**Design and methodology**

**K-Means**

Algorithm:

Let  X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.

Step 1: Randomly select *‘c’* cluster centers.

Step 2: Calculate the distance between each data point and cluster centers.

*‘||xi- vj||’* is the Euclidean distance between *xi* and *vj*

*‘ci’* is the number of data points in *ith* cluster.

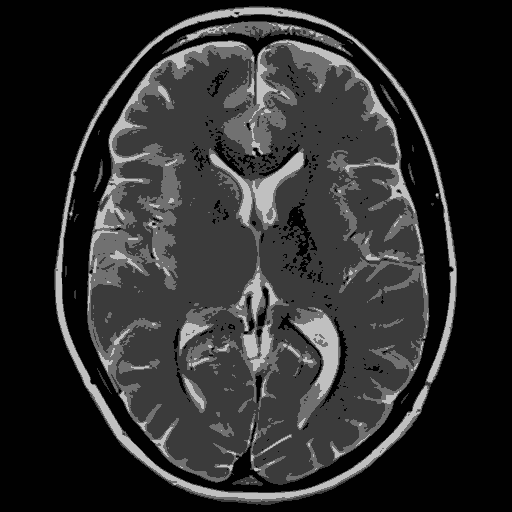
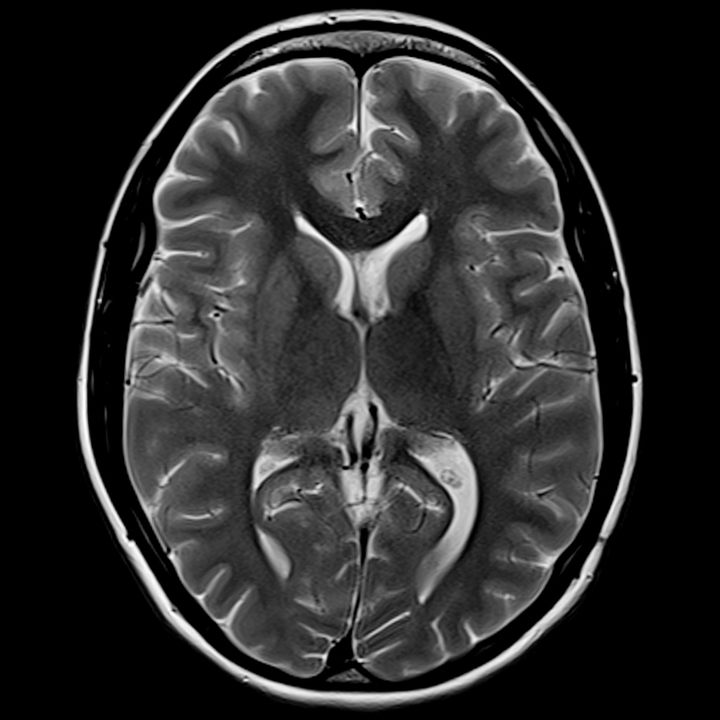
*‘c’* is the number of cluster centers.

Step 3: Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers.

Step 4: Recalculate the new cluster center using:

, where,*‘ci’* represents the no. of data points in *ith* cluster.

Step 5: Recalculate the distance between each data point and new obtained cluster centers.



**Fuzzy C-Means**

In Fuzzy C-Means, data has to be processed by giving the partial membership value to each pixel in the image. The membership value of the fuzzy set ranges from 0 to 1. Generally, it is hard to determine whether a pixel belongs to a region or not. This is due to unsharp transitions at region boundaries. Fuzzy partition is carried out by an iterative optimization of object function, with the update of the membership function and cluster centre. Nearer the data point to the cluster centre the more possible its membership towards the particular centre.

Algorithm:

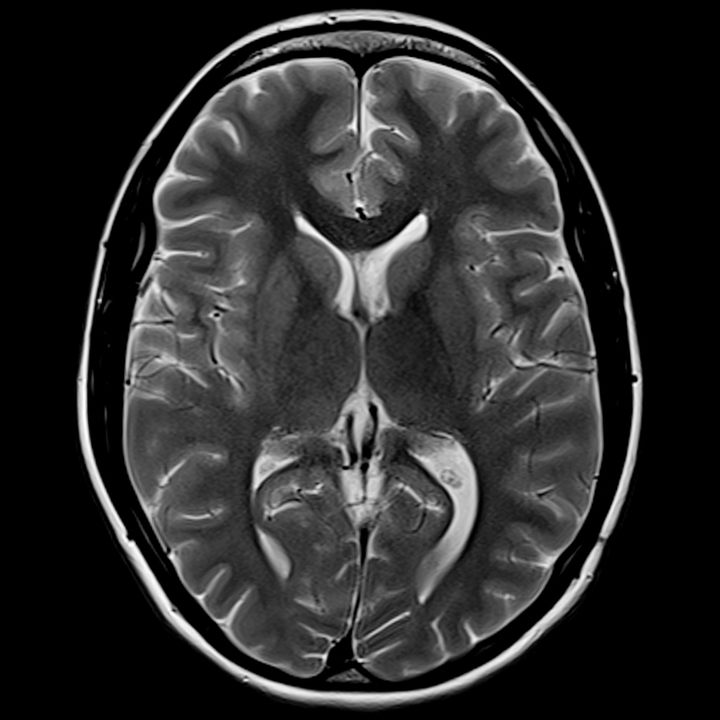
Step 1:U=[uij] matrix, U(0)

Step 2:At k-step: calculate the centers vectors C(k)=[cj] with U(k)

, where

Step 3: Update U(k) , U(k+1)

Step 4: If || U(k+1) - U(k)||<https://home.deib.polimi.it/matteucc/Clustering/tutorial_html/images/image002.gif then STOP; otherwise return to step 2.

