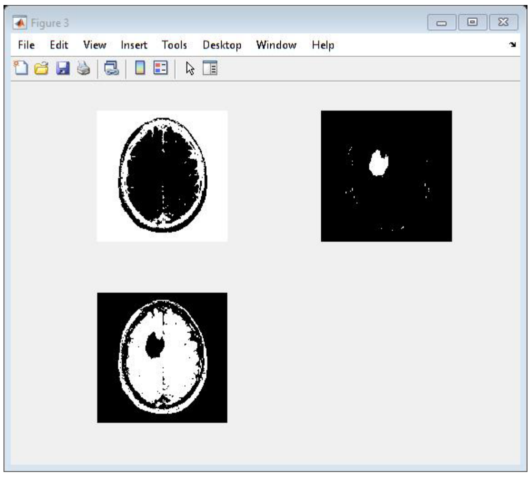
FAT – Digital Image processing

Aim
To use nprtool for Brain tumor segmentation

Algorithm
1. We begin by inputting the boiler plate code:
    a. Clearing the screen
    b. Clearing the workspace (Memory)
    c. Clearing the current figures (Closes all the figures)
2. We input the image (Computer vision toolbox is used)
3. We show the figure and set a title.
4. We then check if an image is in grayscale or not, this is to make sure that if an image is already in grayscale we do not apply any operation on it since it might cause loss.
5. We reshape the image by using the reshape command.
    a. Reshape(x, <variable-columns>, <variable-rows>)
    b. We make sure that the number of rows is 1
    c. This Results in an array of 65536x1 dimension
6. We then convert the array data to double format.
7. We then use kmeans clustering over the data, this returns (Statistics and machine learning toolbox is used)
    a. Cluster indices, returned as a numeric column vector. idx has as many rows as X, and each row indicates the cluster assignment of the corresponding observation.
    b. Centroid data
    c. Number of clusters is given as 3
8. We reshape the resultant image into the same dimensions as the input image.
9. We then proceed to plot all three images that we get from the clusters
    a. 

10. We now proceed to segment the tumor by selecting the cluster that is of interest.
11. imopen(I,SE) performs morphological opening on the grayscale or binary image I, returning the opened image, J. SE is a single structuring element object returned by the strel or offsetstrel functions. The morphological open operation is an erosion followed by a dilation, using the same structuring element for both operations.

12. BW2 = bwareaopen(BW,P) removes all connected components (objects) that have fewer than P pixels from the binary image BW, producing another binary image, BW2. This operation is known as an area opening.
13. We now display the segmented tumour.
14. We store the rows and columns of bw image to parameter signal 1
15. We now use DWT [Discrete wavelet transform] [wavelet toolkit] with db4 to Perform single-level decomposition
16. A 2D Discrete Wavelet transform is performed, this returns the:
    a. Approximation coefficients, returned as an array whose size depends on X. Let sx = size(X) and lf = the length of the decomposition filters.
    b. Horizontal detail coefficients, returned as an array whose size depends on X. Let sx = size(X) and lf = the length of the decomposition filters.
    c. Vertical detail coefficients, returned as an array whose size depends on X. Let sx = size(X) and lf = the length of the decomposition filters.
    d. Diagonal detail coefficients, returned as an array whose size depends on X. Let sx = size(X) and lf = the length of the decomposition filters.
17. We supply the Approximation coefficients to dwt again to get the second level 2DDWT
18. We supply the Approximation Coefficients to dwt again to get the third level 2DDWT
19. We now store the Approximation Coefficients, the Horizontal detail coefficients, Vertical detail coefficients and the diagonal detail coefficients as a feature array.
20. We now get the principal coefficients using pca from the statistics and machine learning toolbox
    a. We supply the Features we get from the **third level 2D-DWT**
21. We use the principal coefficients to get the gray level co-occurance matrix [GLCM]
22. We now use the GLCM to calculate the statistics specified in properties.
    a. We have to specify the statistics we want from the GLCM
23. We now use the statistics structure that we got from using graycoprops to get the contrast, correlation, Energy and homogeneity.
24. We now calculate the standard deviation, Entropy, RMS values, and variance. We add all the values of the array to calculate the smoothness.
25. We also calculate the Kurtosis and skewness of the image.
26. We calulate the IDM by: [Inverse difference moment]
    a. Running a loop and summation of all the image pixels and dividing by (1+[i-j]^2)

