Brain Tissue Segmentation Using Expectation Maximization Algorithm Lab 2 Report

Gonzalo Esteban Mosquera Rojas, Kazi Saeed Alam
gonzalo.mosquerarojas2000@gmail.com, saeed.utsha@gmail.com
Erasmus Mundus Joint Master's Degree in Medical Imaging and Applications
University of Girona

I. Introduction and Problem Definition

Magnetic Resonance Imaging (MRI hereinafter) is a medical imaging technique used for the analysis of organs and anatomic structures of the human body (tissues, bones, etc). It is widely used for the examination of the brain, e.g, spot the presence of tumors, aneurysms, inflammation, etc. Brain Tissue Segmentation is one of the most significant medical imaging research fields. With the help of MRI image modalities, it is possible to segment the brain tissues into different anatomical regions of interest for surgical planning, monitoring therapy, clinical drug trials, image registration, stereotactic neurosurgery, radiotherapy, etc. The main purpose of this lab was to segment MRI brain images into three different classes (tissues), namely, cerebrospinal fluid (CSF), gray matter, and white matter. This was done by using the Expectation-Maximization algorithm (EM from now on). EM is a cluster-based learning algorithm that discovers latent variables from observed data. Image segmentation is an image processing procedure to label pixels of similar kinds into the same cluster groups. Therefore, EM can be used for the segmentation of brain MRI images. During the lab, the algorithm was implemented from scratch. In order to achieve this, a rigorous analysis of all the stages and components involved in the algorithm was performed. Afterward, its performance was evaluated using the brain images from five different patients and two imaging modalities (T1 and T2Flair). The comparison was done taking the K-means algorithm as a benchmark (K-means clustering Vs Randomly Initialized EM vs K-means initialized EM).

II. Algorithm Analysis

A. Expectation-Maximization algorithm

The Expectation-Maximization algorithm is an optimization technique used to estimate some unknown parameters of a problem given some measurement data [1]. The algorithm assumes that the data available come from K's different Gaussian distributions. If it was known which data points were generated from which distribution, the parameters of these distributions could be easily computed. On the other way, if the K different Gaussian distribution parameters were known, it would be possible to find which data points might have been generated by each of the Gaussian sources. However, in the real world, none of these two aspects are known. Therefore, the EM algorithm emerges as a way of tackling this problem. Its general idea is to initialize the parameters of the K distributions in a certain fashion (randomly, using k-means algorithm, etc), where K is equal to the number of clusters. Given the initial parameters of the Gaussian sources, the algorithm computes the data point labels in a soft way, which means that each point is given a certain probability of having been generated by each of the sources. In the following step, the algorithm iteratively updates the source parameters so that they can fit the data better until a certain convergence criterion is met [2]. In this case, stopping criteria were used: number of iterations and absolute difference of log likelihood between iterations.

1) Initialization of parameters: During each stage, given a dataset $D = [x_1, x_2, ..., x_N]$ where each x_i corresponds to a vector of measured data. Each vector x_i is modeled as a finite mixture model, defined as follows [2]:

$$p(x|\Theta) = \sum_{1}^{K} \alpha_k p(x|z_k, \theta_k) \tag{1}$$

Where $p(x|z_k, \theta_k)$ represents the mixture components, also known as density or probability distribution with parameters θ_k. On the other hand, z corresponds to a random variable that represents the identity of the source (mixture component) that generated a given measurement vector x. α_k are known as mixture weights, and those represent the probability that a randomly selected measurement vector x was generated from a Gaussian source k. This means that the summation for all weights for all values of K must be equal to 1. Each of the mixture components is modeled as a multivariate Gaussian density defined as follows:

$$p_k(x|\theta_k) = \frac{1}{(2 * \pi)|\sum_k|^{1/2}} e^{\frac{1}{2}(x-\mu_k)^t \sum_k^{-1}(x-\mu_k)} \tag{2}$$

where $\theta_k = \{\mu_k, \sum_k\}$ corresponds to the density parameters (mean and covariance of the Gaussian, respectively).

Once the theoretical basis is defined, the initialization of parameters in the EM algorithm consists on selecting the number of clusters according to the problem and also defining the values for the mean and covariance of the mixture components. For this task, several approaches can be adopted. In this laboratory two different methods are compared (random and k-means initialization) in terms of three metrics: dice score, computation time and number of iterations. Once the parameters are initialized, these are used to compute the gaussian mixture density model for each cluster (tissue), and the initial value of the loglikelihood is computed using the following formula [2]:

$$logl(\Theta) = \sum_{i=1}^{N} logp(x_i|\Theta) = \sum_{i=1}^{N}(log(\sum_{k=1}^{K} \alpha_k p_k(x_i|z_k, \theta_k)) \tag{3}$$

where $p_k(x_i|z_k, \theta_k))$ corresponds to the Gaussian density for the k-th gaussian component.

