****

**Course Information**

Course Title: Digital Image Processing

Section: 2

Course Instructor: Dr. Ahmed Wasif Reza

Associate Professor

Department of Computer Science & Engineering

**Lab-06**

**Student’s Information**

**Name:** Aahadul Islam Fardin

**ID:** 2019-1-60-224

**Department:** Computer Science & Engineering

**Date of Submission: 09 May 2023**

1. Segment the tumor from Figure 1 by using:

i. Region growing approach

function J=L6011(I,x,y,reg\_maxdist)

if(exist('reg\_maxdist','var')==0), reg\_maxdist=0.2; end

if(exist('y','var')==0), figure, imshow(I,[]); [y,x]=getpts; y=round(y(1)); x=round(x(1)); end

J = zeros(size(I));

Isizes = size(I);

reg\_mean = I(x,y);

reg\_size = 1;

neg\_free = 10000; neg\_pos=0;

neg\_list = zeros(neg\_free,3);

pixdist=0;

neigb=[-1 0; 1 0; 0 -1;0 1];

diff = 01;

while(pixdist<reg\_maxdist && reg\_size<numel(I) && diff ~=0)

num1 = sum(sum(reg\_size));

% Add new neighbors pixels

for j=1:4,

% Calculate the neighbour coordinate

xn = x +neigb(j,1); yn = y +neigb(j,2);

% Check if neighbour is inside or outside the image

ins=(xn>=1)&&(yn>=1)&&(xn<=Isizes(1))&&(yn<=Isizes(2));

% Add neighbor if inside and not already part of the segmented area

if(ins&&(J(xn,yn)==0))

neg\_pos = neg\_pos+1;

neg\_list(neg\_pos,:) = [xn yn I(xn,yn)]; J(xn,yn)=1;

end

end

if(neg\_pos+10>neg\_free), neg\_free=neg\_free+10000; neg\_list((neg\_pos+1):neg\_free,:)=0; end

dist = abs(neg\_list(1:neg\_pos,3)-reg\_mean);

[pixdist, index] = min(dist);

J(x,y)=2; reg\_size=reg\_size+1;

reg\_mean= (reg\_mean\*reg\_size + neg\_list(index,3))/(reg\_size+1);

x = neg\_list(index,1); y = neg\_list(index,2);

neg\_list(index,:)=neg\_list(neg\_pos,:); neg\_pos=neg\_pos-1;

num2 = sum(sum(reg\_size));

diff = num2-num1;

end

J=J>1;

%1. Segment the tumor from Region growing approach

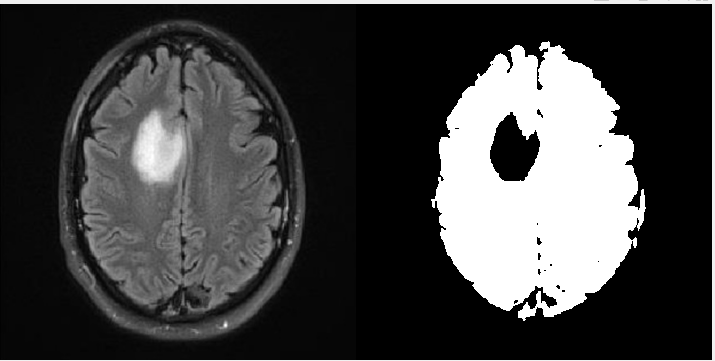
I = rgb2gray(im2double(imread('Tumor.png')));

J=imadjust(I,[],[],0.5);

J=imgaussfilt(J);

J= regiongrowing(J,0.2);

imshowpair(I,J,'montage')



ii. Region Splitting and Merging approach

function g = L6012(f, mindim, fun)

Q = 2.^nextpow2(max(size(f)));

[M, N] = size(f);

f = padarray(f, [Q - M Q - N], 'post');

S = qtdecomp(f, @split\_test, mindim, fun);

Lmax = full(max(S(:)));

g = zeros(size(f));

MARKER = zeros(size(f));

for K = 1:Lmax

[vals, r, c] = qtgetblk(f, S, K);

if ~isempty(vals)

for I = 1:length(r)

xlow = r(I); ylow = c(I);

xhigh = xlow + K - 1; yhigh = ylow + K - 1;

region = f(xlow:xhigh, ylow:yhigh);

flag = feval(fun, region);

if flag

g(xlow:xhigh, ylow:yhigh) = 1;