The formulae for all the features are as follows

$$Mean\ \boldsymbol{\mu} = \sum_{\mathrm{i}=0}^{\mathrm{N}-1} i, p(i)$$

$$\boldsymbol{Standard\ deviation\ \sigma} = \sqrt{\sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{N}-\mathbf{1}} (\boldsymbol{i}-\boldsymbol{\mu})^2 - \boldsymbol{p}(\boldsymbol{i})}$$

$$\boldsymbol{Entropy\ En} = -\sum_{\boldsymbol{i}=\mathbf{0}}^{\boldsymbol{N}-\mathbf{1}} \boldsymbol{p}(\boldsymbol{i})\boldsymbol{log}\mathbf{2}[\boldsymbol{p}(\boldsymbol{i})]$$

$$Kurtosis\ \mu^4 = \sigma^4 \sum_{i=0}^{N-1} \left((i-\mu)^4 - p(i) \right) - 3$$

$$Skewness\ \mu^3 = \sigma^3 \sum_{i=0}^{N-1} \left((i-\mu)^3 - p(i) \right)$$

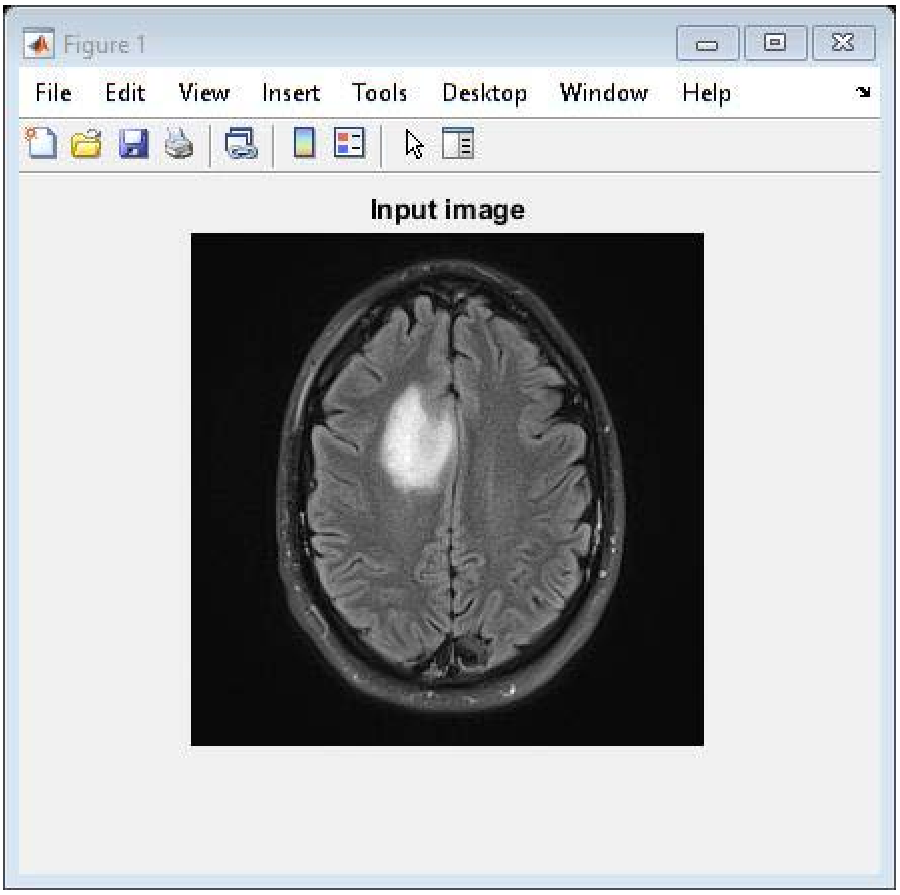
$$Inverse\ Difference\ Moment\ IDM = \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} \frac{1}{1+(i-j)^2} p(i,j)$$