2) Expectation step: Once the parameters of the algorithm are initialized and a first estimation of the Gaussian mixture model is computed, the algorithm starts iterating and enters in the expectation step, which consists of computing the membership weights w_{ik} for all the data points and all the mixture components. These weights must be defined in a way such that $\sum k = 1^K w_{ik} = 1$. Therefore, the resulting weight matrix will be of size NxK, where N corresponds to the amount of data points usead as input to the algorithm, and K corresponds to the number of clusters the algorithm is trying to group. The weights are computed using the following formula [2]:

$$w_{ik} = p(z_{ik} = 1|x_i, \Theta) = \frac{p_k(x_i|z_k, \theta_k)\alpha_k}{\sum_{m=1}^{K} p_m(x_i|z_m, \theta_m)\alpha_m} \tag{4}$$

With $1 \leq k \leq K, 1 \leq i \leq N$.

This equation allows to see that weights computation for a data point x_i is not more than taking the gaussian density of that point multiplied by its mixture weight over the sum of the multiplication of the gaussian density and the mixture weight of each of the other points.

3) Maximization step : For the current number of iteration i, once the membership weights have been updated, the maximization step is performed. It consists of updating the parameters mixture weights, meand and covariance of each of the gaussian sources of the problem, and then re-estimating the probability density functions of the components with these new parameters. The formulas for computing the new mixture weights, mean and covariance are, respectively [2]:

$$\alpha_k^{new} = \frac{N_k}{N}, 1 \leq k \leq K \tag{5}$$

$$\mu_k^{new} = (\frac{1}{N_k}) \sum_{i=1}^{N} w_{ik} x_i, 1 \leq k \leq K \tag{6}$$

$$\sum_{k}^{new} = (\frac{1}{N_k}) \sum_{i=1} N w_{ik}(x_i - \mu_k^{new})(x_i - \mu_k^{new})^t, 1 \leq k \leq K \tag{7}$$

For the all formulas, $N_k = \sum_{i=1} k w_{ik}$, which corresponds to the sum of membership weights for each of the k gaussian components (the amount of points assigned to being generated from source k). In the same fashion, N represents the total number of data points.

B. K-means algorithm

K-means is one of the most used clustering algorithms that can be used for brain tissue segmentation. It is an iterative algorithm that tries to partition the dataset into 'K' pre-defined distinct non-overlapping subgroups (clusters) where each data point belongs to only one group. The objective is to keep the intra-class cluster data as similar as possible and inter-cluster data as different as possible, so it tries to reduce intra-class variance whereas tries to increase inter-class variance. At each iteration, it assigns data points to the predefined number of clusters and tries to reduce the squared distance between the data and centroids as minimum as possible. The drawback of K-means compared to EM is that it assigns hard levels to the data points whereas EM does soft labeling. So the initialization step of the EM algorithm can be done by the hard levels assigned by K-means. The way K-means algorithm works is as follows: Lists are easy to create:

- Specify number of clusters K.
- Initialize centroids by first shuffling the dataset and then randomly selecting K data points for the centroids without replacement.
- Keep iterating until there is no change to the centroids. i.e assignment of data points to clusters isn't changing.
- Compute the sum of the squared distance between data points and all centroids.
- Assign each data point to the closest cluster (centroid).
- Compute the centroids for the clusters by taking the average of the all data points that belong to each cluster.

The approach K-means follows to solve the problem is called Expectation-Maximization. The E-step is assigning the data points to the closest cluster. The M-step is computing the centroid of each cluster. The objective function is:

$$J = \sum_{i=1}^{m} \sum_{k=1}^{K} w_{ik} \left\| x^i - \mu_k \right\|^2$$

Where x means the data points, μ represents the clusters and k denotes the number of clusters. It's a minimization problem of two parts. We first minimize J w.r.t. w_{ik} and treat μ_k fixed. Then we minimize J w.r.t. μ_k and treat w_{ik} fixed. Technically speaking, we differentiate J w.r.t. w_{ik} first and update cluster assignments (E-step). Then we differentiate J w.r.t. μ_k and recompute the centroids after the cluster assignments from the previous step (M-step). Therefore, E-step is:

$$\frac{\partial J}{\partial w_{ik}} = \sum_{i=1}^{m} \sum_{k=1}^{K} \left\| x^i - \mu_k \right\|^2$$

$$\Rightarrow w_{ik} = \begin{cases} 1 & \text{if } k = \text{argmin}_j \left\| x^i - \mu_j \right\|^2 \\ 0 & \text{otherwise.} \end{cases}$$

