Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

BRAIN SEGMENTATION USING MATLAB

ECE2006-DIGITAL SIGNAL PROCESSING

(Project Component)
Fall Semester: (2019 – 20)

Team Members:

NAME	REGISTRATION NUMBER
Muskan Jain	17BEC0339
Mishu Dave	17BEC0835

BRAIN SEGMENTATION USING MATLAB

A Project Report

Submitted in partial fulfillment of the requirement for the

evaluation of J component for the subject

ECE 2006 Digital Signal Processing

By

NAME	REG.NO	MOBILE NO
MUSKAN JAIN	17BEC0339	7570042562
MISHU DAVE	17BEC0835	9003306957

Faculty Incharge

Prof. Prakasam P

School of Electronics Engineering Vellore Institute of Technology, Vellore-632014

NOVEMBER 2019

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

DECLARATION

I hereby declare that the project work entitled "Brain Segmentation using MATLAB" submitted by our team for J component evaluation for the subject ECE 2006 Digital Signal Processing, the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Place : Vellore

NAME	REG.No.	MOBILE No.
MUSKAN JAIN	17BEC0339	7570042562
MISHU DAVE	17BEC0835	9003306957

Date: 08-11-2019

ACKNOWLEDGEMENT

We would like to take this opportunity to express our heartfelt gratitude to our faculty Prof. Prakasam P sir for his guidance, and instruction that he has given us to complete our thesis project in such a short time. We have been able to learn the skills from him, which have benefitted us immensely and will continue to help us throughout our future endeavors, both academically and professionally. I would also like to thank all of my group members for their collaboration, patience and co- ordination in the work with me which eventually bore fruits and gave proper shape to this report. Without their inputs and criticism this project would have been incomplete.

INDEX

Topic	Page number
Objective	4
Introduction	4
Anisotropic Diffusion	5
Algorithm	6
Matlab Code	11
Output	16
Conclusion	17
Future Scope	17

Objective

There are two main objectives of our project:

1) To detect if there is a tumor in a given MRI Image

2) If there is, specify the location of the tumor

Introduction: -

Brain segmentation is a procedure in which different MRI scans are processed and conclusions are derived from this. IN brain MRI, analysis image segmentation is commonly used for measuring and visualizing the brain's anatomical structures, for analysing brain changes, for delineating pathological regions, and for surgical planning and image-guided interventions. In the last few decades, various segmentation techniques of different accuracy and degree of complexity have been developed and reported in the literature.

Brain MRI segmentation is an essential task in many clinical applications because it influences the outcome of the entire analysis. This is because different processing steps rely on accurate segmentation of anatomical regions. For example, MRI segmentation is commonly used for measuring and visualizing different brain structures, for delineating lesions, for analysing brain development, and for image-guided interventions and surgical planning. This diversity of image processing applications has led to development of various segmentation techniques of different accuracy and degree of complexity.

The ideology/technique that we have adopted for Image processing is:

