A logo for a university

AI-generated content may be incorrect.

**Lab Report 9**

**Digital Image Processing**

**CSE438**

**Section:** 03

**Semester:** Spring-2025

**Submitted To:**

**Md. Asif Khan Rifat**

**Lecturer**

Department of Computer Science

and Engineering

**Submitted By:**

                                                                         Suddip Paul Arnab

**2022-1-60-356**

**Date of submission:** 22 May 2025

**Final Lab-09 (Assignment)**

1. Segment the tumor from Figure 1 by using:

i. Region growing approach

**Code**:

img = imread('Picture1.png');

gray\_img = im2gray(img);

gray\_img = double(gray\_img);

gray\_img = imgaussfilt(gray\_img, 1);

gray\_img = (gray\_img - min(gray\_img(:))) / (max(gray\_img(:)) - min(gray\_img(:)));

[M, N] = size(gray\_img);

figure('Name', 'Select Tumor Seed Point', 'NumberTitle', 'off');

imshow(gray\_img, []);

title('Click on the tumor to select a seed point, then press Enter');

[seed\_y, seed\_x] = ginput(1);

seed\_x = round(seed\_x(1));

seed\_y = round(seed\_y(1));

close(gcf);

seed\_intensity = gray\_img(seed\_y, seed\_x);

threshold = 0.2;

bw = false(M, N);

bw(seed\_y, seed\_x) = true;

stack = [seed\_y, seed\_x];

neighbors = [-1, -1; -1, 0; -1, 1; 0, -1; 0, 1; 1, -1; 1, 0; 1, 1];

while ~isempty(stack)

current = stack(end, :);

stack(end, :) = [];

y = current(1);

x = current(2);

for k = 1:size(neighbors, 1)

ny = y + neighbors(k, 1);

nx = x + neighbors(k, 2);

if ny >= 1 && ny <= M && nx >= 1 && nx <= N && ~bw(ny, nx)

if abs(gray\_img(ny, nx) - seed\_intensity) <= threshold

bw(ny, nx) = true;

stack = [stack; ny, nx];

end

end

end

end

se = strel('disk', 2);

bw = imopen(bw, se);

bw = imclose(bw, se);

bw = bwareaopen(bw, 30);

tumor\_outline = bwperim(bw);

overlay = imoverlay(gray\_img \* 255, tumor\_outline, [1 0 0]);

figure('Name', 'Tumor Segmentation Using Region Growing', 'NumberTitle', 'off');

subplot(1, 2, 1), imshow(img), title('Original Image');

subplot(1, 2, 2), imshow(bw), title('Tumor Mask');

disp('Tumor segmentation using Region Growing complete!');

**Output**:

A close-up of a brain scan

AI-generated content may be incorrect.

ii. Region Splitting and Merging approach

**Code**:

img = imread('Picture1.png');

gray\_img = im2gray(img);

gray\_img = double(gray\_img);

gray\_img = imgaussfilt(gray\_img, 1);

gray\_img = (gray\_img - min(gray\_img(:))) / (max(gray\_img(:)) - min(gray\_img(:)));

[M, N] = size(gray\_img);

min\_region\_size = 8;

split\_threshold = 0.1;

function should\_split = needSplit(img\_region, threshold)

if isempty(img\_region)

should\_split = false;

return;

end

region\_std = std(img\_region(:));

should\_split = region\_std > threshold;

end

regions = {[1, 1, M, N]};

homogeneous\_regions = {};

while ~isempty(regions)

current\_region = regions{1};

regions(1) = [];

r\_start = current\_region(1);

c\_start = current\_region(2);

height = current\_region(3);

width = current\_region(4);

img\_region = gray\_img(r\_start:r\_start+height-1, c\_start:c\_start+width-1);

if height <= min\_region\_size || width <= min\_region\_size || ~needSplit(img\_region, split\_threshold)

homogeneous\_regions{end+1} = current\_region;

else

h\_half = floor(height/2);

w\_half = floor(width/2);

regions{end+1} = [r\_start, c\_start, h\_half, w\_half];

regions{end+1} = [r\_start, c\_start+w\_half, h\_half, width-w\_half];

regions{end+1} = [r\_start+h\_half, c\_start, height-h\_half, w\_half];

regions{end+1} = [r\_start+h\_half, c\_start+w\_half, height-h\_half, width-w\_half];

end

end

region\_intensities = zeros(length(homogeneous\_regions), 1);

for i = 1:length(homogeneous\_regions)

region = homogeneous\_regions{i};

r\_start = region(1);

c\_start = region(2);

height = region(3);

width = region(4);

img\_region = gray\_img(r\_start:r\_start+height-1, c\_start:c\_start+width-1);

region\_intensities(i) = mean(img\_region(:));

end

figure('Name', 'Select Tumor Seed Point', 'NumberTitle', 'off');

imshow(gray\_img, []);

title('Click on the tumor to select a seed point, then press Enter');