**Spatial Fuzzy C-Means**

Due to the effect of noise in brain MR images, it is difficult for the traditional fuzzyc-means (FCM) clustering algorithm to obtain desirable segmentation results. Combining the information of patch to reduce the effect of noise has been a focus of current research. Thus, a model that takes both the non-local information and spatial structural similarity measurement (SSIM) between image patches has proved to give better results. A new distance function is established between every pixels and category centres for image segmentation.

Algorithm:

Step 1: Set the number of clusters.

Step 2: Initialize cluster centers,

Step 3: Initialize centroids, wi

Step 4: Set e = 0.001

Step 5: Set patch size and search window size. (3x3)

Step 6: Calculate w(xj, xk), using:

w(xj, xk) = (1 – λk) wssim(xj, xk) + λkwnl(xj, xk)

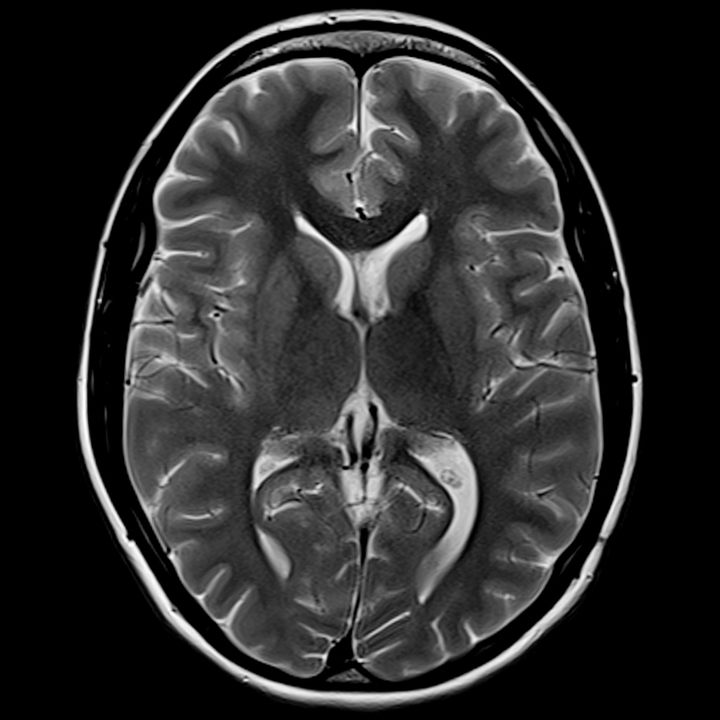
where λk is a trade-off parameter = max(wnl(xj, xk))

Step 7: Update the distance factor, D2(xj, vi), as:

D2(xj, vi) = w(xj, xk) D2(xj, vi)

Step 8: Update membership function, for each pixel I, belonging to cluster k, as:

, where

**Kernelized Fuzzy C-Means**

Algorithm:

Step 1: Fix c, tmax, m > 1 and e > 0 for some positive constant.

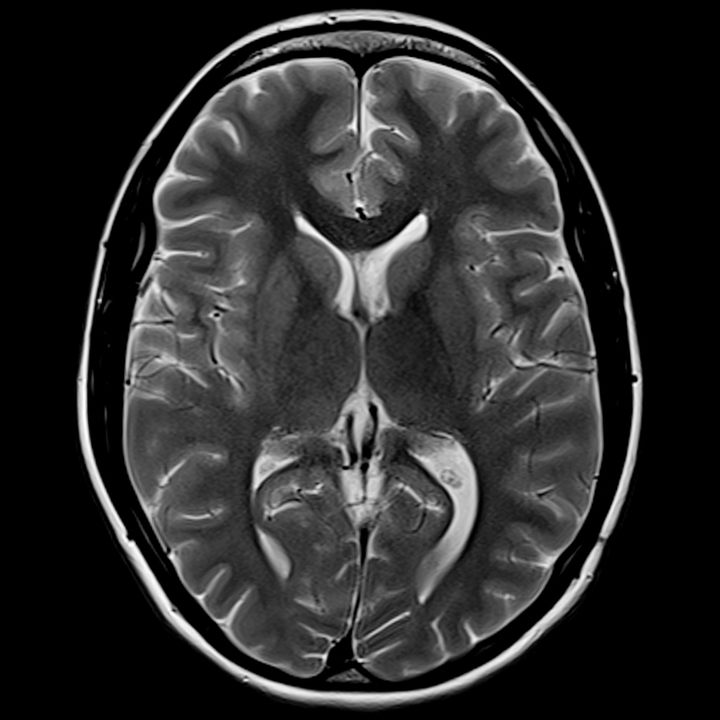
Step 2: Initialize the membership matrix : U=[uij] matrix, U(0)