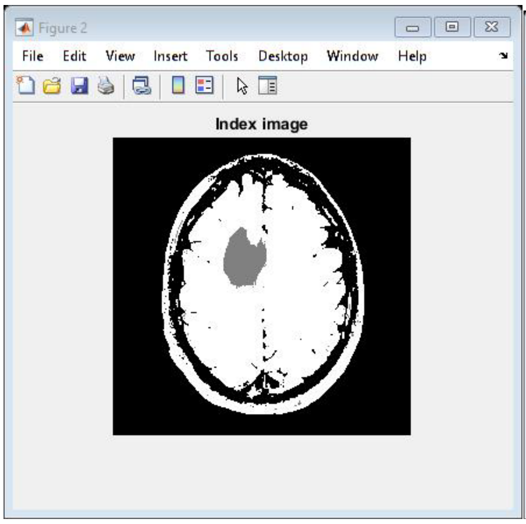
$$Correlation = \frac{1}{\sigma_a \sigma_b} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i,j)p(i,j)^2 - \mu_a \mu_b$$

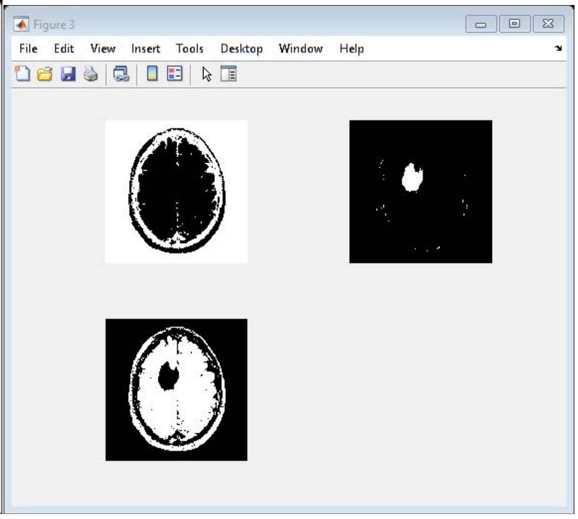
We now create a complete array of the features and display them.

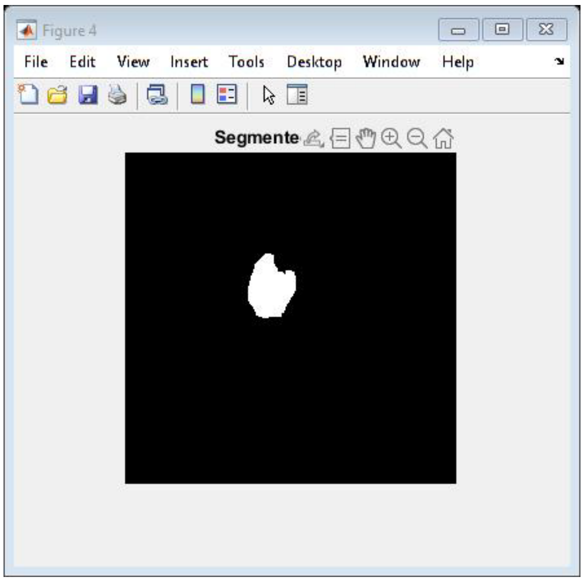
Output

```
The features are: [0.417763 0.120251 0.9044 0.969891
0.00480556 0.0809754 1.2235 0.0811107 0.00657544
0.964319 63.5488 5.21125 11.5726]
```