In other words, assign the data point xi to the closest cluster judged by its sum of squared distance from the cluster's centroid. And M-step is:

$$\frac{\partial J}{\partial \mu_k} = 2\sum_{i=1}^{m} w_{ik}\left(x^i - \mu_k\right) = 0$$
$$\Rightarrow \mu_k = \frac{\sum_{i=1}^{m} w_{ik} x^i}{\sum_{i=1}^{m} w_{ik}}$$

Which translates to re-computing the centroid of each cluster to reflect the new assignments.

III. Design and Implementation

The design and implementation of the solution for the lab consisted of several steps, which are listed and explained in detail in the following subsections.

A. Skull stripping

To begin with the segmentation process, the first step is to extract the brain data from the original T1 and T2-FLAIR Image. It is the process of isolating brain tissue from non-brain tissue from an MRI image of a brain. There will be erroneous segmentation if whole image data is used as there are six classes that can be visible in human brain images which are CSF, White Matter, Gray Matter, Bones, Air, and Soft Tissues of which CSF, White Matter, and Gray Matter are classes of interest. So the skull which is of no interest should be stripped to produce good results for segmentation. In this process, ground truths are used for simplicity for skull stripping. The results of both image modalities can be seen in Figure 1.

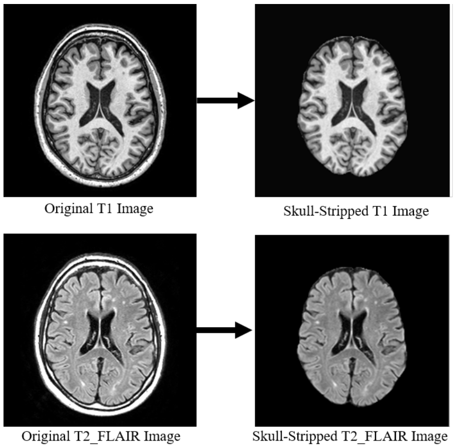

Fig. 1. Skull Stripping Process.

B. Random Initialization

To start with EM algorithm, firstly the unknown parameters (mean, covariance, and prior probabilities) should be initialized. There are two ways of initialization exploited in this work: Random and K-means. In Random initialization, for each tissue class, a random distribution of data (Normal) is generated, and the mean is initialized randomly taking a point from the data. Prior probabilities can also be initialized from the randomly generated data points. For variance computation, it is computed bearing two things in mind that is the variances of each class should be far enough so that they are not merged together and the matrices should be semi-definite or positive to deal with the singularity issue. Finally, for all modalities, same procedure is applied and concatenated for each of the unknown parameters. The system is robust and can be applied to any number of image modalities used.

C. K-means Initialization

K-means clustering is the other method that can be used for initializing unknown parameters. The hard levels assigned to data points by K-means can be used for computing the mean, covariance, and prior probabilities of each cluster. Firstly, K, the number of clusters is specified which is 3 in this case and the points are assigned to the clusters in an iterative process until the algorithm reaches convergence. Kmeans Clustering is used from sklearn.cluster.KMeans. The Inputs are : n-clusters= Number of cluster, K-means++: initial cluster centers for k-mean clustering in a smart way to speed up convergence, Random State : Determines random number generation for centroid initialization. The outputs are the centroids and the levels assigned by K-means. Th flowchart for K-means algorithm can be demonstrated by the Figure 2.