[seed\_y, seed\_x] = ginput(1);

seed\_x = round(seed\_x);

seed\_y = round(seed\_y);

close(gcf);

seed\_region\_idx = -1;

for i = 1:length(homogeneous\_regions)

region = homogeneous\_regions{i};

r\_start = region(1);

c\_start = region(2);

height = region(3);

width = region(4);

if seed\_y >= r\_start && seed\_y < r\_start+height && ...

seed\_x >= c\_start && seed\_x < c\_start+width

seed\_region\_idx = i;

break;

end

end

seed\_intensity = region\_intensities(seed\_region\_idx);

merge\_threshold = 0.15;

tumor\_mask = false(M, N);

for i = 1:length(homogeneous\_regions)

region = homogeneous\_regions{i};

r\_start = region(1);

c\_start = region(2);

height = region(3);

width = region(4);

if abs(region\_intensities(i) - seed\_intensity) <= merge\_threshold

tumor\_mask(r\_start:r\_start+height-1, c\_start:c\_start+width-1) = true;

end

end

se = strel('disk', 2);

tumor\_mask = imopen(tumor\_mask, se);

tumor\_mask = imclose(tumor\_mask, se);

tumor\_mask = bwareaopen(tumor\_mask, 50);

CC = bwconncomp(tumor\_mask);

pixelIdxList = CC.PixelIdxList;

seed\_idx = sub2ind(size(tumor\_mask), seed\_y, seed\_x);

component\_with\_seed = -1;

for i = 1:CC.NumObjects

if any(pixelIdxList{i} == seed\_idx)

component\_with\_seed = i;

break;

end

end

final\_mask = false(M, N);

if component\_with\_seed > 0

final\_mask(pixelIdxList{component\_with\_seed}) = true;

else

num\_pixels = cellfun(@numel, pixelIdxList);

[~, largest\_idx] = max(num\_pixels);

final\_mask(pixelIdxList{largest\_idx}) = true;

end

tumor\_outline = bwperim(final\_mask);

rgb\_img = repmat(gray\_img, [1, 1, 3]);

overlay = rgb\_img;

overlay(:,:,1) = overlay(:,:,1) + tumor\_outline \* 0.7;

overlay(:,:,2) = overlay(:,:,2) - tumor\_outline \* 0.3;

overlay(:,:,3) = overlay(:,:,3) - tumor\_outline \* 0.3;

overlay = min(max(overlay, 0), 1);

figure('Name', 'Tumor Segmentation Using Region Splitting and Merging', 'NumberTitle', 'off');

subplot(2, 2, 1), imshow(img), title('Original Image');

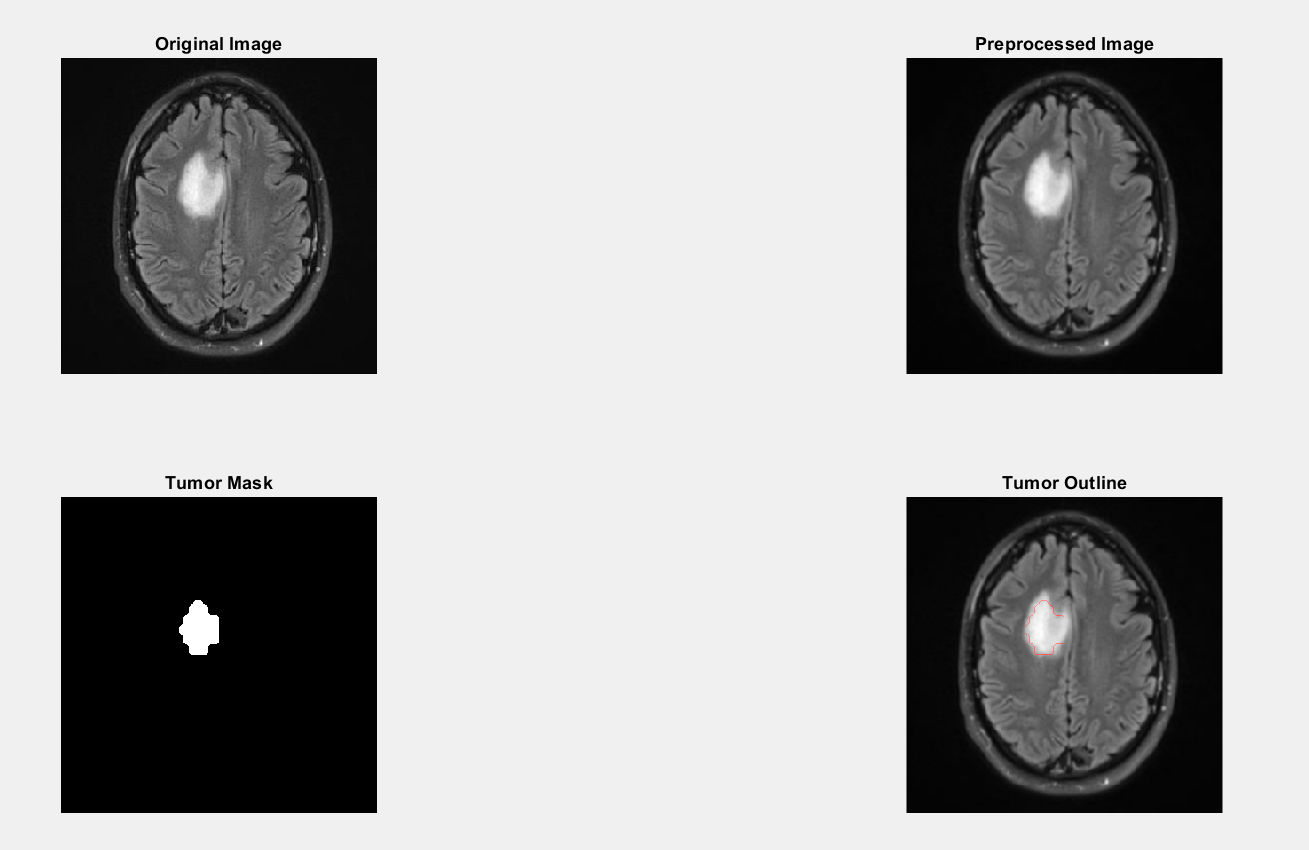
subplot(2, 2, 2), imshow(gray\_img, []), title('Preprocessed Image');

