(coronal\_slice');

% tạo trục Cartesian

axis xy

% chèn tiêu đề mới

title('Coronal Slice Right Side Up')

% hiển thị sagittal slice

subplot(2,2,3)

% lấy một slice tương ứng mặt phẳng y-z

sagittal\_slice = mri(128,:,:);

sagittal\_slice = squeeze(sagittal\_slice);

imagesc(sagittal\_slice');

axis xy

title('Sagittal Slice')

% hiển thị trục cắt

subplot(2,2,4)

% lấy một slice tương ứng mặt phẳng x-y

axial\_slice = mri(:,:,95);

axial\_slice = squeeze(axial\_slice);

imagesc(axial\_slice');

axis xy

title('Axial Slice')

%%

% Dùng load để load dữ liệu file brain\_mask

load brain\_mask

% hiển thị nội dung bộ nhớ

whos

% lấy kích thước mask m

size(m)

% lấy kiểu dữ liệu trong biến m

class(m)

% lấy loại dữ liện trong biến mri

class(mri)

% đổi mri sang mảng double

mri\_new = double(mri);

% hiển thị nội dung bộ nhớ

whos

% lấy loại dữ liện trong biến mri\_new

class(mri\_new)

% xác nhận mask cho ảnh mri

masked\_mri = mri\_new.\*m;

% hiển thị 3 lớp cắt não

% tạo cửa sổ figure mới

figure

subplot(2,2,1)

masked\_coronal\_slice = masked\_mri(:,128,:);

masked\_coronal\_slice = squeeze(masked\_coronal\_slice);

imagesc(masked\_coronal\_slice');

axis xy

colormap(gray)

title('Masked Coronal Slice')

% hiển thị lớp sagittal slice

subplot(2,2,2)

masked\_sagittal\_slice = masked\_mri(128,:,:);

masked\_sagittal\_slice = squeeze(masked\_sagittal\_slice);

imagesc(masked\_sagittal\_slice');

axis xy

colormap(gray)

title('Masked Sagittal Slice')

% hiển thị lớp axial slice

subplot(2,2,3)

masked\_axial\_slice = masked\_mri(:,:,95);

masked\_axial\_slice = squeeze(masked\_axial\_slice);

imagesc(masked\_axial\_slice');

axis xy

colormap(gray)

title('Masked Axial Slice')

**Kết quả:**

**Name Size Bytes Class Attributes**

**BW 256x190 48640 logical**

**I 256x190 97280 int16**

**ans 1x6 12 char**

**axial\_slice 256x256 131072 int16**

**coronal\_slice 256x190 97280 int16**

**m 256x256x190 99614720 double**

**masked\_axial\_slice 256x256 524288 double**

**masked\_coronal\_slice 256x190 389120 double**

**masked\_mri 256x256x190 99614720 double**

**masked\_sagittal\_slice 256x190 389120 double**

**mri 256x256x190 24903680 int16**

**mri\_new 256x256x190 99614720 double**

**sagittal\_slice 256x190 97280 int16**

**ans =**

**256 1 190**

**ans =**

**256 190**

**Name Size Bytes Class Attributes**

**BW 256x190 48640 logical**

**I 256x190 97280 int16**

**ans 1x2 16 double**

**axial\_slice 256x256 131072 int16**

**coronal\_slice 256x190 97280 int16**

**m 256x256x190 99614720 double**

**masked\_axial\_slice 256x256 524288 double**

**masked\_coronal\_slice 256x190 389120 double**

**masked\_mri 256x256x190 99614720 double**

**masked\_sagittal\_slice 256x190 389120 double**

**mri 256x256x190 24903680 int16**

**mri\_new 256x256x190 99614720 double**

**sagittal\_slice 256x190 97280 int16**

**ans =**

**256 256 190**

**ans =**

**double**

**ans =**

**int16**

**Name Size Bytes Class Attributes**

**BW 256x190 48640 logical**

**I 256x190 97280 int16**

**ans 1x5 10 char**

**axial\_slice 256x256 131072 int16**

**coronal\_slice 256x190 97280 int16**

**m 256x256x190 99614720 double**

**masked\_axial\_slice 256x256 524288 double**

**masked\_coronal\_slice 256x190 389120 double**

**masked\_mri 256x256x190 99614720 double**

**masked\_sagittal\_slice 256x190 389120 double**

**mri 256x256x190 24903680 int16**

**mri\_new 256x256x190 99614720 double**

**sagittal\_slice 256x190 97280 int16**

**ans =**

**double**

****

****

Task 1

load raw\_mri

I= mri(:,128,:);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load raw\_mri

I= mri(128,:,:);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load raw\_mri

I= mri(:,:,128);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load raw\_mri

I= mri(:,:,95);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load brain\_mask

load raw\_mri

class(m)

class(mri)

mri\_new = double(mri);

class(mri\_new)

masked\_mri = mri\_new.\*m;

I= masked\_mri(:,128,:);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load brain\_mask

load raw\_mri

class(m)

class(mri)

mri\_new = double(mri);

class(mri\_new)

masked\_mri = mri\_new.\*m;

I= masked\_mri(128,:,:);

I= squeeze(I);

BW = edge(I);

imshow(BW)



load brain\_mask

load raw\_mri

class(m)

class(mri)

mri\_new = double(mri);

class(mri\_new)

masked\_mri = mri\_new.\*m;

I= masked\_mri(:,:,128);