Step 3: For t = 1, 2, … tmax, do:

(a) update all prototypes, vitusing

(b) update all memberships U(k) , U(k+1)

(c) Compute Et = maxi,k , if Et ≤ є

**Two-Stage Multi-threshold Otsu’s method (TSMO)**

TSMO a hybrid algorithm based on a self-adaptive thresholding method to optimize the threshold of the Otsu’s method. It greatly reduces the iterations required for calculating the zeroth- and first-order moments of a class. In the first stage of the TSMO method, the histogram of an image with L (=256) gray levels is divided into Mz groups which contain Nz (=256/Mz) gray levels with a certain range. Let X denote the groups of the total image space; then, X = {XJ /j= 0,1,. . . ,Mz -1}, where j represents the group number. The number of cumulative pixels and the mean intensity for each group can be easily determined. Since each group contains Nz gray levels, the corresponding gray level value for each group can be considered as a mean value for those Nz gray levels. Therefore, the set iX = {iX0 , iX1 , . . . ,iXMz\_1} of the corresponding gray level values for all groups in an image can be easily determined. Subsequently, Otsu’s method can be applied to find the optimal threshold by maximizing the between-class variance.

1. I = Input Image.

2. Obtain the histogram values (h) of the image I.

3. Set the initial Threshold value:

4. Segment using Tin. This will produce two groups of pixels: C1 and C2.

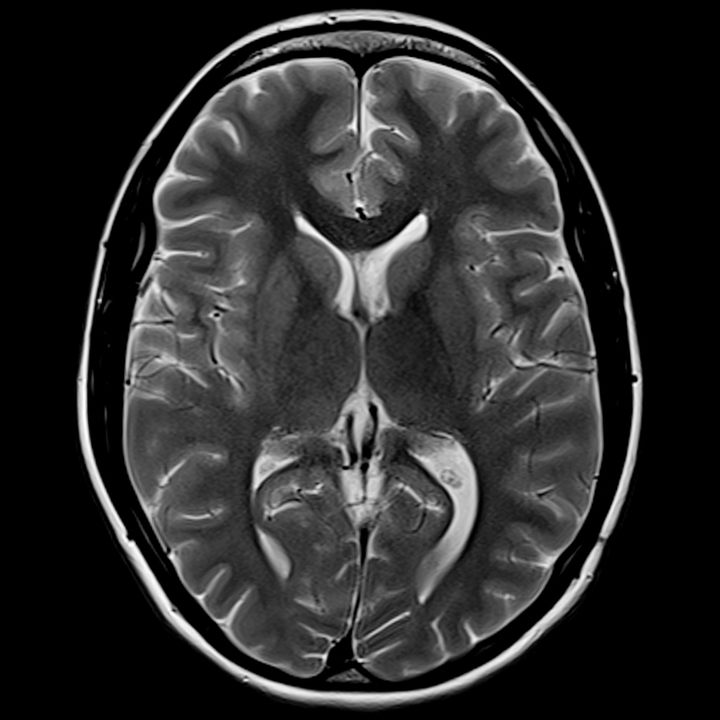
5. Repeat step-3 to obtain the new threshold values for each class. (TC1 &TC2)

6. Compute the new threshold value:

7. Repeat the steps 3-6 until the difference in T in successive iterations is not tends to zero

8. Now apply the Conventional Otsu method for the obtained threshold value for further segmentation process.

Finally the time (t) and η is calculated and compared with the original Otsu method.

**Support Vector Machine**

1. Creating Training dataset

i. Number of images = fCount ;

featuresmat is a matrix that stores the extracted features of each image ;

type is an array to store the type of image (Tumour or non-tumour)

ii. CheckImage(fIndex) function returns the 4 features of the image at the fIndex position:

Step 1: Read image

Step 2: Apply median filter on image I as a preprocessing step

Step 3: Perform Fuzzy C-means segmentation on the median filtered image

Step 4: The features of the segmented image are extracted using featurext(image) and stored in e1.

The 4 properties (energy, contrast, homogeneity, correlation) of each image extracted using GLCM (graycomatrix(image)). This value is returned as e1.

Step 5: The 4 properties are then returned as an array.

iii. The properties are copied to featuresmat matrix.

iv. The matrix is normalized

v. The features & known class of images are saved into the training dataset in a .mat file

2. Creating Testing dataset

Same process as Training dataset. Type is not provided as the SVM model will predict it.

3. i. Train SVM model based on Training dataset

SVMModel = fitcsvm(featuresmat,C);

This function is used after loading the training dataset. It trains the model.

ii. Predict Test data using SVM model –

After the model is trained, the testing dataset is loaded. The predict() function returns an array with the detected class of the images in the test data.

label = predict(SVMModel,featuresmat);