Proceed to next page to see matlab code

```
clc;%clear command window
clear all;%clear workspace
close all;%close all current figures
a = imread('brain3.jpeg');%reading the image
figure, imshow(a); %new figure %displaying the image
title('Input image'); %title of image

%loop to check if image is gray scale,
%if not then converting from RGB to GRAY
try
    Dimg = rgb2gray(a);%conversion RGB to GRAY
catch
    Dimg = a;% no need to convert it is already GRAY image
end
imdata = reshape(Dimg,[],1);%reshape the image into 65536x1 dimension
imdata = double(imdata);%Handling large data (Convert to Double)

%Clustering the image
[IDX,nn] = kmeans(imdata,3);%Kmeans inbuilt command to cluster the
                            %input image into 3 different clusters
imIDX = reshape(IDX, size(Dimg)); %reshaping the resultant image into
                                  %the dimension of input image
figure, imshow(imIDX,[]); %showing index image
title('Index image');
figure,
subplot(2,2,1),imshow(imIDX==1,[]); %showing the 1st cluster
subplot(2,2,2),imshow(imIDX==2,[]);%showing the 2nd cluster
subplot(2,2,3),imshow(imIDX==3,[]);%showing the 3rd cluster

%Segmenting the tumor
bw = (imIDX==2); %selecting the cluster in which tumor is present
se = ones(5); %creating a structuring element of all ones(5x5)
bw = imopen(bw,se); %The morphological open operation is an erosion
                    %followed by a dilation, using the same
                    %structuring element for both operations.
                    %Basically used to remove noise and, also for
                    %sharpening and smoothening of the image

bw = bwareaopen(bw,1200); %removes all connected components (objects)
                          %that have fewer than 1200 pixels from the
                          %binary image bw.

figure,imshow(bw);   %Display cleaned image
title('Segmented tumor');

%Feature extraction
signal1 = bw(:,:); %Storing all rows and all columns of bw image to
                   %parameter signal1


%Feature extraction using DWT
```