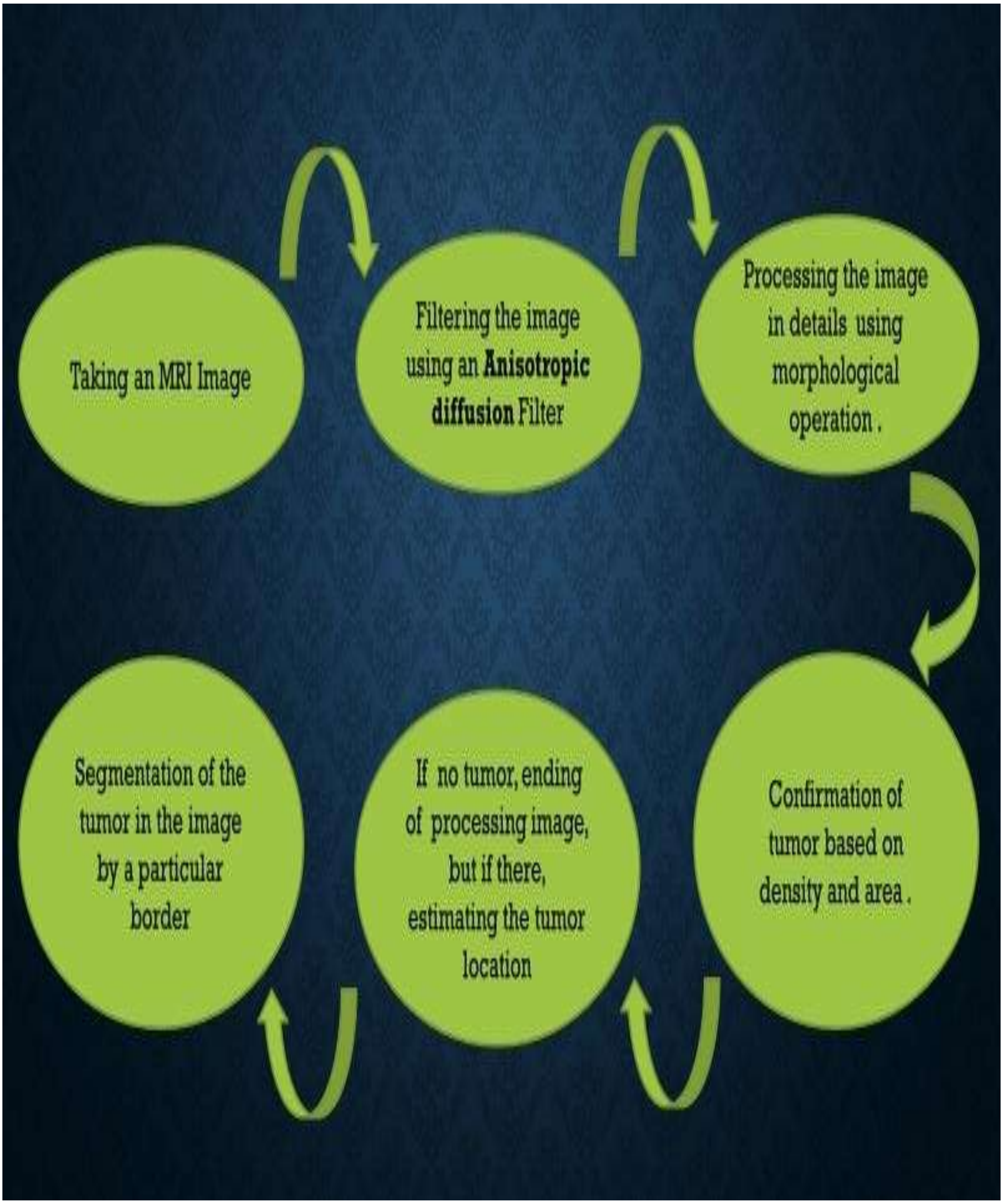
ANISOTROPIC DIFFUSION

- **Anisotropic diffusion**, also called **Perona–Malik diffusion**, is a technique aiming at reducing image noise without removing significant parts of the image content, typically edges, lines or other details that are important for the interpretation of the image.

- Anisotropic diffusion resembles the process that creates a scale space, where an image generates a parameterized family of successively more and more blurred images based on a diffusion process. Each of the resulting images in this family are given as a convolution between the image and a 2D isotropic Gaussian, where the width of the filter increases with the parameter.

- Anisotropic diffusion is a generalization of this diffusion process: it produces a family of parameterized images, but each resulting image is a combination between the original image and a filter that depends on the local content of the original image. As a consequence, anisotropic diffusion is a *non-linear* and *space-variant* transformation of the original image.

ALGORITHM adopted by us for the experiment: -



Taking an MRI Image

Filtering the image using an **Anisotropic diffusion** Filter

Processing the image in details using morphological operation .

Confirmation of tumor based on density and area .

If no tumor, ending of processing image, but if there, estimating the tumor location

Segmentation of the tumor in the image by a particular border

The diffusion equation is a general case of the heat equation that describes the density changes in a material undergoing diffusion over time. Isotropic diffusion, in image processing parlance, is an instance of the heat equation as a partial differential equation (PDE), given as:

$$\frac{\partial I}{\partial t} = \nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$$

where, I is the image and *t* is the time of evolution.