I= squeeze(I);

BW = edge(I);

imshow(BW)



# XỬ LÝ ẢNH Y TẾ - PHẦN 2

Phần 2, cung cấp các kiến thức:

(1) load Xray Image data into MATLAB

(2) display Xray Image

(3) apply a filter to extract Xray Lung area

(4) display lung area

Sinh viên thực hiện theo hướng dẫn, ghi nhận kết qủa từng lệnh và giải thích, thực hiện việc phát hiện vùng phổi còn lại

w =(1/9)\*[1 1 1 , 1 1 1 , 1 1 1];%tạo mask

f = imread('JPCLN018.bmp');

imshow('JPCLN018.bmp'); %xuất ảnh ra



%Dùng imadjust để kéo dài mức xám

j = imadjust(f,[],[1; 0]);% Chuyển đổi hình ảnh thành Double sau đó thay đổi kích thước nó thành 512 \* 512 (Điều này làm giảm thời gian xử lý)

ggg=im2double(j);%đổi sang double

L = imresize(ggg,[512 512]);% Thay đổi kích thước hình ảnh thành (512 \* 512)

L11=L(1:512,1:255);

L12=L(1:512,257:512);

w= fspecial('gaussian',99,16); %Dùng bộ lọc gaussian

[x,y]=size(L);

L2=L.^2; % bình phương ảnh gốc

L\_blurred=imfilter(L,w); %lọc ảnh gốc

L2\_blurred=imfilter(L2,w); %lọc ảnh đã bình phương

L\_blurred\_2=L\_blurred.^2; %lọc ảnh của phần lọc ảnh gốc đã bình phương

t=(L-L\_blurred)./((sqrt(L2\_blurred-L\_blurred\_2))); %thuật toán

t=mat2gray(t);

smask = fspecial('gaussian', ceil(3\*.9), .9);

t = filter2(smask, t, 'same');

%imtool(t) để xem ảnh sau khi lọc



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%% biểu đồ HIST thứ nhất %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

claheI = adapthisteq(t,'NumTiles',[40 40]);%dùng bộ lọc NumTiles

claheI = imadjust(claheI);

%imtool(claheI);



z1=immultiply(claheI,t); % immultiltiply để nhân 2 ảnh claheI với t

%imtool(z1);



z11=z1(1:512,1:256);

z12=z1(1:512,257:512);

me1=mean2(z1);%mean của ảnh

st1=std2(z1);%độ lệch chuẩn của ảnh

gg11=(z1-me1)\*(1/st1);%thuật toán để phân đoạn lại ảnh

%imtool(gg11)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%% biểu đồ HIST thứ hai %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

claheI = adapthisteq(t,'NumTiles',[127 20]);%, dùng bộ lọc NumTiles

claheI = imadjust(claheI);

%imtool(claheI);



z2=immultiply(claheI,t);

%imtool(z2);



z21=z2(1:512,1:256);

z22=z2(1:512,257:512);

me1=mean2(z2);

st1=std2(z2);

gg2=(z2-me1)\*(1/st1);%tương tự ở biểu đồ Hist thứ nhất

%imtool(gg2)



%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%% Giảm biên độ sáng%%%%%%%%%%%%%%%%%%%%

%%%%%%%%%%%%%%%%%%%%

% IM2 = imclearborder(gg2)

% imtool(IM2)



%%%%%%%%%%%%%% ACTIVE COUNTOR %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

mask = zeros(size(z11)); %tạo mask ma trận 0 với độ lớn bằng với z11

mask(55:end-55,75:end-75) = 1; %

bw1 = activecontour(z11,mask,500);

mask = zeros(size(z21));

mask(55:end-55,75:end-75) = 1;

bw2 = activecontour(z21,mask,500);

bw=bw1+bw2;

%imtool (bw)



se1 = strel('disk',1);

erodedBW = imerode(bw,se1);

closeBW = imclose(erodedBW,se1);

%%%%Khu vực phổi (ảnh phổi) %%%%%%%%%%%%%%%%%%%%%%%%%%%%

f = inline('sum(x(:)) >= 3'); %tạoo thuật toán

lut = makelut(f,3); %lập bảng tra cứu

BW2 = applylut(bw,lut); %biên dịch ảnh bằng bảng tra cứu

L2 = bwlabel(BW2); %gắn nhãn cho ảnh

stats2 = regionprops(L2,'all');

%%% tìm khu vực ảnh phổi có điểm ảnh lớn hơn 20000 pixels

idx2 = find([stats2.Area] > 20000);

BW2 = ismember(L2,idx2);

%figure, imshow(BW2)%%% ảnh phổi không có nốt



small\_lung\_R = imfill(BW2,'holes');

%imtool(small\_lung\_R)



% sob = edge(small\_lung\_R, 'canny', (graythresh(z21) \* .1));

% BWoutline = bwperim(sob);

%%figure, imshow(BWoutline)



%I2 = imfill(small\_lung\_R,'holes');

% tô ảnh vị trí phổi xác định được

I2=small\_lung\_R;

se = strel('line',11,90);

bw2 = imdilate(I2,se);

sob = edge(bw2, 'canny', (graythresh(z21) \* .1));

BWoutline = bwperim(sob); %tạo đường kẻ trên xác định vị trí phổi

figure, imshow(I2);



Segout1 = L11;

Segout1(BWoutline) =0;

imtool(Segout1)

