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

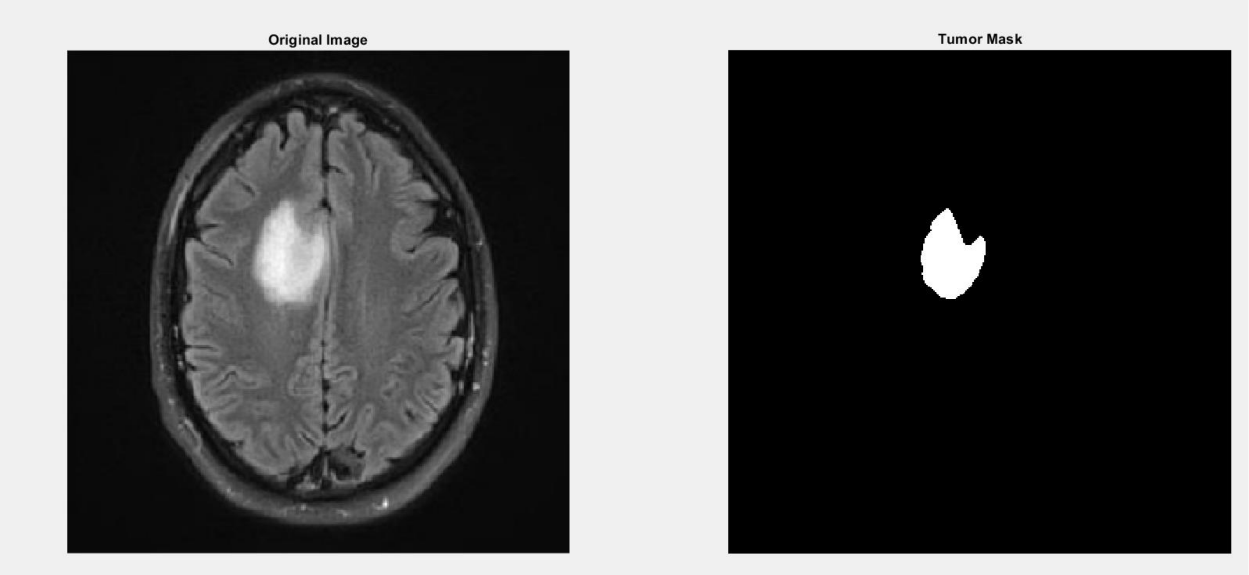
Final Lab-09 (Assignment)

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:



ii. Region Splitting and Merging approach

Code:

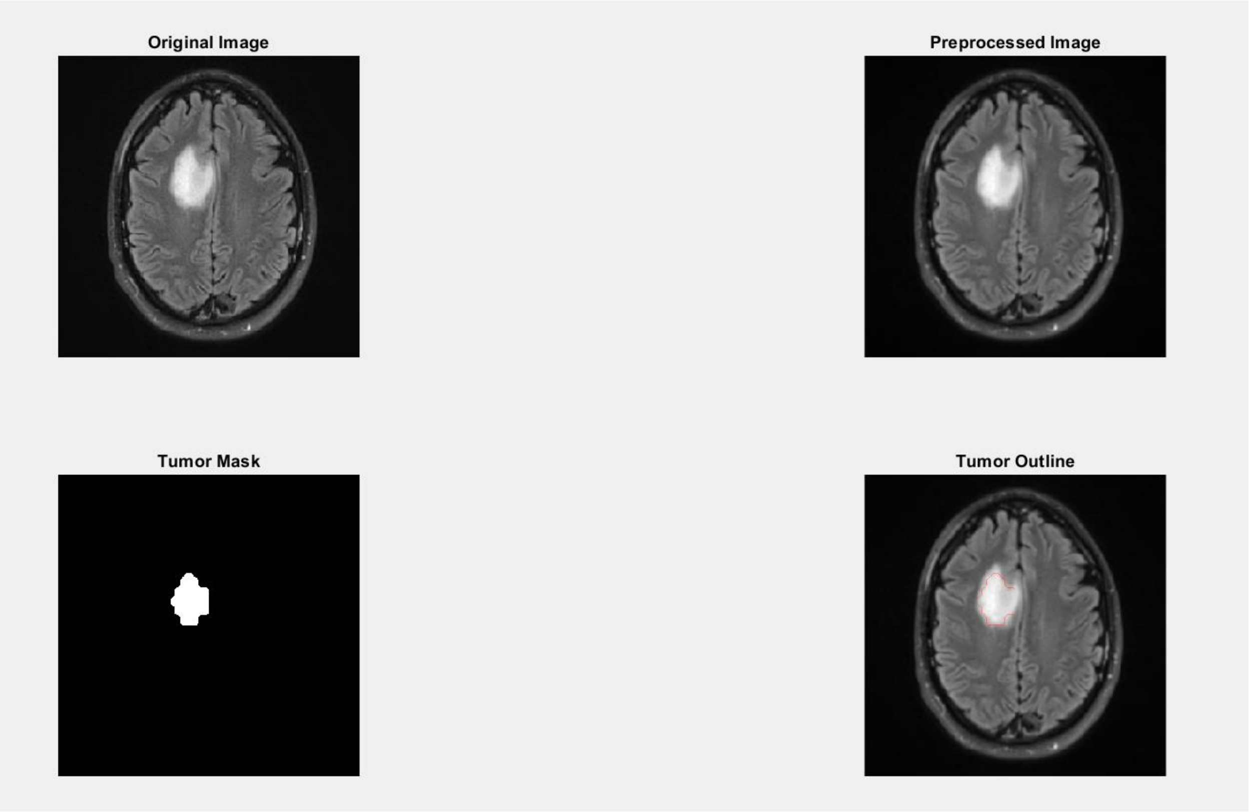
```
img = imread('Picture1.png');
gray_img = im2gray(img);
gray_img = double(gray_img);

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:

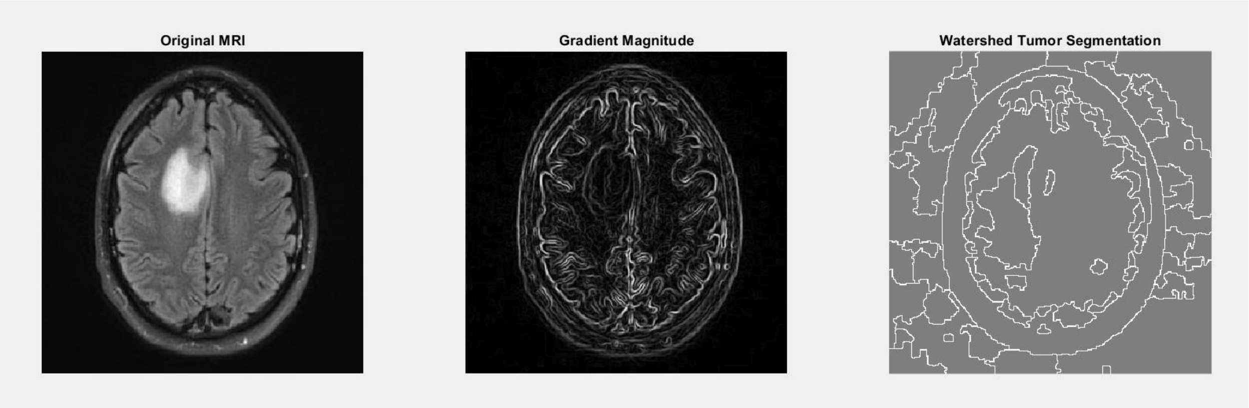


2. Segment the tumor from Figure 1 by using Marker Controlled Watershed segmentation.

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:

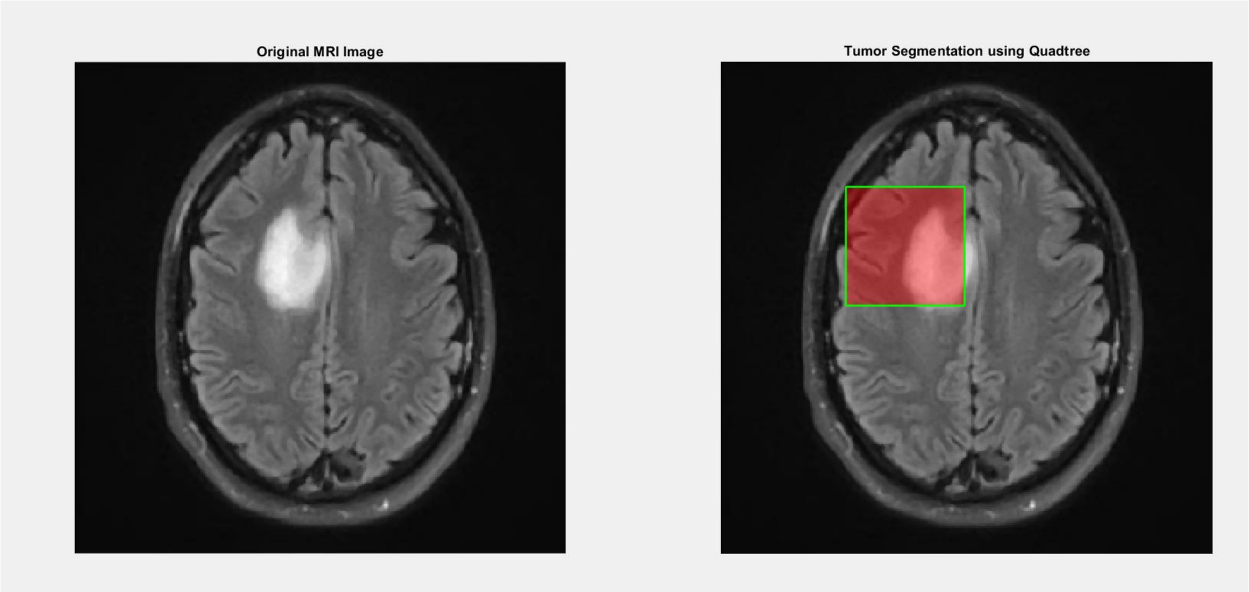


3. Segment the tumor from Figure 1 by using Quadtree Segmentation.

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:


Original MRI Image
Tumor Segmentation using Quadtree