Perona & Malik introduce the flux function as a means to constrain the diffusion process to contiguous homogeneous regions, but not cross region boundaries. The heat equation (after appropriate expansion of terms) is thus modified to:

$$\frac{\partial I}{\partial t} = c(x, y, t)\Delta I + \nabla c \cdot \nabla I$$

where *c* is the proposed flux function which controls the rate of diffusion at any point in the image.

A choice of *c* such that it follows the gradient magnitude at the point enables us to restrain the diffusion process as we approach region boundaries. As we approach edges in the image, the flux function may trigger inverse diffusion and actually enhance the edges !

Perona & Malik suggest the following two flux functions:

$$c\left(||\nabla I||\right) = e^{-(||\nabla I||/K)^2}$$

$$c\left(||\nabla I||\right) = \frac{1}{1 + \left(\frac{||\nabla I||}{K}\right)^2}$$

- The flux functions offer a trade-off between edge-preservation and blurring (smoothing) homogeneous regions. Both the functions are governed by the free parameter κ which determines the edge-strength to consider as a valid region boundary. Intuitively, a large value of κ will lead back into an isotropic-like solution.

We will experiment with both the flux functions in this report.

A discrete numerical solution can be derived for the anisotropic case using the FTCS method as follows:

$$I_{i,j}^{t+1} = I_{i,j}^{t} + \lambda \left[c_N \cdot \nabla_N I + c_S \cdot \nabla_S I + c_E \cdot \nabla_E I + c_W \cdot \nabla_W I \right]_{i,j}^{t}$$

where {N,S,W,E} correspond to the pixel above, below, left and right of the pixel under consideration (i,j).

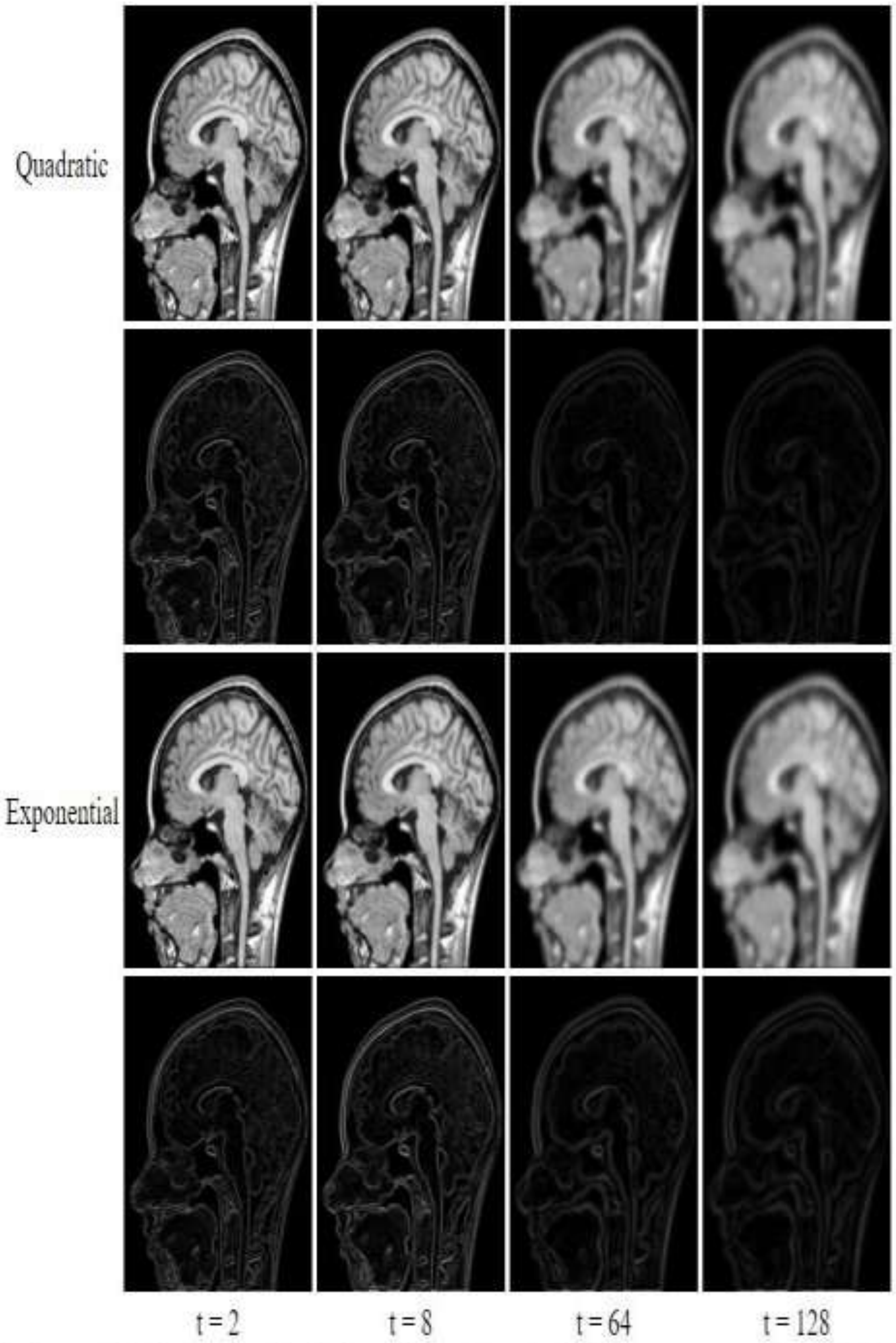
- Another nice property is that *c* is based on the gradient magnitude, thus it does not matter if we take forward or backward gradients ! The following table displays the results of th anisotropic diffusion process for the same example image as above. κ=0.35 was used for both flux functions.