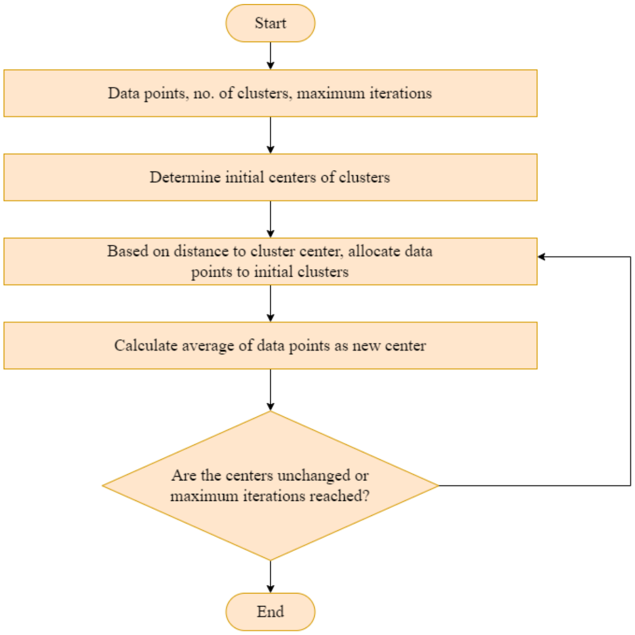
Start
Data points, no. of clusters, maximum iterations
Determine initial centers of clusters
Based on distance to cluster center, allocate data points to initial clusters
Calculate average of data points as new center
Are the centers unchanged or maximum iterations reached?
End

Fig. 2. Flowchart for K-means algorithm.

One problem with K-means is that it assigns the clustered index randomly at each run. To make the level assignment robust, a labeling trick is used. The trick is to sort the mean intensity values each time and assign the labels in ascending order to CSF, GM and WM. In this way, the robust assignment of the labels is ensured.

D. EM Algorithm

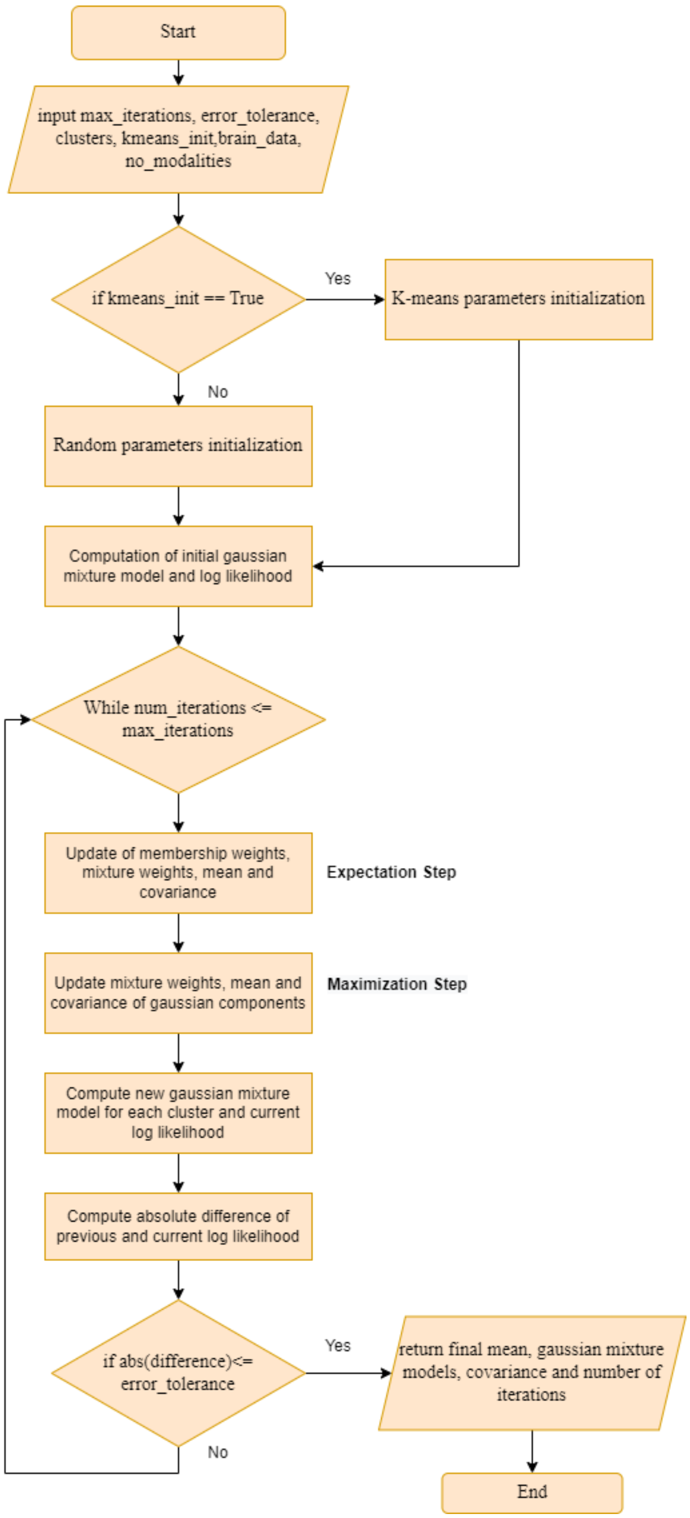
Start
input max_iterations, error_tolerance, clusters, kmeans_init,brain_data, no_modalities
if kmeans_init == True
Yes
K-means parameters initialization
No
Random parameters initialization
Computation of initial gaussian mixture model and log likelihood
While num_iterations <= max_iterations
Update of membership weights, mixture weights, mean and covariance
Expectation Step
Update mixture weights, mean and covariance of gaussian components
Maximization Step
Compute new gaussian mixture model for each cluster and current log likelihood
Compute absolute difference of previous and current log likelihood
if abs(difference)<= error_tolerance
Yes
return final mean, gaussian mixture models, covariance and number of iterations
No
End