subplot(2, 2, 3), imshow(final\_mask), title('Tumor Mask');

subplot(2, 2, 4), imshow(overlay), title('Tumor Outline');

disp('Tumor segmentation using Region Splitting and Merging complete!');

**Output**:



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

**Code**:

img = imread('Picture1.png');

gray = rgb2gray(img);

gray\_filtered = medfilt2(gray, [3 3]);

gray\_eq = adapthisteq(gray\_filtered);

gmag = imgradient(gray\_eq);

se = strel('disk', 3);

Io = imopen(gray\_eq, se);

Ie = imerode(gray\_eq, se);

Iobr = imreconstruct(Ie, gray\_eq);

Iobrd = imdilate(Iobr, se);

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

Iobrcbr = imcomplement(Iobrcbr);

fgm = imregionalmax(Iobrcbr);

bw = imbinarize(Iobrcbr);

D = bwdist(~bw);

DL = watershed(D);

bgm = DL == 0;

gmag2 = imimposemin(gmag, bgm | fgm);

L = watershed(gmag2);

tumorMask = L == 0;

figure('Name', 'Marker-Controlled Watershed Tumor Segmentation');

subplot(1, 3, 1);

imshow(gray); title('Original MRI');

subplot(1, 3, 2);

imshow(gmag, []); title('Gradient Magnitude');

subplot(1, 3, 3);

imshow(gray); hold on;

h = imshow(tumorMask);

set(h, 'AlphaData', 0.5);

title('Watershed Tumor Segmentation');

**Output**:

A close-up of a brain

AI-generated content may be incorrect.

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

**Code**:

img = imread('Picture1.png');

gray = rgb2gray(img);

gray = medfilt2(gray, [3 3]);

gray\_resized = imresize(gray, [512 512]);

threshold\_std = 15;

predicate = @(block) std2(block) > threshold\_std;

S = qtdecomp(gray\_resized, predicate);

mask = zeros(size(gray\_resized));

block\_sizes = [512 256 128 64 32 16 8 4];

for k = 1:length(block\_sizes)

bsize = block\_sizes(k);

[vals\_x, vals\_y] = find(S == bsize);

for i = 1:length(vals\_x)

x = vals\_x(i); y = vals\_y(i);

if x + bsize - 1 <= size(gray\_resized,1) && y + bsize - 1 <= size(gray\_resized,2)

block = gray\_resized(x:x+bsize-1, y:y+bsize-1);

if mean(block(:)) > 100

mask(x:x+bsize-1, y:y+bsize-1) = 1;

end

end

end

end

mask = bwareaopen(mask, 50);

mask = imfill(mask, 'holes');

mask = imclose(mask, strel('disk', 3));

mask = imerode(mask, strel('disk', 1));

mask\_resized = imresize(mask, size(gray), 'nearest');

figure('Name', 'Quadtree Tumor Segmentation');

subplot(1, 2, 1);

imshow(gray); title('Original MRI Image');

subplot(1, 2, 2);

imshow(gray); hold on;

redMask = cat(3, ones(size(gray)), zeros(size(gray)), zeros(size(gray)));

h = imshow(redMask);

set(h, 'AlphaData', double(mask\_resized) \* 0.4);

boundaries = bwboundaries(mask\_resized);

for k = 1:length(boundaries)

boundary = boundaries{k};

plot(boundary(:,2), boundary(:,1), 'g', 'LineWidth', 1.5);

end

title('Tumor Segmentation using Quadtree');

**Output**:

A close-up of a brain scan

AI-generated content may be incorrect.