The following points can be concluded from anisotropic diffusion: -

- Various experiments with nonlinear scale-space suggest that it can be a powerful tool for various applications.

- However, linear (Gaussian) scale-spaces are not far behind. The inherent properties of the Gaussian kernel make it convenient to use and easy to understand.

- On the other hand, nonlinear scale-spaces allow for the inclusion of knowledge about local criteria such as edge-strength to smooth homogeneous regions, while preserving the boundaries.



As seen in the above table, the flux functions behave differently as the diffusion progresses and can lead to interesting choices based on the application at hand.

MATLAB CODE:

(Function for anisotropicdiffusion)

```
function diff_im = anisodiff(im, num_iter, delta_t, kappa, option)
fprintf('Removing noise\n');


fprintf('Filtering Completed.');

% Convert input image to double.
im = double(im);

% PDE (partial differential equation) initial condition.
diff_im = im;

% Center pixel distances.
dx = 1;
dy = 1;
dd = sqrt(2);


% 2D convolution masks - finite differences.
hN = [0 1 0; 0 -1 0; 0 0 0];
hS = [0 0 0; 0 -1 0; 0 1 0];
hE = [0 0 0; 0 -1 1; 0 0 0];
hW = [0 0 0; 1 -1 0; 0 0 0];
hNE = [0 0 1; 0 -1 0; 0 0 0];
hSE = [0 0 0; 0 -1 0; 0 0 1];
hSW = [0 0 0; 0 -1 0; 1 0 0];
hNW = [1 0 0; 0 -1 0; 0 0 0];


% Anisotropic diffusion.
for t = 1:num_iter


        % Finite differences. [imfilter(.,.,'conv') can be replaced by
conv2(.,.,'same')]

        nablaN = imfilter(diff_im,hN,'conv');
        nablaS = imfilter(diff_im,hS,'conv');
        nablaW = imfilter(diff_im,hW,'conv');
        nablaE = imfilter(diff_im,hE,'conv');
        nablaNE = imfilter(diff_im,hNE,'conv');
        nablaSE = imfilter(diff_im,hSE,'conv');
        nablaSW = imfilter(diff_im,hSW,'conv');
        nablaNW = imfilter(diff_im,hNW,'conv');

        % Diffusion function.

        if option == 1
            cN = exp(-(nablaN/kappa).^2);
            cS = exp(-(nablaS/kappa).^2);
            cW = exp(-(nablaW/kappa).^2);
            cE = exp(-(nablaE/kappa).^2);
            cNE = exp(-(nablaNE/kappa).^2);
            cSE = exp(-(nablaSE/kappa).^2);
            cSW = exp(-(nablaSW/kappa).^2);
            cNW = exp(-(nablaNW/kappa).^2);
        elseif option == 2

            cN = 1./(1 + (nablaN/kappa).^2);
            cS = 1./(1 + (nablaS/kappa).^2);
            cW = 1./(1 + (nablaW/kappa).^2);
            cE = 1./(1 + (nablaE/kappa).^2);
            cNE = 1./(1 + (nablaNE/kappa).^2);
            cSE = 1./(1 + (nablaSE/kappa).^2);
            cSW = 1./(1 + (nablaSW/kappa).^2);
            cNW = 1./(1 + (nablaNW/kappa).^2);
        end

        % Discrete PDE solution.

        diff_im = diff_im + ...
                  delta_t*(...
                  (1/(dy^2))*cN.*nablaN + (1/(dy^2))*cS.*nablaS + ...
                  (1/(dx^2))*cW.*nablaW + (1/(dx^2))*cE.*nablaE + ...
                  (1/(dd^2))*cNE.*nablaNE + (1/(dd^2))*cSE.*nablaSE + ...
                  (1/(dd^2))*cSW.*nablaSW + (1/(dd^2))*cNW.*nablaNW );



end
```