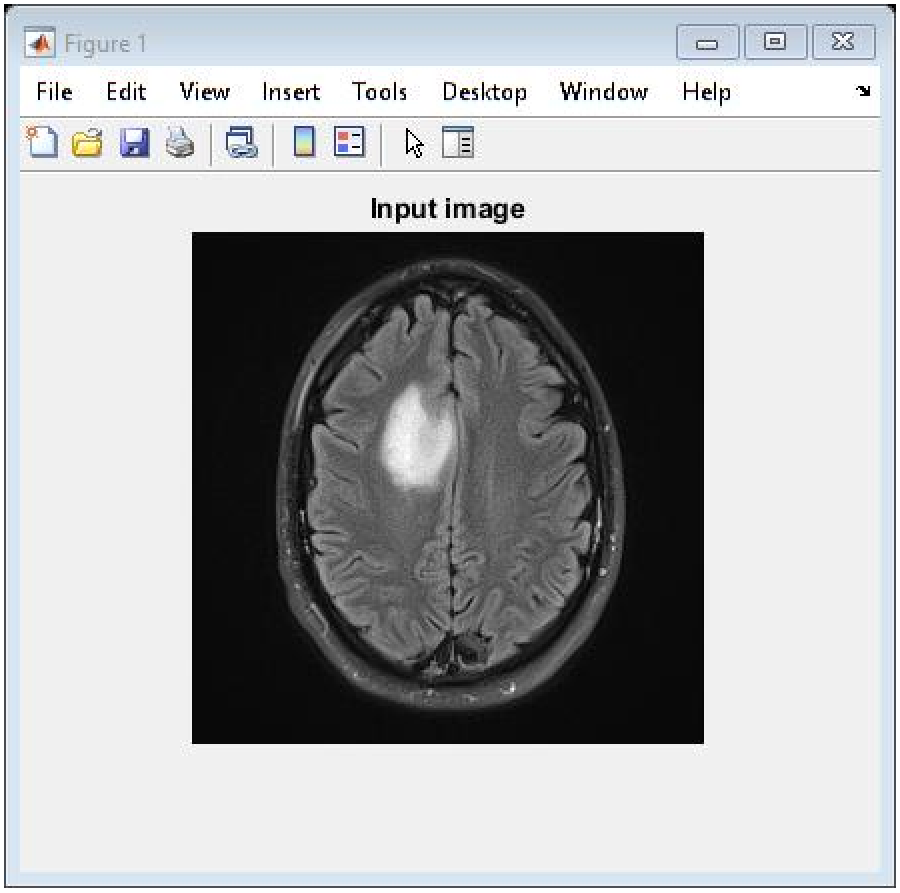
```
%In numerical analysis and functional analysis, a discrete wavelet
 transform (DWT) is any wavelet transform for which the wavelets are
 discretely sampled.
%As with other wavelet transforms, a key advantage it has over Fourier
 transforms is temporal resolution: it captures both frequency and
 location information (location in time).
%db4 or Daubechies wavelets is based on the use of recurrence
 relations to generate progressively finer discrete samplings of an
 implicit mother wavelet function; each resolution is twice that of
 the previous scale.

[cA1,cH1,cV1,cD1] = dwt2(signal1,'db4'); %Computes the single-level 2-
D discrete wavelet transform (DWT) of the input data signal1 using the
 db4 wavelet. dwt2 returns the approximation coefficients matrix cA
 and detail coefficients matrices cH, cV and cD (horizontal, vertical,
 and diagonal, resp.)
[cA2,cH2,cV2,cD2] = dwt2(cA1,'db4'); %Second level 2D DWT
[cA3,cH3,cV3,cD3] = dwt2(cA2,'db4'); %Third level 2D DWT
DWT_feat=[cA3,cH3,cV3,cD3]; %Storing all the coefficients in 1
                            %variable
G = pca(DWT_feat); %Returns the principal coefficients
g = graycomatrix(G); %Creates the gray-level co-occurrence matrix
 (GLCM) by calculating how often a pixel with gray-level (grayscale
 intensity) value i occurs horizontally adjacent to a pixel with the
 value j.
stats = graycoprops(g,'Contrast Correlation Energy Homogeneity');
                %Calculates the statistics specified in properties
                %from the gray-level co-occurrence matrix glcm
Contrast = stats.Contrast; %Returns cached value of Contrast
Correlation = stats.Correlation;
Energy = stats.Energy;
Homogeneity = stats.Homogeneity;
Mean = mean2(G); %Computes the standard deviation of all values in
                   %array G.
Standard_Deviation = std2(G); %Computes the standard deviation of
                                %all values in array G.
Entropy = entropy(G); %Computes the entropy of all values in array G.
RMS = mean2(rms(G)); %Computes the RMS value of all values in array G.
Variance = mean2(var(double(G))); %Computes the variance of all
                                    %values in array G
b = sum(double(G(:))); %Adding all the values of matrix G
Smoothness = 1-(1/(1+b)); %Calculating Smoothness of G
Kurtosis = kurtosis(double(G(:))); %Calculating Kurtosis of G
Skewness = skewness(double(G(:))); %Calculating Skewness of G
m = size(G,1);
n = size(G,2);
in_diff = 0;
for i = 1:m
    for j = 1:n
        temp = G(i,j)./(1+(i-j).^2);
        in_diff = in_diff+temp;
    end
end
IDM = double(in_diff); %Calculating IDM of G
```

```
feat = [Contrast, Correlation, Energy, Homogeneity, Mean,
 Standard_Deviation, Entropy, RMS, Variance, Smoothness, Kurtosis,
 Skewness, IDM];
%Displaying all the features in 1 variable
disp(['The features are: [' num2str(feat(:).') ']']) ;


% Conclusion:
% In this experiment we have taken 15 samples 10 are Malignant(HGG)
 and S are Benign(LGG)
% Kmean clustering technique is applied to detect the tumour,
 alongside 13 features were detected of each sample.
% The database was established for 15 samples, named as TRAIN_ANN.mat
 and target was created using database
% Using nprtool the experiment was performed and the accuracy was
 calculated through confusion matrix((TP+TN)/TR+FN+FR+TF)
% It is observed for 15 samples we achieved 60% of accuracy, by
 increasing the number of training set we aim to achieve more
 accuracy.
```

```
The features are: [0.417763     0.120251       0.9044      0.969891
  0.00480556     0.0809754       1.2235     0.0811107    0.00657544
 0.964319       63.5488       5.21125       11.5726]
```



