Medical Image Segmentation and Applications
Lab 2: Image Segmentation (EM algorithm)

Submitted By

Md. Kamrul Hasan
Basel Alayfi
Fakrul Islam Tushar

Submitted To
Xavier Llado, PhD
xavier.llado@udg.edu

November 4, 2018

Contents

1 Introduction and Problem Definition

The problem definition is to implement from scratch the algorithm of Expectation Maximization using Matlab. This algorithm has been applied on brain images (T1 and FLAIR). Three regions have to be segmented, the cerebrospinal fluid (CSF), the gray matter (GM), and the white matter (WM). All the equations used were taken from MISA course slides, see [1].

2 Algorithm Analysis

All the phases of the algorithm are discussed here with the equations needed. Starting from initialization, expectation, maximization, and stopping criteria. To make all the symbols used clear, a brief definition is given in Table 1. Three Gaussian Mixture Models are assumed. Each pixel is processed and the probability of belonging to each mixture model is calculated. The maximum probability of every observation is found and the corresponding index is considered the predicted label what is called soft assignment of pixel label.

Symbol	**meaning**
N	number of observations (relative to the context)
K	number of distinct classes
μ_k	the k^{th} class mean
Σ_k	the k^{th} class covariance matrix
α_k	the k^{th} mixture weight
w_ik	the probability that x^i belongs to class k
$N-k$	the effective number of observation belonging to class k

Table 1: Symbols definition

2.1 Initialization

To initial the algorithm, two methods can be used, namely: Kmeans and random initialization. Kmeans is usually more precise so the algorithm needs less iterations to reach the convergence which is defined as a negligible increase in the likelihood. After that, the mean (returned as centroids by the kmeans++ or picked randomly if random initialization) and covariance matrix for each class are known and a Maximization step is processed. Starting from the second iteration, expectation then maximization sequence is followed until one of the convergence criteria is satisfied.

2.2 Expectation

In this step, the weights for every pixel being belonged to each class are calculated. Those weights are computed using the Gaussian mixture model multiplied (eq(3)) by alpha (which is in the first iteration, the proportion of the corresponding class :number of class elements divided by the total number of elements). To use the mixture model, the mean and covariance matrix for each class are computed using equations (1, 2), respectively. The new weights are computed via eq(4).

$$\mu_k^{new} = \frac{1}{N_k}\sum_{i=1}^{N} w_{ik} \, . \, x^i \tag{1}$$

$$\sigma_k^{new} = \frac{1}{Nk}\sum_{i=1}^{N} w_{ik}.(x^i - \mu_k^{new})(x^i - \mu_k^{new})'; 1 \leq k \leq K \tag{2}$$

$$p(x|\theta_k) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \, e^{-\frac{1}{2}(x-\mu_k)' \, \Sigma_k^{-1} \, (x-\mu_k)}; \; \theta_k := \mu_k, \Sigma_k \tag{3}$$

$$w_{ik} = \frac{p(x_i|\theta_k).\alpha_k}{\sum_{m=1}^{K} p(x_i|\theta_m).\alpha_m} \tag{4}$$

2.3 Maximization

In this step, the new mixture weights are computed via the following set of equations. Where, N_k is the effective number of observations belonging to the k^{th} class eq(5), and alpha is updated using eq (6).

$$N_k = \frac{1}{N}\sum_{i=1}^{N} w_{i,k} \tag{5}$$

$$\alpha_k = \frac{N_k}{N} \tag{6}$$

2.4 Stopping Criteria

To recognize convergence, a small change (if any) of the order of 10^{-3} in the log likelihood is used, while, a maximum number of iterations is used to assure the stability. The log likelihood is defined in eq (7).

$$\log l(\Theta) = \sum_{i=1}^{N} \log \sum_{k=1}^{K} \alpha_k . p(x_i|\theta_k) \tag{7}$$

3 Design and Implementation of the Proposed Solution

The work-flow that has been done is shown in Fig. 1 . Firstly, from both T1 and FLAIR MRI image, region of interest (ROI) has been extracted using the ground truth image. ROI selection is done by neglecting the background pixels (labeled as zeros in the ground truth). Following pseudocode used for the extraction of ROI from both T1 and FLAIR MRI image.