Fig. 3. Flowchart for EM algorithm

The figure 3 presents the flowchart of the implementation design for the EM algorithm, following the theoretical algorithm analysis of the previous section. As shown, the main stages of the algorithm are the initialization of parameters, expectation and maximization. The two stopping criteria for the algorithm are: number of iterations and absolute differences of log-likelihood between iterations (this one used for checking convergence if reached before the maximum number of iterations initially defined as input for the algorithm).

E. Image Reconstruction

After the EM step, the posterior probabilities for each of the clusters for each data point are generated. The maximum of the posterior probabilities from each cluster is chosen and the index of that maximum value is assigned to a data point. Thus all the data points are assigned to their labels. 1 is added to all the labels to comply them with the ground truth as python indices start from 0. CSF, GM, and WM tissue classes are assigned 1, 2, and 3 respectively and using the function slice-show image slices can be visualized with the help of image visualization libraries.

IV. Experimental Results and Analysis

The constructed pipeline for the solution of the lab is very robust and allows the user to define the following important parameters from the beginning:

- Patient for which the experiment is being done.
- Desired initialization (Random or Kmeans)
- Simple modality or multimodality (T1T2FLAIR)

Once understanding how the pipeline works, the following set of experiments was run:

- **Experiment 1:** Evaluation of performance of the EM algorithm for all patients using single modality (T1) and the two initialization methods: k-means and random
- **Experiment 2:** Evaluation of performance of the EM algorithm for all patients using single modality (T2FLAIR) and the two initialization methods: k-means and random.
- **Experiment 3:** Evaluation of performance of the EM algorithm for all patients using multimodality (T1T2FLAIR) and the two initialization methods: k-means and random.
- **Experiment 4:** Evaluation comparison of performance between K-means as a standalone segmentation method and after applying the EM algorithm.

All the evaluations of performance were made using three metrics: number of iterations until convergence, computational time and dice segmentation score.

1) Results Analysis for Experiment 1: The table I presents the quantitative performance results obtained for the first experiment. From this table, it can be seen that both methods of parameters initialization (Random and K-means) converge to the same dice segmentation score for all the three tissues. The dice scores are good (with the lowest one being 0.773 for grey matter segmentation in patient one). However, the big difference that must be marked is that for all patients the K-means initialization converges faster (meaning less number of iterations and less computational time). In some cases, such as with patient 1, the difference in the velocity of convergence is big (more than half), and in other cases like in patient 5 it is much more smaller.

2) Results Analysis for Experiment 2: The table II presents the quantitative performance results obtained for the second experiment. From this table, it can be seen that for most of the cases (such as patient 1, 3, 4 and 5), k-means initialization makes the EM algorithm to converge slower than random initialization. Furthermore, the dice scores are very bad (some of them even close to zero), which is theoretically expected as T2 FLAIR modality is not used for anatomic segmentation standalone.

3) Results Analysis for Experiment 3: Table III presents the quantitative performance results obtained for the third experiment. From this table, it can be noticed that when combining the 2 modalities, the number of iterations for all cases decreases with respect to single-modality experiments. Moreover, the dice score results for k-means initialization are considerably higher than with single modalities. This does not happen for random initialization, where the dice scores get lower values when compared with a single T1 modality.

4) Results Analysis for Experiment 4: Table IV presents a comparison of quantitative performance (dice segmentation score) when performing the segmentation with K-means (initialization step) and after running the EM algorithm for all the 3 tissues of interest. It can be noticed a remarkable improvement in dice score for all tissues after applying the EM algorithm except for patient 2 in gray matter and White matter tissues. This might be due to bad image acquisition or the presence of noise in the image which can reduce the partial volume effect.

