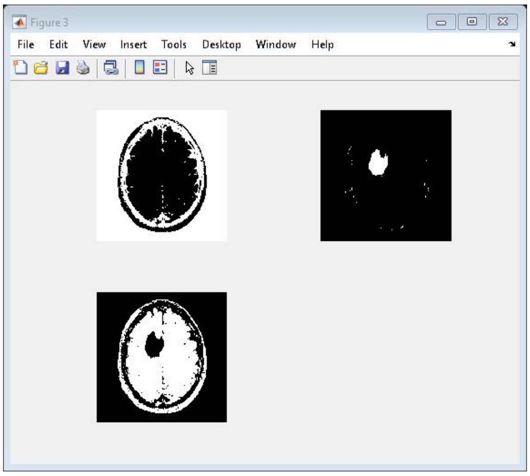
FAT – Digital Image processing

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

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

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

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

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

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

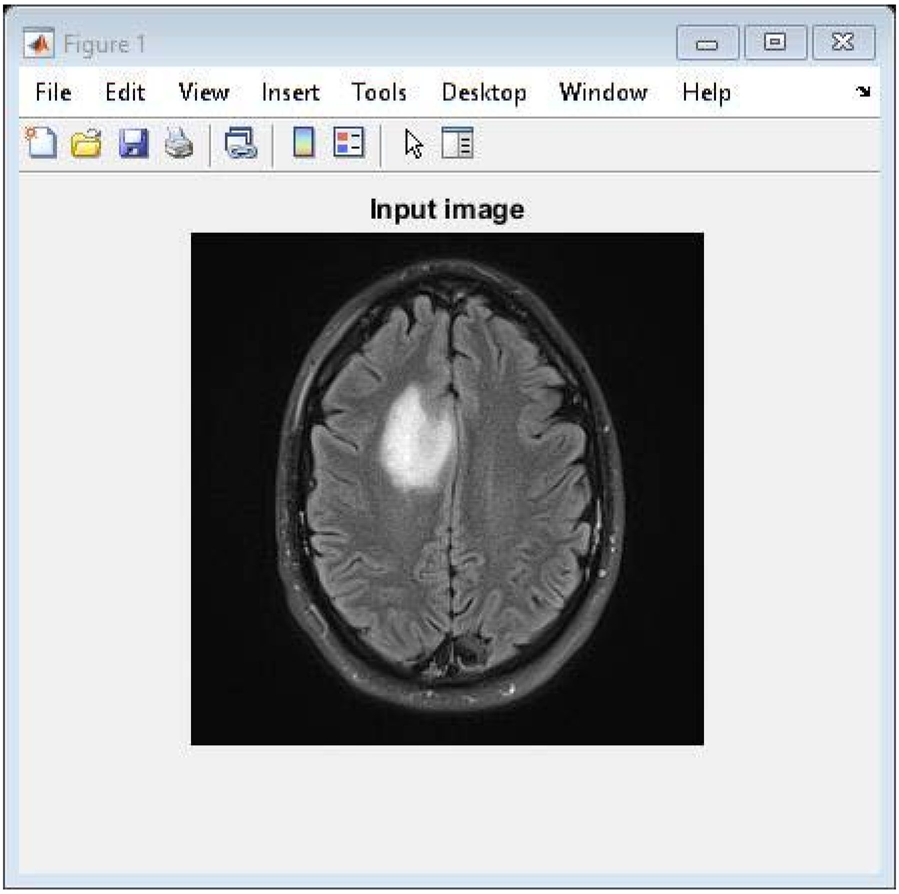
$$Inverse\ Difference\ Moment\ IDM = \sum_{i=0}^{N-1} \sum_{i=0}^{N-1} \frac{1}{1+(i-j)^2} p(i,j)$$

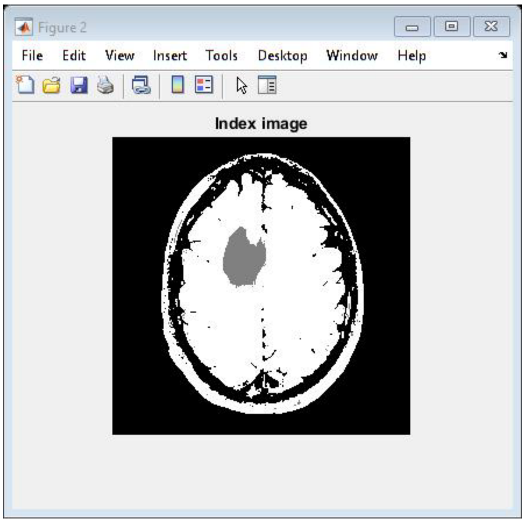
$$Correlation = \frac{1}{\sigma_a \sigma_b} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (i,j) p(i,j)^2 - \mu_a \mu_b$$

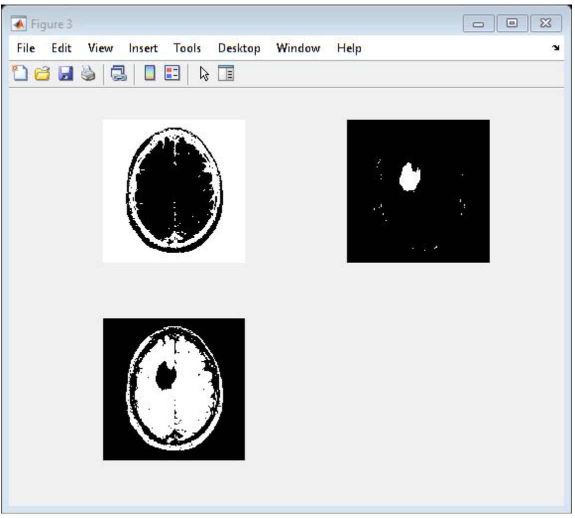
We now create a complete array of the features and display them.

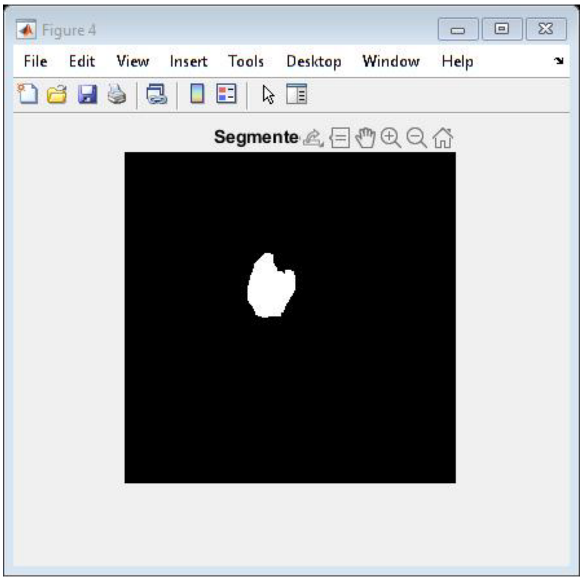
Output

```
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