

---

```

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(:).') ']]);

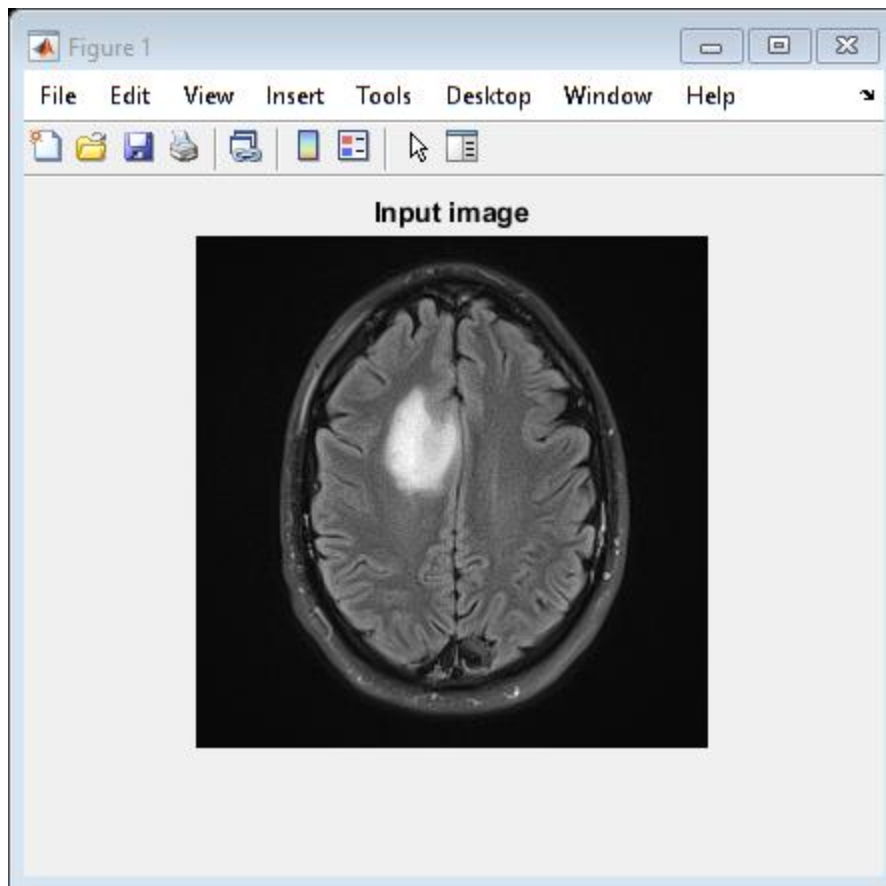
% Conclusion:
% In this experiment we have taken 15 samples 10 are Malignant(HGG)
  and S are Benign(LGG)
% Kmean clustering technique is applied to detect the tumour,
  alongside 13 features were detected of each sample.
% The database was established for 15 samples, named as TRAIN_ANN.mat
  and target 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.