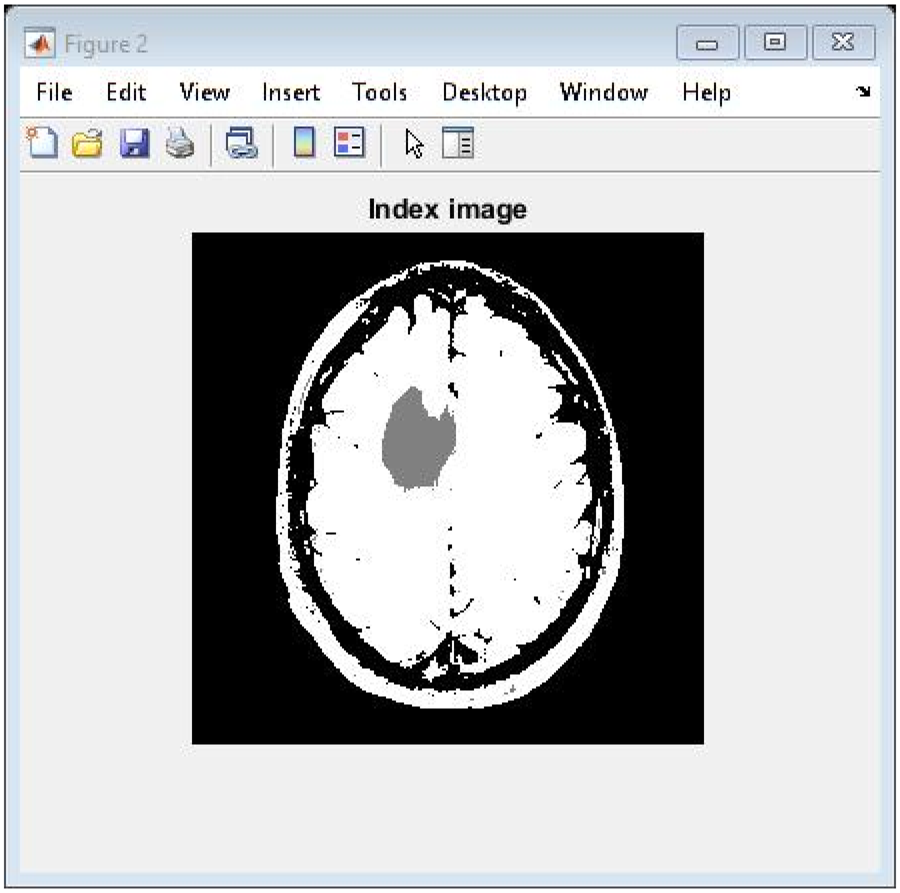


Figure 2
File
Edit
View
Insert
Tools
Desktop
Window
Help
Index image

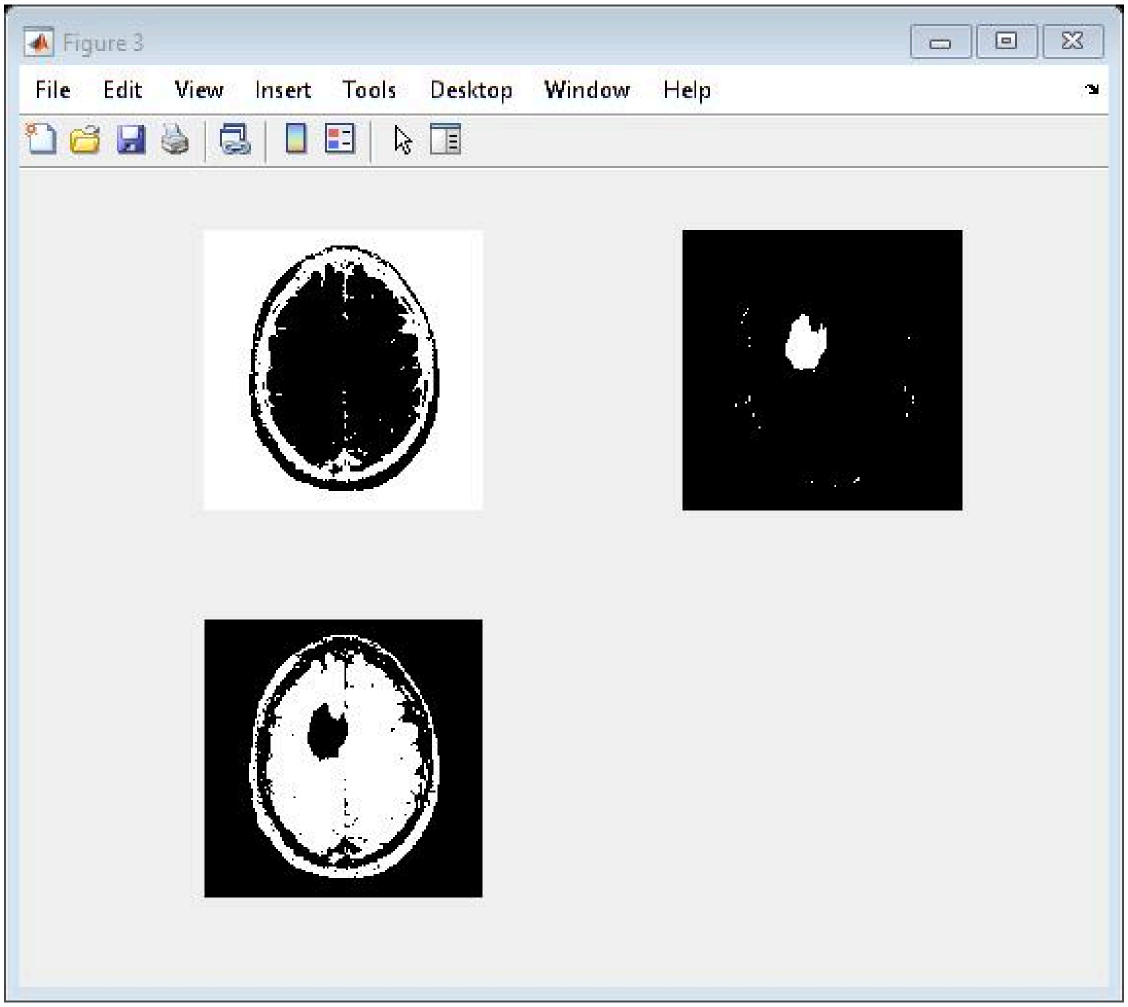