5) Visual Results Presentation: A set of different pictures are presented in order to provide some visual results for interesting cases. All the images correspond to the case of patient 5. The figure 4 shows the visual result of the segmentation when performing K-means in an initial step and when performing the EM algorithm. Figure IV-5 presents the visual results for random initialization. The first row corresponds to the ground truth for the segmentation of all tissues, and the second, third and fourth label correspond to the segmentation results obtained for random initialization when using T2, T1 and multimodality. Finally, figure IV-5 presents the visual results for K-means initialization before and after applying EM, respectively. The first pair of rows corresponds to the ground truth, whereas, the second, third and fourth pairs represent the results obtained when using single modality (T1, T2) and multimodality.

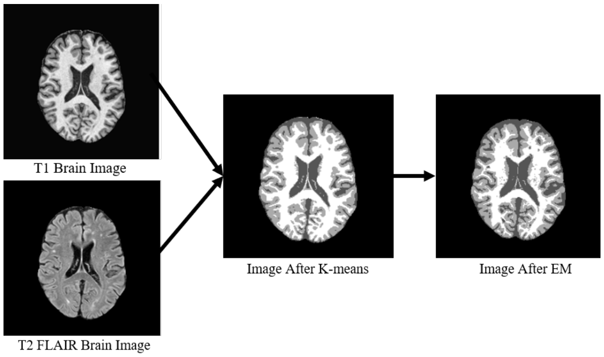

Fig. 4. Kmeans and EM Results.

V. Project Management and Details

Some problems arose during the lab work like Randomness in K-means label assignment, singularity in covariance computation, and bad initialization during the random process. The problems were solved later which took some for us to perform the proper investigation to make the whole process robust and dynamic. Other than that, the workload has been divided equally between the partners both for coding and writing part.

VI. Conclusions

- During this lab, a robust implementation of EM algorithm from scratch (supporting multimodality) was successfully implemented. During the development of the algorithm, most of the difficulties were faced when trying to make it suitable to handle both 1D and 2D data (single and multimodality).
- From the quantitative performance results presented in the tables, it can be concluded that using multimodality (T1T2FLAIR) and K-means initialization has a positive impact in the evaluation metrics for the segmentation of all tissues and for most of the patients (except for patient 2, which might be caused due to bad image acquisition), since the dice scores are considerably bigger. Also, using multiple image modalities helps the algorithm to converge faster when compared to the number of iterations performed with single modalities.
- During the development of the experiments it was noticed and therefore can be concluded that for T1 and and T11T2 modalities K-means helps the EM algorithm to converge faster (in less time and with fewer iterations).
- With the quantitative results obtained it was theoretically proven that T2FLAIR image modality cannot be used alone for anatomic segmentation, since the dice scores obtained are very low. On the other case, the quantitative results obtained with only T1 were good but the highest one was gotten when combining both modalities.+

References

[1] F. Dellaert, “The expectation maximization algorithm,” Georgia Institute of Technology, Tech. Rep., 2002.

[2] X. Lladó, “Misa: Image segmentation,” in *Lecture Slides*, 2022.

TABLE I
COMPARISON TABLE FOR T1 IMAGE MODALITY.

Patient Number	Initialization Method	No. of Iterations	Computational Time (s)	Dice Score		
				T1 Modality		
				CSF	GM	WM
1	K-means	114	20.7651	0.782	0.773	0.856
	Random	313	67.8352	0.782	0.773	0.856
2	K-means	119	23.6108	0.851	0.802	0.774
	Random	290	38.2567	0.851	0.802	0.774
3	K-means	112	22.3376	0.811	0.779	0.839
	Random	210	25.6992	0.811	0.776	0.839
4	K-means	120	20.8185	0.805	0.807	0.869
	Random	222	24.1552	0.8054	0.807	0.869
5	K-means	114	21.6153	0.802	0.847	0.887
	Random	203	24.9179	0.802	0.847	0.887

TABLE II
COMPARISON TABLE FOR T2FLAIR IMAGE MODALITY.

Patient Number	Initialization Method	No. of Iterations	Computational Time (s)	Dice Score		
				T2 Modality		
				CSF	GM	WM
1	K-means	408	69.8644	0.685	0.621	0.011
	Random	313	36.2429	0.685	0.621	0.011
2	K-means	330	59.58	0.231	0.046	0.494
	Random	373	49.186	0.494	0.214	0.483
3	K-means	405	67.662	0.719	0.612	0.07
	Random	302	38.8831	0.719	0.612	0.07
4	K-means	417	64.4028	0.538	0.082	0.481
	Random	239	26.5665	0.538	0.082	0.481
5	K-means	500	82.8653	0.756	0.692	0.004
	Random	352	43.2077	0.756	0.692	0.004

TABLE III
COMPARISON TABLE FOR MULTIPLE IMAGE MODALITY.