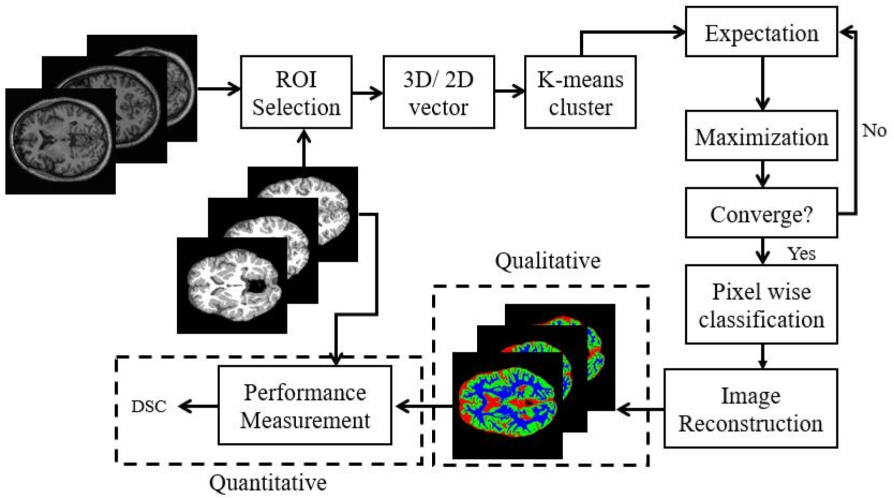


Figure 1: Pipeline for brain tissue segmentation using EM algorithm

Algorithm 1 Region of Interest (ROI) selection

```
$GT\_Image \leftarrow$ Read Ground Truth Image
$Target\_Image \leftarrow$ Read Target Image (both T1 and FLAIR MRI)
for <Looping for all slices> do
    $ROI\_Index \leftarrow find(GT\_Image(slice) \neq 0)$
    $ROI\_Image \leftarrow$ Target_Image(ROI_Index)
end for
```

After selecting ROI, feature vector has been created which have NxD dimension. Where, N indicates the numbers of pixels inside the ROI and D indicates dimension of the feature vector which is 2 (T_1 weighted and FLAIR weighted MRI). In 3D implementation, N is the total numbers for pixels inside the ROI for all slices, while in 2D implementation, N is the total pixels inside the ROI for one slice inside the loop of slice by slice processing.

k-means clustering has been used to get the initial parameters, i.e., the mean of each cluster, covariance matrices and cluster priorities. For different runs, k-means assigns different cluster labels randomly. But, in the ground truth, cluster labels are fixed, i.e., CSF=1, GM=2, and WM=3. To solve this random cluster labeling caused by k-means, the following pseudocode has been used. This pseudocode fixes the labels i.e. CSF=1, GM=2, and WM=3.

Algorithm 2 Fixing cluster label e.g. CSF=1, GM=2, and WM=3

```
$[cluster\_indices, cluster\_center] \leftarrow$ k-means (Feature Vector)
$[new\_cluster\_center, cluster\_center\_label] \leftarrow$ sort (cluster_center)
for <i=1:1:Total_cluster> do
    $temp\_label \leftarrow cluster\_center\_label(i))$
    $temp\_label\_index \leftarrow$ find(temp_label==cluster_indices)
    $new\_cluster\_indices(temp\_label\_index) \leftarrow$ i
end for
```

The main clue for fixing the cluster labels is that in case of T1 weighted MRI the mean pixel intensity for CSF is the lowest, for GM is higher, and the highest is for the WM.
After fixing the cluster label, parameters, i.e., mean of each clusters, covariance matrices and cluster priorities have been initialized according to the eq. (1), eq. (2), and eq. (5, 6). Taking the those initial parameters, expectation step has been done which is evaluating the responsibilities using the current parameters. Using responsibilities, in the maximization step parameters have been updated using the equation eq. (1 to 6). If the updated responsibilities from new parameters satisfy the convergence criterion, iteration has been stopped.
Using those updated parameters from maximization step, pixel wise classification has been done that will softly assigned pixels label from probability belonging to each clusters. The following pseudocode used for the pixel wise classification with soft assignment.