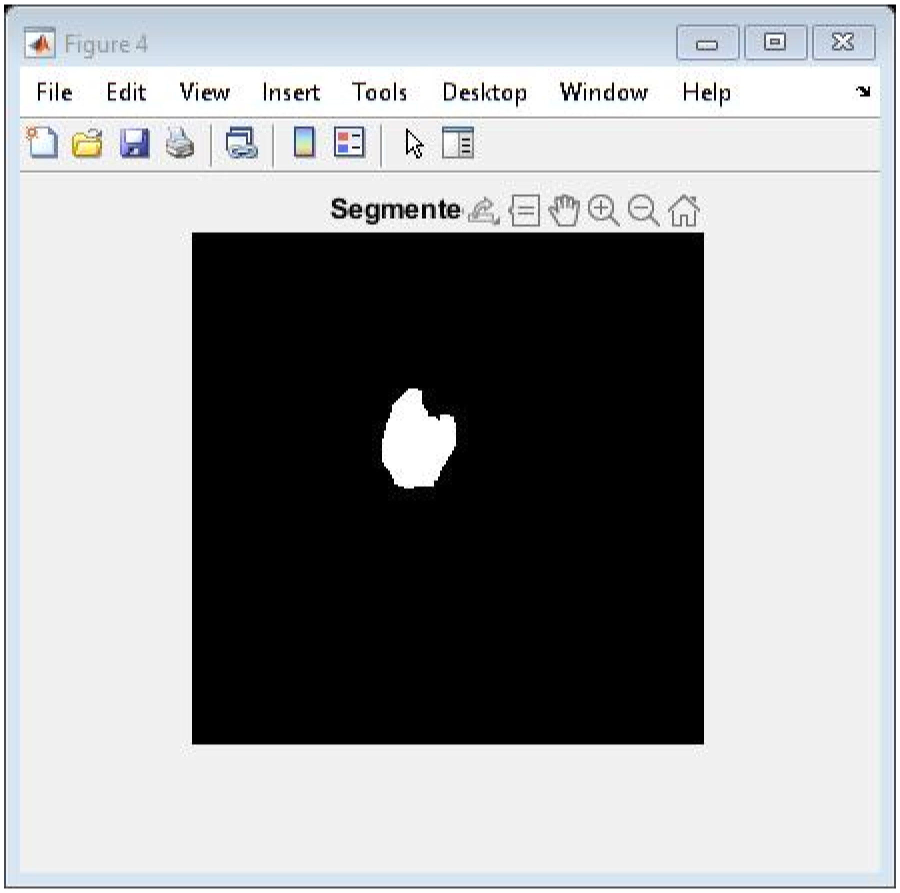


Figure 3
File Edit View Insert Tools Desktop Window Help



Published with MATLAB® R2020a