

# Brain Tumor Detection from MRI Images Using MATLAB Code

Divya Gavadi, Mahammadjaveed N, S Hemapriya

Department of Electronics & Communication Engineering

TONTADARYA COLLEGE OF ENGINEERING Mundaragi Road gadag

Corresponding Author: Ravi Gadad

\*\*\*

**Abstract** -Engineering have been actively developing tools to detect tumors and to process medical images. Medical image segmentation is a powerful tool that is often used to detect tumors. This project is about detecting brain tumors from MRI images using an interface of GUI in MATLAB. Using the GUI, this program can use various combinations of segmentation, filters, and other images processing algorithm to achieve the best results. Brain tumor segmentation from MR images of patients taken from 'Brain web'. Hybrid methodology is used to extract and detect tumor MR images using basic concept of digital image processing. MATLAB is used for this purpose in this paper for applying the proposed algorithm

**Key Words:** Matlab, MRI

## 1. INTRODUCTION (Size 11, Times New roman)

In these days, the interest in digital biomedical image processing methods takes a most important position in two principal and important areas[1]. The most important one is an improvement of pictorial information for human studies and processing of biomedical image data for storage. A biomedical image sometimes is defined as a two dimensional function,  $F(x,y)$ , where  $x$  and  $y$  are the value or Gray level of a biomedical image at a specific point.  $F$  are all finite, discrete quantities. We should know that to say an image is a digital image is when it is composed of a

finite number of elements, each of which has a particular location and value.

Tumor is defined as the abnormal growth of the tissues. Brain tumors are generally classed into two types benign and malignant tumors. MRI imaging plays an important role in brain tumor for analysis, diagnosis and treatment planning. It is helpful to doctor for determining the previous Steps Of brain tumor (II. The detection of brain tumor using MRI images are a challenging task because of the complex structure of the brain. MRI images provide better results than CT scan, Ultrasound and X-ray because it is an advanced medical imaging technique and it powerful magnets to produce high resonance images of all parts of the body, These MRI image can be processed and one can detect the brain tumor using image processing technique by forming automatic detection process using various algorithms because the manual detection of tumor from MRI images give human error. In this project our focus will be mainly to improve the existing approaches of image processing or design a better approach for the detection of the tumor

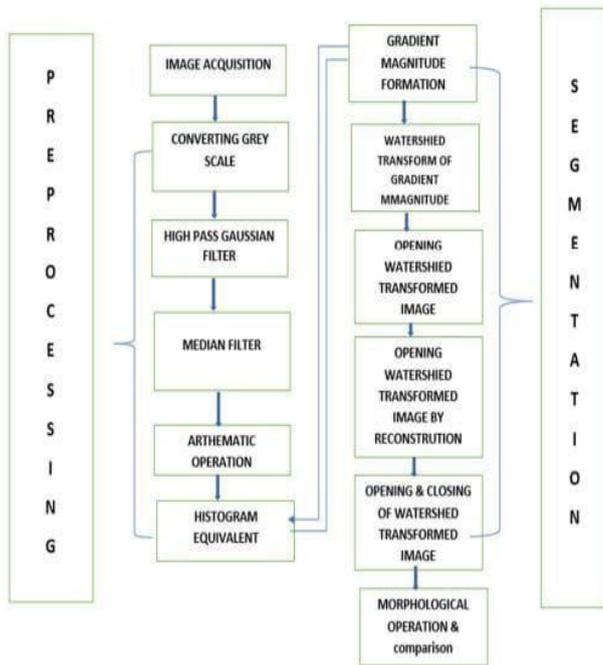
## 2. Requirement Analysis

Operating System : Window 10

Programming Language : C

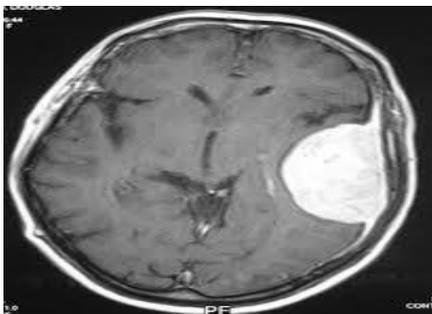
programmingSoftware : Matlab

### 3. Block Diagram



BLOCK DIAGRAM

### 4. Input Image



The first stage i.e. the image acquisition stage which starts with taking a collection of images forming the database. Images are stored in MATLAB will be displayed as a Gray scale image. As our MRI DICOM data set was in mat file, we have extracted the images using MATLAB code.

### 5. Image pre-processing

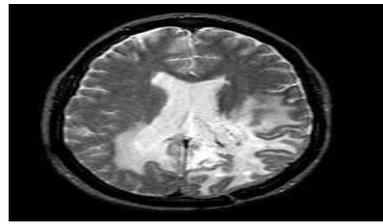
This is the second step of image processing where the images are used to enhance the chances of detecting the suspicious region. Images are enhanced and noises are removed from the images. The goal of pre-processing is to remove the noise and to provide Contrast Enhancement to improve the image quality. Pre-processing process is

- ✓ Gray scale conversation

- ✓ Noise Removal

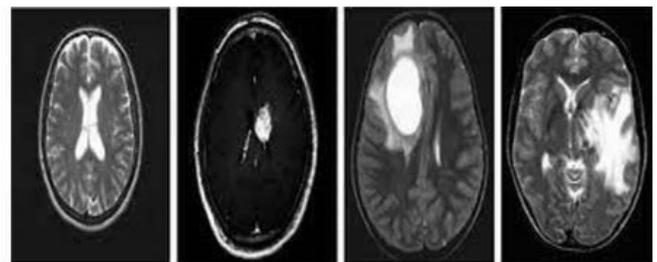
- ✓ Contrast Enhancement

### 3. Conversion to Gray scale



A grayscale image only consists of Gray scale values, but MRI images consist of primary color (RGB) content. When MRI images are viewed, they look like black and white but they contain some primary color (RGB). So, for further processing of MRI brain image, it must be converted to perfect Gray scale image

### 6. Filtering



Filtering is a technique for modifying or enhancing an image. Image processing operations implemented with filtering include smoothening, sharpening and edge enhancement. Filtering is a neighborhoods operation, in which the value of any given pixel in the output image is determined by applying some algorithms to the values of the pixels in the neighborhood of the corresponding input pixel.