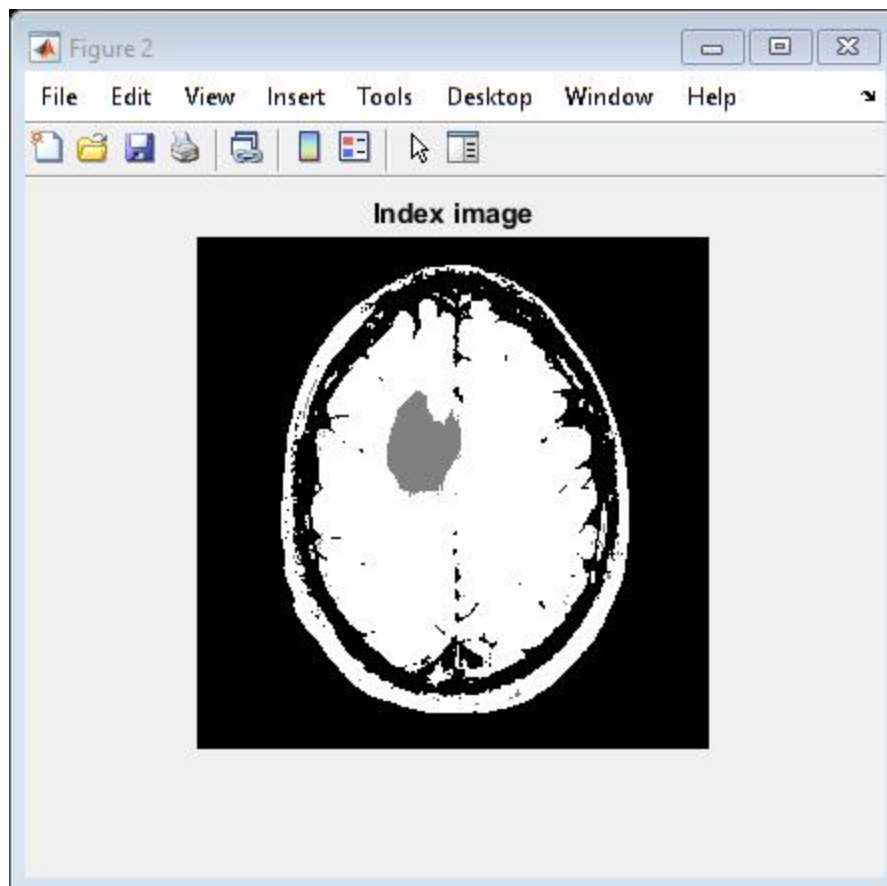
```

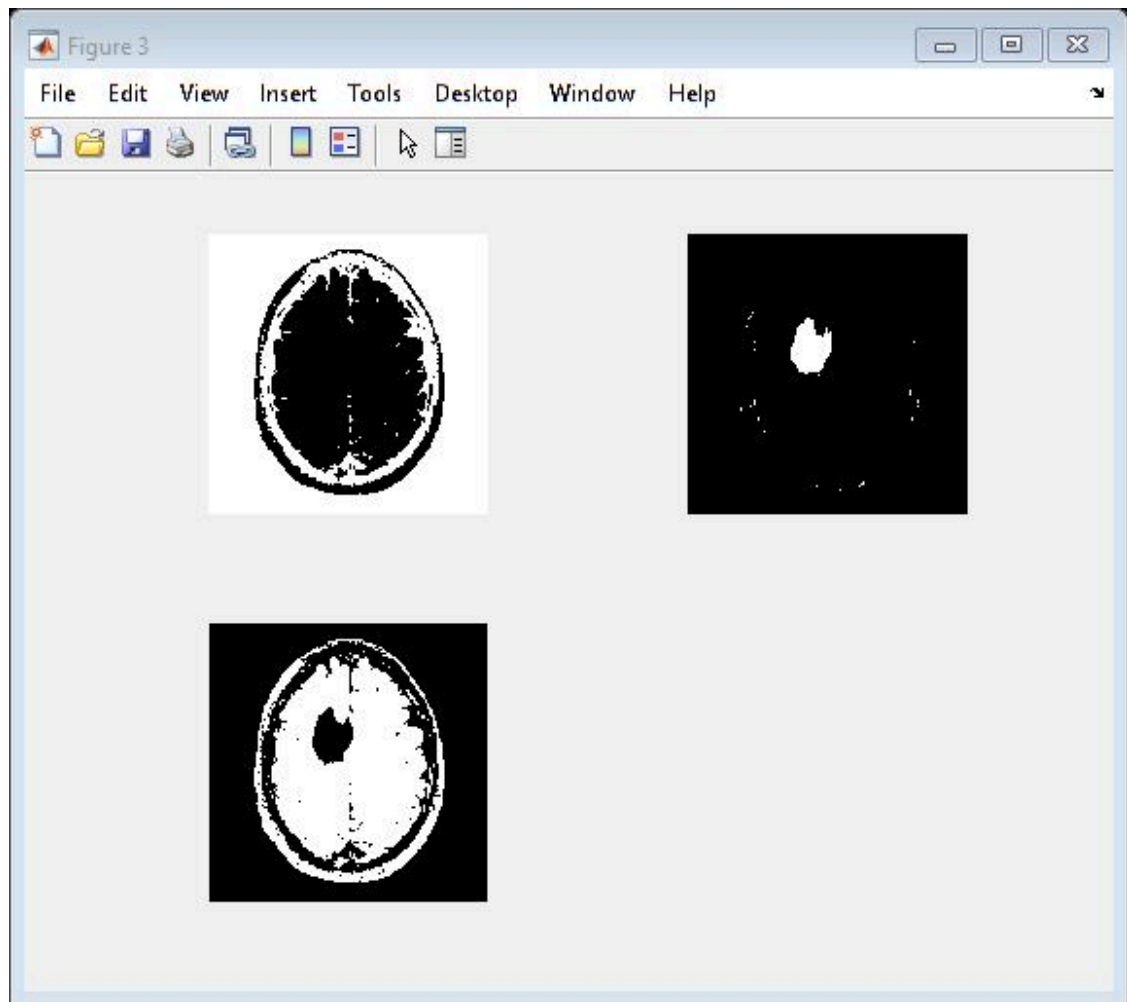
```

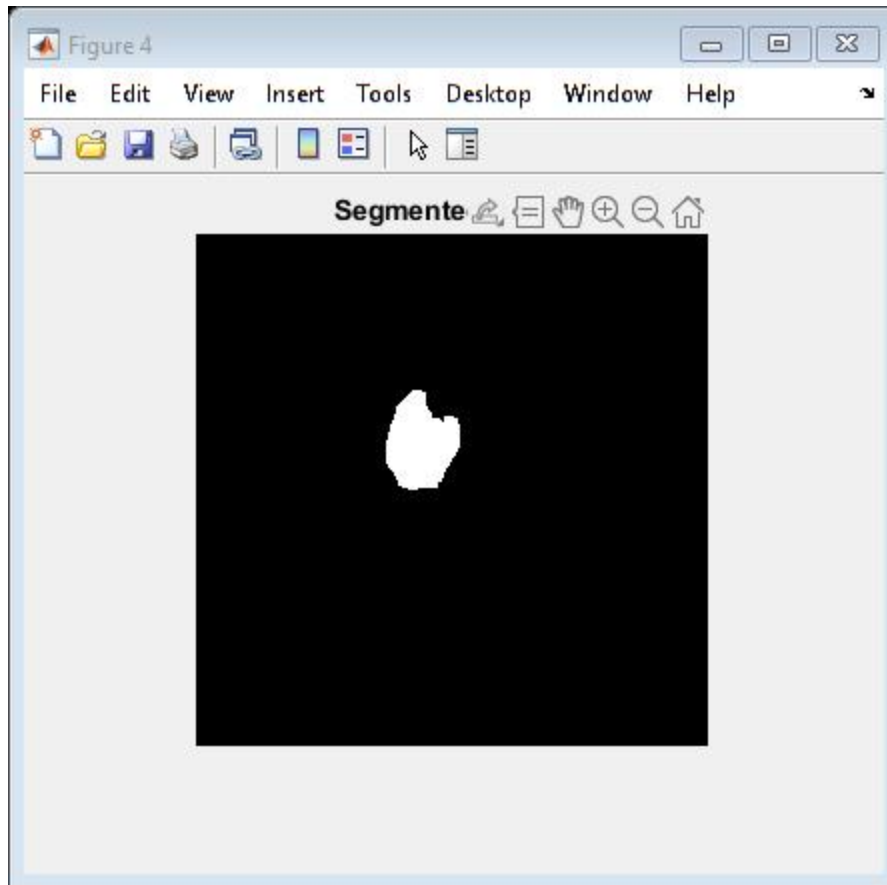
The features are: [0.417763      0.120251      0.9044      0.969891
                  0.00480556   0.0809754      1.2235      0.0811107   0.00657544
                  0.964319      63.5488      5.21125      11.5726]

```









*Published with MATLAB® R2020a*