**BỘ GIÁO DỤC VÀ ĐÀO TẠO**

**ĐẠI HỌC CÔNG NGHỆ TP.HCM**

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

SV :

Lớp :

www.hutech.edu.vn

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

Ấn bản 2019

MỤC LỤC

[MỤC LỤC 1](#_Toc60485346)

[BÀI 1: CƠ SỞ VỀ XỬ LÝ ẢNH 2](#_Toc60485347)

[BÀI 2: XỬ LÝ ẢNH NÂNG CAO 3](#_Toc60485348)

[BÀI 3: XỬ LÝ ẢNH Y TẾ - PHẦN 1 4](#_Toc60485349)

[BÀI 4: XỬ LÝ ẢNH Y TẾ - PHẦN 2 11](#_Toc60485350)

# CƠ SỞ VỀ XỬ LÝ ẢNH

**Bài 1.1** Đọc ảnh:

1. Code
2. Kết quả

**Bài 1.2**

1. Code
2. Kết quả

**Bài 1.3**

**Bài 1.4**

**Bài 1.5**

# XỬ LÝ ẢNH NÂNG CAO

**Bài 2.1**

**Bài 2.2**

**Bài 2.3**

**Bài 2.4**

**Bài 2.5**

**Bài 2.6**

**Bài 2.7**

**Bài 2.8**

**Bài 2.9**

**Bài 2.10**

# 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.

# 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];%avarage mask

f = imread('JPCLN018.bmp');

imshow('JPCLN018.bmp');

%Applay Imadjust to image (Stretching gray level)

j = imadjust3(f,[],[1; 0]);%//get the negative%%%%%%%%%%%%%%%%5

%%%Convert Image to Double then resize it to

%512\*512 (This reduce the processing time)

ggg=im2double(j);%covert to double

L = imresize(ggg,[512 512]);% Resize image to (512\*512)102013(File)

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

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

w= fspecial('gaussian',99,16); %filtro gaussiano de sigma 16

[x,y]=size(L);

%local normalization

L2=L.^2; %imagem original ao quadrado

L\_blurred=imfilter(L,w); %imagem original filtrada

L2\_blurred=imfilter(L2,w); %imagem original ao quadrado filtrada

L\_blurred\_2=L\_blurred.^2; %imagem original filtrada ao quadrado

t=(L-L\_blurred)./((sqrt(L2\_blurred-L\_blurred\_2))); % expressão da local normalization

t=mat2gray(t);

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

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

%imtool(t)

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

%%%%%%%%%%%%% FIRST HIST EQULIZATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

claheI = adapthisteq(t,'NumTiles',[40 40]);%,'Distribution' ,'rayleigh');

claheI = imadjust(claheI);

%imtool(claheI);

z1=immultiply(claheI,t);

%imtool(z1);

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

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

me1=mean2(z1);%mean of image

st1=std2(z1);%standerd deviation of image

gg11=(z1-me1)\*(1/st1);%%normalized image to segment the lung

%imtool(gg11)

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

%%%%%%%%%%%%% SECOND HIST EQULIZATION %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

claheI = adapthisteq(t,'NumTiles',[127 20]);%,'Distribution' ,'rayleigh');

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);%mean of image

st1=std2(z2);%standerd deviation of image

gg2=(z2-me1)\*(1/st1);%%normalized image to segment the lung

%imtool(gg2)

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

%%%%%%%%%%%%%%%%%%%% REDUCE LIGHTER SPOTS BORDER%%%%%%%%%%%%%%%%%%%%

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

% IM2 = imclearborder(ss)

% imtool(IM2)

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

mask = zeros(size(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);

%%%%Lung Area (Lung Image) %%%%%%%%%%%%%%%%%%%%%%%%%%%%

f = inline('sum(x(:)) >= 3');

lut = makelut(f,3);

BW2 = applylut(bw,lut);

L2 = bwlabel(BW2);

stats2 = regionprops(L2,'all');

%%% Serech for lung area larger than 20000 pixels

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

BW2 = ismember(L2,idx2);

%figure, imshow(BW2)%%% lung without nod

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');

I2=small\_lung\_R;

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

bw2 = imdilate(I2,se);

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

BWoutline = bwperim(sob);

figure, imshow(I2);

Segout1 = L11;

Segout1(BWoutline) =0;

imtool(Segout1)