Main program: -

```
clc;
close all;
clear all;

%% Input
[I,path]=uigetfile('*.jpg','select a input image');
str=strcat(path,I);
s=imread(str);

figure;
imshow(s);
title('Input image','FontSize',20);

%% Filter
num_iter = 10;
    delta_t = 1/7;
    kappa = 15;
    option = 2;
    disp('Preprocessing image please wait . . .');
    inp = anisodiff(s,num_iter,delta_t,kappa,option);
    inp = uint8(inp);

inp=imresize(inp,[256,256]);
if size(inp,3)>1
    inp=rgb2gray(inp);
```

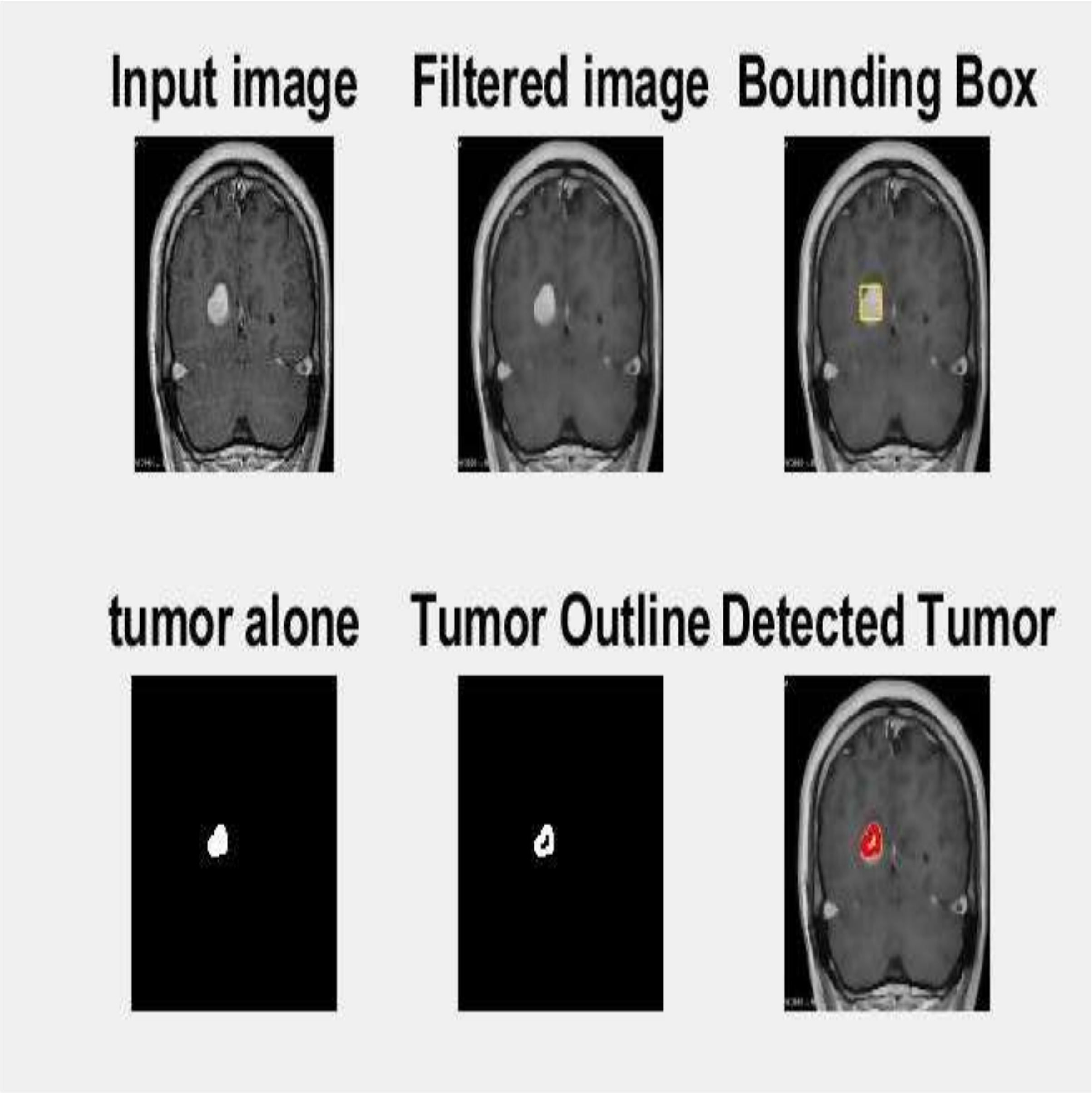
```
end
figure;
imshow(inp);
title('Filtered image','FontSize',20);

%% thresholding
sout=imresize(inp,[256,256]);
t0=60;
th=t0+((max(inp(:))+min(inp(:)))./2);
for i=1:1:size(inp,1)
    for j=1:1:size(inp,2)
        if inp(i,j)>th
            sout(i,j)=1;
        else
            sout(i,j)=0;
        end
    end
end


%% Morphological Operation


label=bwlabel(sout);
stats=regionprops(logical(sout),'Solidity','Area','BoundingBox');
density=[stats.Solidity];
area=[stats.Area];
high_dense_area=density>0.6;
max_area=max(area(high_dense_area));
tumor_label=find(area==max_area);
tumor=ismember(label,tumor_label);

if max_area>100
   figure;
   imshow(tumor)
   title('tumor alone','FontSize',20);
else