Patient Number	Initialization Method	No. of Iterations	Computational Time	Dice Score		
				Both Modalities		
				CSF	GM	WM
1	K-means	67	22.404	0.906	0.821	0.849
	Random	187	46.1346	0.566	0.215	0.625
2	K-means	314	99.7585	0.846	0.173	0.527
	Random	321	83.6958	0.846	0.173	0.527
3	K-means	64	22.7996	0.854	0.778	0.837
	Random	235	51.7247	0.863	0.078	0.612
4	K-means	61	19.4919	0.9	0.839	0.868
	Random	318	66.2885	0.496	0.186	0.571
5	K-means	78	25.2641	0.865	0.846	0.889
	Random	133	29.7745	0.865	0.846	0.889

TABLE IV
K MEANS VS EM COMPARISON TABLE.

Patient Number	After K-means			After EM		
	Dice Score			Dice Score		
	CSF	Gray Matter	White Matter	CSF	Gray Matter	White Matter
1	0.819	0.69	0.812	0.906	0.821	0.849
2	0.812	0.668	0.733	0.846	0.173	0.527
3	0.792	0.682	0.825	0.854	0.778	0.837
4	0.822	0.694	0.782	0.9	0.839	0.868
5	0.856	0.8	0.844	0.865	0.846	0.889

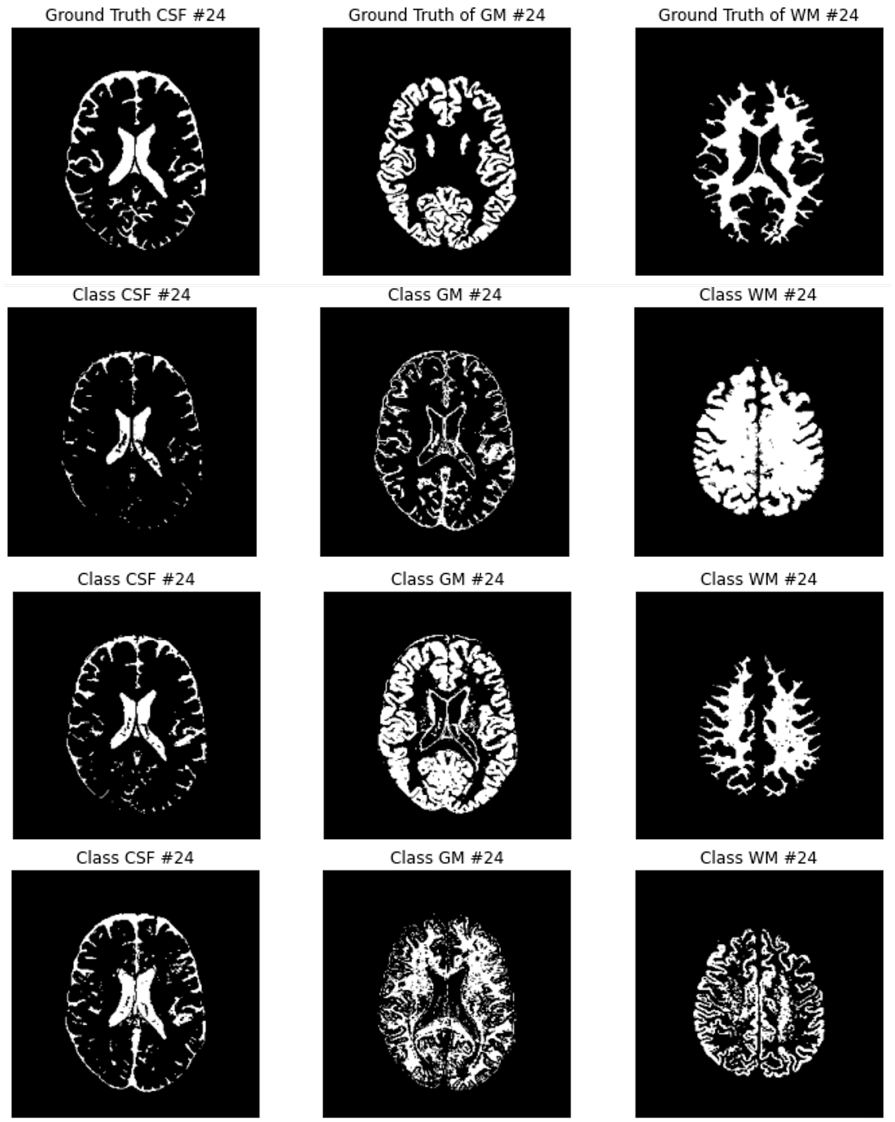

Fig. 5. Results for Random Initialization (GT, T2, T1 and MultiModality)