### 7. Image Enhancement

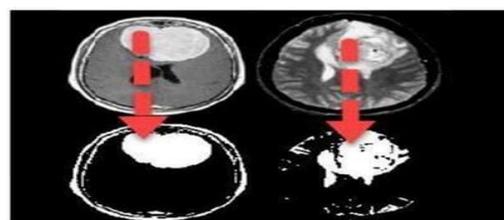
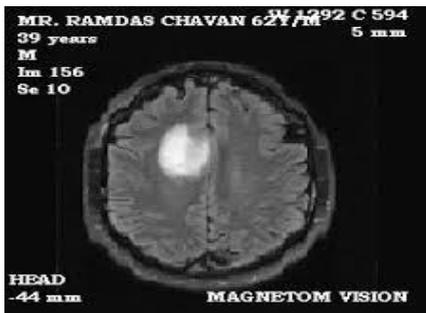


Image enhancement the details that are obscured and highlights certain features in an image. The fundamental enhancement needed in the MRI images is the contrast enhancement. Here we have Image enhancement provides the details that are obscured and highlight certain features in an image.

## 8. Image Segmentation

The next stage is image segmentation. Image segmentation is the process of partitioning a digital image into multiple segments. Image segmentation will be typically used to locate objects and boundaries in images and it helps to make the analysis easier. Here we have implemented Threshold method for image segmentation. For intensity based segmentation it is better to use threshold rather than colour based segmentation technique. In threshold based segmentation each an every pixel is compared with the threshold value.

## 9. Feature Extraction



The next stage is the feature extraction. The Image Feature Extraction stage is very important in our work where we will use algorithm and techniques to detect and isolate various desired portions or shapes of an image. After the segmentation is performed the feature; can be obtained from it and the diagnosis rule can be designed of exactly detect the tumor. Feature extraction is the transformation of input data into a set of features; it is a key stage of performing the task, which identifies set of features that describe the visual texture of an image

## 10. Algorithm used for Detection of Tumor

Step 1: Input the image

Step 2: Convert the RGB image to gray

Step 3: Use the filters for removing noise and perform subtraction operation and form histogram equivalent for Contrast enhancement.

Step 4: Apply Watershed Segmentation for detection

Step 5: Perform morphological operations for calculating Area, Third moment, Entropy, Mean, Standard Deviation.

Step 6: Tabulate and Compare the parameters calculated for normal and abnormal images.

## 11. Source code

```

%Input
% [I, path]=uigetfile('*.*jpg','select ainput image');
% str=strcat(path,I);
s=imread('C:\Users\HP\Documents\MATLAB\br ain.JPEG');
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 . . .');
%Preprocessing image please wait . . .inp =
anisodiff(s,num_iter,delta_t,kappa,option)
;
Removing noise Filtering Completed !! 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:size(inp,1)
for j=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
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 pixelsdilationAmount
= 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
eroded        Image(i,j)
min(min(filledImage(x1:x2,y1:y2))); end
end figure
imshow(erodedImage);
title('eroded image','FontSize',20);

```

**%subtracting eroded image from originalBW image**

```

tumorOutline=tumor;
tumorOutline(erodedImage)=0;    figure;
imshow(tumorOutline);
title('Tumor Outline','FontSize',20);

```

**%Inserting the outline in filtered imagein 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
Tumor','FontSize',20);

```

**%Display Togetherfigure**

```

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);

```

**%Filter defining**

```

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;
dd = sqrt(2);
% 2D convolution masks - finitedifferences.
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

```

• Instead of rectangular boxes, can work with general boundaries: level set based framework.

• By creating 3d anatomical models from individual patients, training, planning and computer guidance during surgery is improved

**REFERENCES**

1. Text Book : Study on brain tumor detection using MRimages – written by Abdullah Al Mahmud
2. Digital Image Processing by Kenneth R.Castleman
3. Digital Image Processing Using Matlab by Rafael C.Gonzalez,Ricard E.Woods,Steven L.Eddins

**CONCLUSIONS**

MRI images are best suitable for brain tumor detection. In this study Digital Image Processing Techniques are important for brain tumor detection MRI images. The pre-processing techniques include different methods like Filtering, Contrast enhancement, Edge detection is used for image smoothing. The pre-processed images are used for post processing operation like threshold and Chen features are extracted using GLCM and classification is done by Artificial Neural Network where we have obtained 76.5% of accuracy when run on a dataset of 300 images in classifying tumor and non tumor images. This work help in detection of tumor which in turn save the precious time of doctor and pathologist to diagnose the tumor automatically in short span of time.

**Future Scope**

- In future, this technique can be developed to classify the tumors based on feature extractions.
- This technique can be applied for ovarian, breast, lungs, skin, tumors.