MARKER(xlow, ylow) = 1;

end

end

end

end

g = bwlabel(imreconstruct(MARKER, g));

g = g(1:M, 1:N);

function v = split\_test(B, mindim, fun)

v(1:k) = false;

for I = 1:k

quadregion = B(:, :, I);

if size(quadregion, 1) <= mindim

v(I) = false;

continue

end

flag = feval(fun, quadregion);

if flag

v(I) = true;

end

end

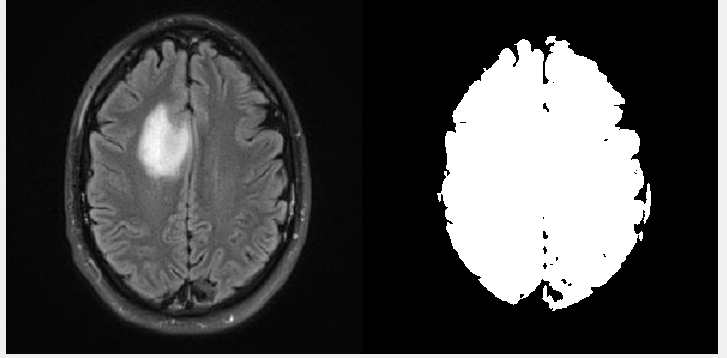
%split&merge

I2 = imread('Tumor.tif');

g = splitmerge(I2,8,@predicate);

figure,subplot(1,2,1);imshow(I2,[]);

subplot(1,2,2);imshow(g,[]);



1. Examine the differences between the methods from Question 2 to highlight their differences in quality and accessibility.

As shown in the segmented image provided in Question 1. The tumor is clearly segmented and emphasized in black form in the region growth approach. because it also produces a mask. The brain is well segmented in the split-merge procedure, but the tumor is not. So segmentation using the region-growing method is effective.

1. Segment the tumor from Figure 1 by using Marker Controlled Watershed segmentation.

%watershed

x=imbinarize(rgb2gray(imread('Tumor.png')));

subplot(2,2,1);

imshow(x);

title('Original Image');

a=x;

x=~x;

ms=bwdist(x);

ms=255-uint8(ms);

subplot(2,2,2);

imshow(ms);

title('Image after applying Distance Transformation');

hs=watershed(ms);

ws=hs==0;

subplot(2,2,3);

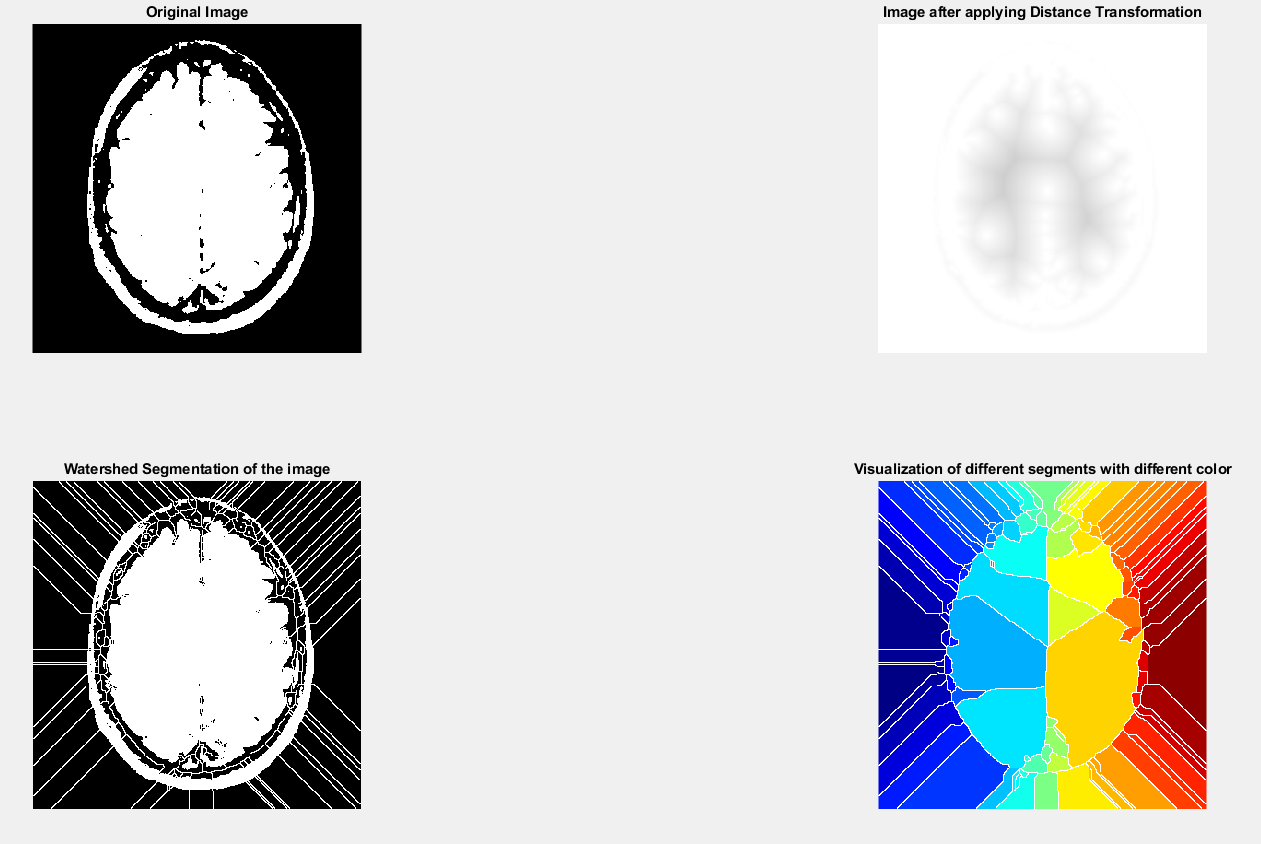
imshow(a | ws);

title('Watershed Segmentation of the image');

subplot(2,2,4);

imshow(label2rgb(hs));

title('Visualization of different segments with different color');