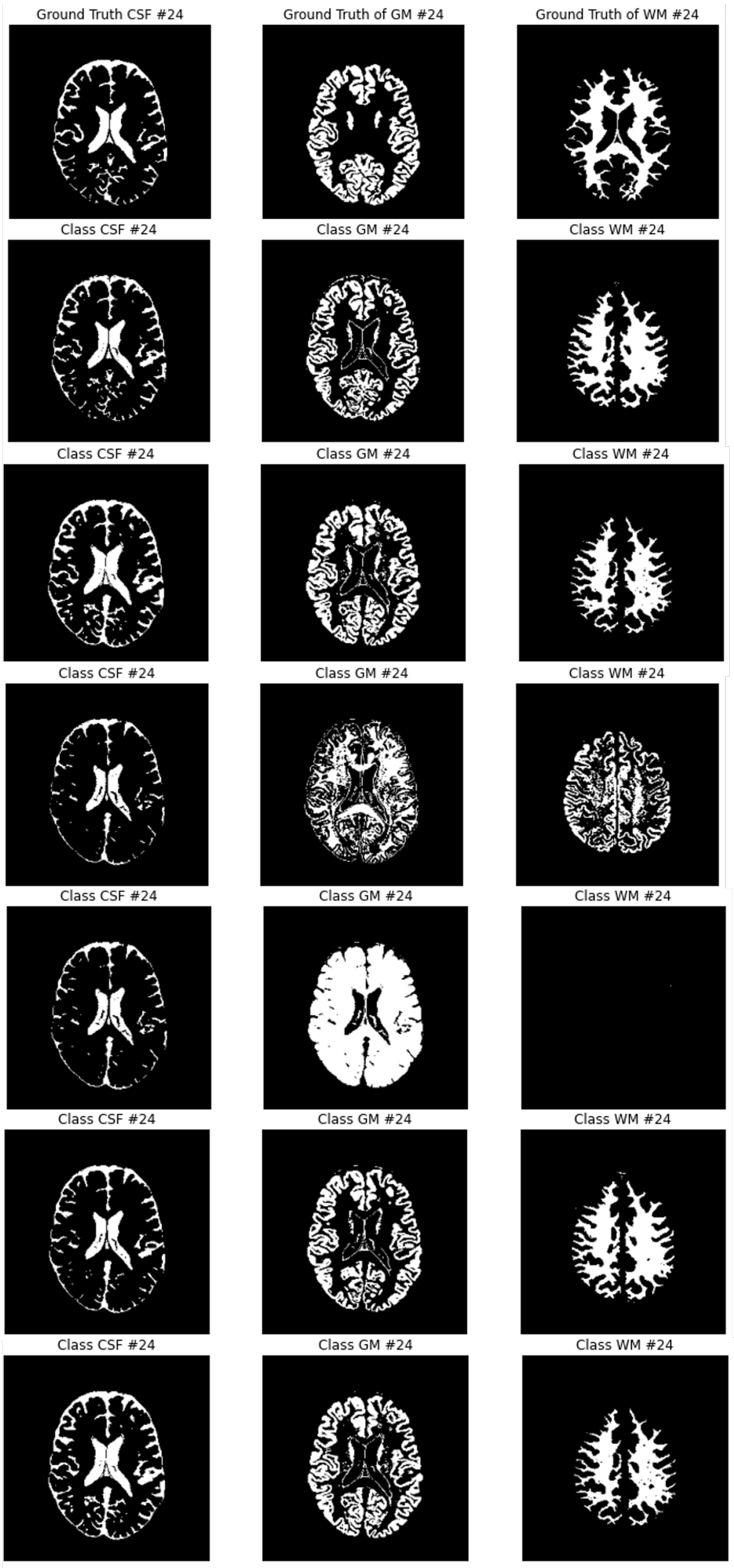

Fig. 6. Ground Truth and Kmeans-EM Pair(GT, T1, T2 and MultiModality)