    h = msgbox('No Tumor!!','status');
    %disp('no tumor');
    return;
end

%% Bounding box

box = stats(tumor_label);
wantedBox = box.BoundingBox;
figure
imshow(inp);
title('Bounding Box','FontSize',20);
hold on;
rectangle('Position',wantedBox,'EdgeColor','y');
hold off;



%% Getting Tumor Outline - image filling, eroding, subtracting
% erosion the walls by a few pixels
```

```
dilationAmount = 5;
rad = floor(dilationAmount);
[r,c] = size(tumor);
filledImage = imfill(tumor, 'holes');

for i=1:r

   for j=1:c

       x1=i-rad;
       x2=i+rad;
       y1=j-rad;
       y2=j+rad;
       if x1<1
           x1=1;
       end

       if x2>r
           x2=r;
       end
       if y1<1
           y1=1;
       end

       if y2>c
           y2=c;
       end

       erodedImage(i,j) = min(min(filledImage(x1:x2,y1:y2)));
   end

end

figure
imshow(erodedImage);
title('eroded image','FontSize',20);


%% subtracting eroded image from original BW image


tumorOutline=tumor;
tumorOutline(erodedImage)=0;

figure;
imshow(tumorOutline);
title('Tumor Outline','FontSize',20);


%% Inserting the outline in filtered image in green color


rgb = inp(:,:,[1 1 1]);
red = rgb(:,:,1);

red(tumorOutline)=255;
green = rgb(:,:,2);
green(tumorOutline)=0;
```