1. Segment the tumor from Figure 1 by using Quadtree Segmentation.

I = imread('Tumor.png');

I=rgb2gray(I);

Ifilt = medfilt2(I,[8 8]);

[S] = qtdecomp\_var(I,10);

[Sfilt] = qtdecomp\_var(Ifilt);

blocks = repmat(uint8(0),size(S));

for dim = [256 128 64 32 16 8 4 2 1];

numblocks = length(find(S==dim));

if (numblocks > 0)

values = repmat(uint8(1),[dim dim numblocks]);

values(2:dim,2:dim,:) = 0;

blocks = qtsetblk(blocks,S,dim,values);

end

end

blocks(end,1:end) = 1;

blocks(1:end,end) = 1;

blocks\_filt = repmat(uint8(0),size(Sfilt));

for dim = [128 64 32 16 8 4 2 1];

numblocks = length(find(Sfilt==dim));

if (numblocks > 0)

values = repmat(uint8(1),[dim dim numblocks]);

values(2:dim,2:dim,:) = 0;

blocks\_filt = qtsetblk(blocks\_filt,Sfilt,dim,values);

end

end

blocks\_filt(end,1:end) = 1;

blocks\_filt(1:end,end) = 1;

figure;subplot(221);imshow(I); title('Input image');

subplot(222);imshow(Ifilt); title('Filtered image');

subplot(223);imshow(blocks,[]);title('Quad tree input image with var weight');

subplot(224);imshow(blocks\_filt,[]);title('Quad tree filtered image without var weight');



1. Generate a binary mask of the tumor from Figure 1 using any segmentation method of your choice, then apply:

Binary mask:

rgb = imread('Tumor.png');

I = rgb2gray(rgb);

gmag = imgradient(I);

L = watershed(gmag);

Lrgb = label2rgb(L);

se = strel('disk',20);

Io = imopen(I,se);

Ie = imerode(I,se);

Iobr = imreconstruct(Ie,I);

Ioc = imclose(Io,se);

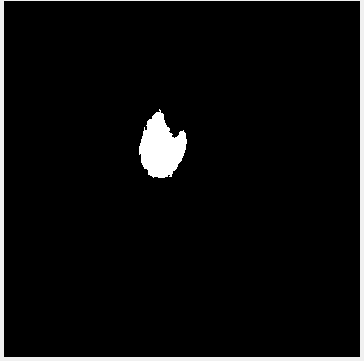
Iobrd = imdilate(Iobr,se);

Iobrcbr = imreconstruct(imcomplement(Iobrd),imcomplement(Iobr));

Iobrcbr = imcomplement(Iobrcbr);

fgm = imregionalmax(Iobrcbr);

figure,imshow(fgm)



i. Morphological Dilation

%dialation

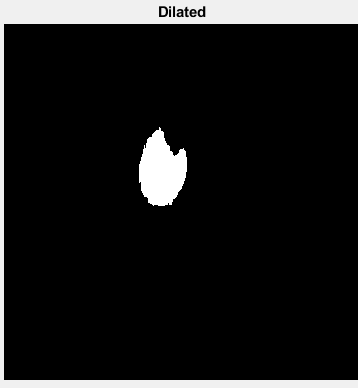
se = strel('line',11,90);

BW2 = imdilate(fgm,se);

imshow(BW), title('Original')

figure, imshow(BW2), title('Dilated')

figure,imshow(fgm)



ii. Morphological Erosion

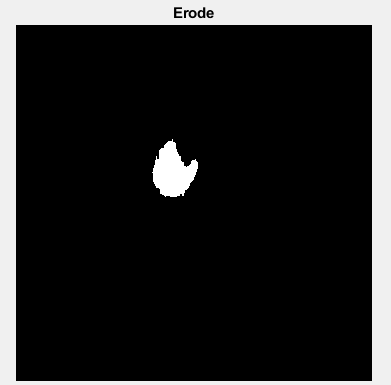
%erosion

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

erodedBW = imerode(fgm,se);

figure,imshow(erodedBW), title('Erode')

figure,imshow(fgm)



6.Apply Hough transform to Figure 2 and draw the detected lines on the original image.

RGB = imread('X-Ray Image.jpg');

I = im2gray(RGB);

BW = edge(I, 'canny');

[H,T,R] = hough(BW, 'RhoResolution', 0.5, 'Theta', -90:0.5:89);

subplot(2,1,1);

imshow(RGB);

title('X-Ray Image.jpg');

subplot(2,1,2);

imshow(imadjust(rescale(H)), 'XData', T, 'YData', R, ...

'InitialMagnification', 'fit');

title('Hough transform of X-Ray Image.jpg');

xlabel('\theta'), ylabel('\rho');

axis on, axis normal, hold on;

colormap(gca, hot);

A picture containing text, screenshot

Description automatically generated