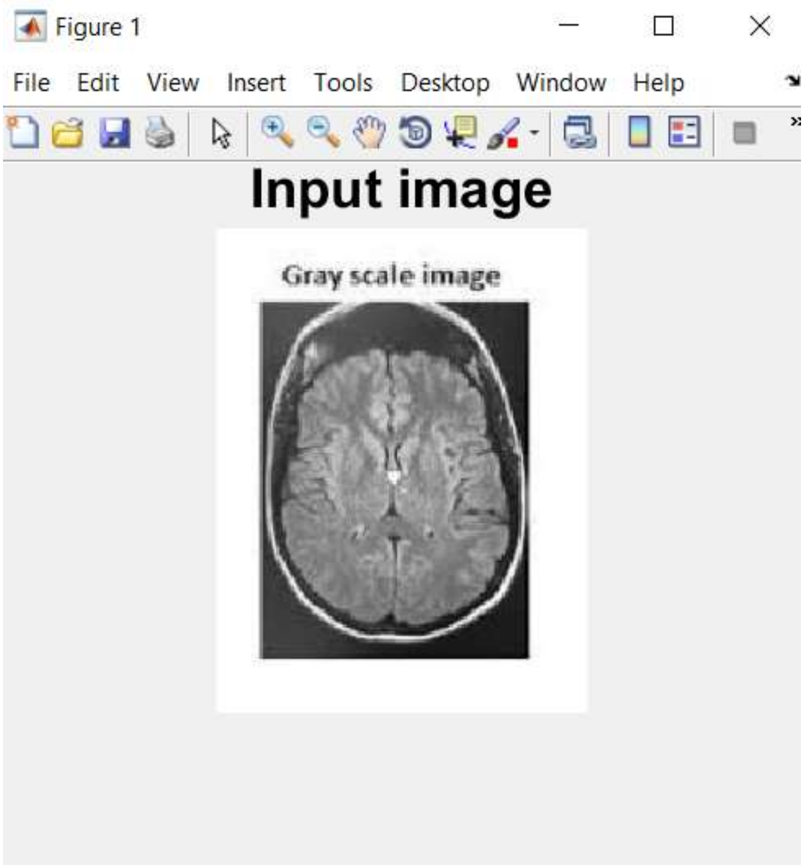
```
blue = rgb(:,:,3);
blue(tumorOutline)=0;

tumorOutlineInserted(:,:,1) = red;
tumorOutlineInserted(:,:,2) = green;
tumorOutlineInserted(:,:,3) = blue;


figure

imshow(tumorOutlineInserted);
title('Detected Tumer','FontSize',20);

%% Display Together


figure
subplot(231);imshow(s);title('Input image','FontSize',20);
subplot(232);imshow(inp);title('Filtered image','FontSize',20);


subplot(233);imshow(inp);title('Bounding Box','FontSize',20);
hold on;rectangle('Position',wantedBox,'EdgeColor','y');hold off;



subplot(234);imshow(tumor);title('tumor alone','FontSize',20);
subplot(235);imshow(tumorOutline);title('Tumor Outline','FontSize',20);
subplot(236);imshow(tumorOutlineInserted);title('Detected
Tumor','FontSize',20);
```

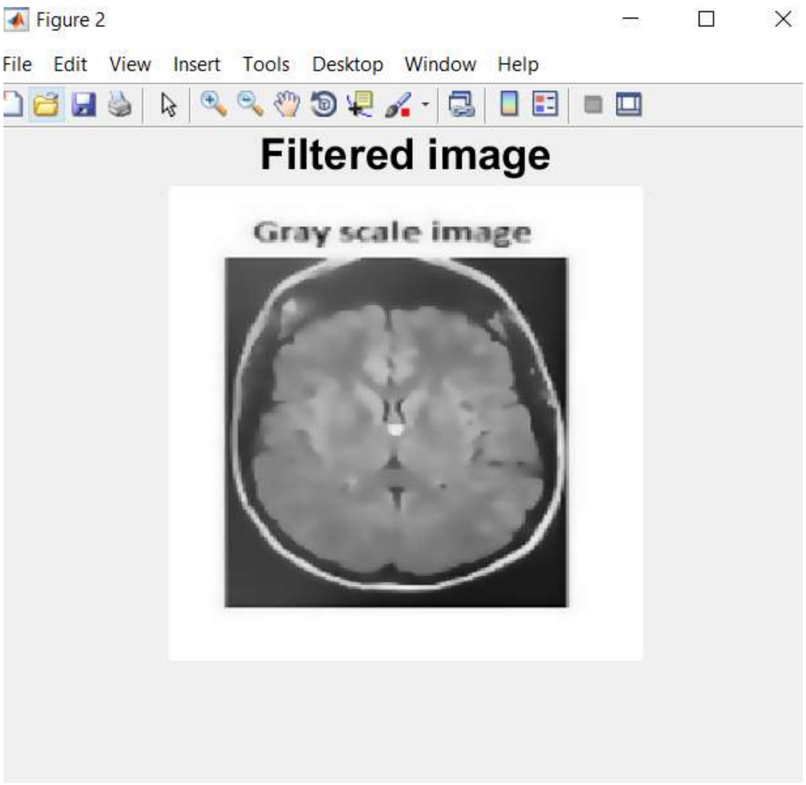
OUTPUT:

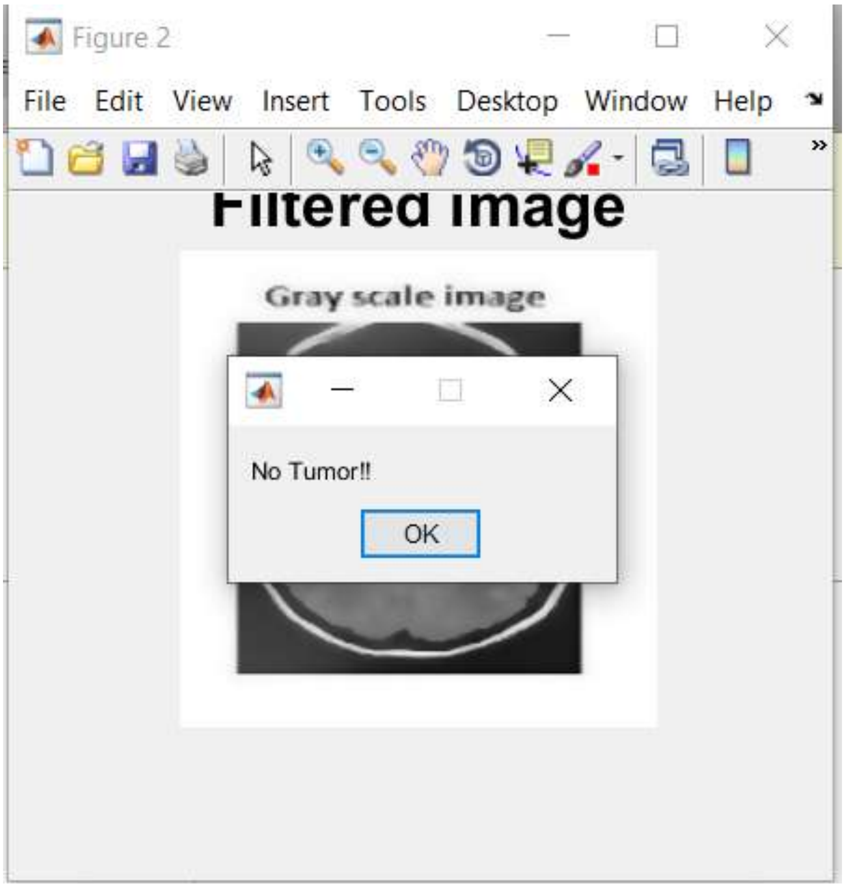
For a brain MRI which has brain tumour



For a brain MRI which has not brain tumour







No Tumor!!

CONCLUSION: -

- □ In this paper, a brain tumour MRI image is applied to pre-processing and after that tumour is extracted by anisotropic diffusion process. The medical image segmentation has difficulties in segmenting complex structure with uneven shape, size, and properties. In such condition it is better to use unsupervised methods such as anisotropic diffusion algorithm.
- □ For accurate diagnosis of tumor patients, appropriate segmentation method is required to be used for MRI images to carry out an improved diagnosis and treatment. The brain tumor detection is a great help for the physicians and a boon for the medical imaging and industries working on the production of MRI imaging.

FUTURE SCOPE

In near future, a database can be created for different patients having different types of brain tumours and locate them. Tumour growth can be analysed by plotting graph which can be obtained by studying sequential images of tumour affect. Possible extension of the presented work could use more features. It would be beneficial to connect the system to cloud storage of patient‟s information in hospital. This application can be extended to accessibility and usability through mobile phones. If this application is developed to analyse all types of MRI scans of same patient and result of all scans are integrated, it can suggest appropriate treatment and medication as well.