Algorithm 3 Pixel wise classification and soft assignment of pixel label

```
$Pixel\_Classification \leftarrow$ zeros(length_ROI_pixel,1)
for <i=1:1:length_ROI_pixel> do
    $Each\_observation \leftarrow ROI\_vector(i))$
    $GMM\_pdf \leftarrow Gaussian\_Mixture(Each\_observation, parameters))$
    $Pixel\_label \leftarrow max(GMM\_pdf))$
    $Pixel\_Classification \leftarrow$ Pixel_label
end for
```

In the image reconstruction step, those labeled pixels are added to the 1D vector in the previously recorded index position and others index filled with the zeros that are responsible for background. Afterward, 1D vector has been reshaped to get original 2D slice of the segmented image. This segmented slice is the qualitative results and to calculate the quantitative results those segmented slices along with corresponding ground truth slices have been passed to the dice co-efficient function that will return numerical value of dice co-efficient for each slice for three different tissues (CSF, GM, and GM).

4 Experimental Results Analysis

For running the algorithm over all the slices, two methods were considered, slice by slice or what is called 2D, and the 3D way , which takes all the useful pixels from all slices in one shot. Next, each method is discussed and representative cases are shown.

4.0.1 Results for 2D Implementation

In 2D implementation, slice 25, which is considered in the heart of the 3D object and includes a relatively-large number of tissue pixels, is tested in patients 1, 2, 3, 4, and 5.

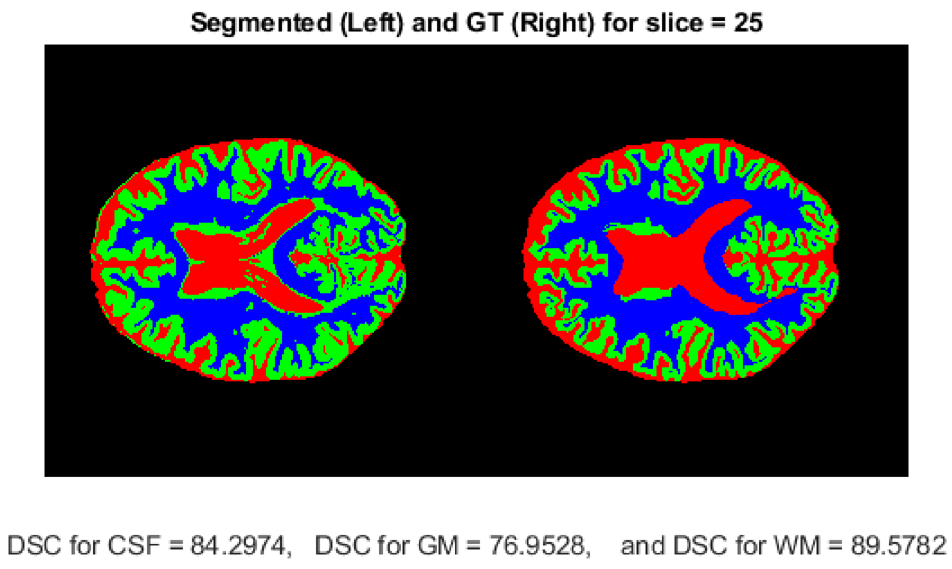

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF = 84.2974, DSC for GM = 76.9528, and DSC for WM = 89.5782

Figure 2: Results for Slice 25, patient 1, 2D method

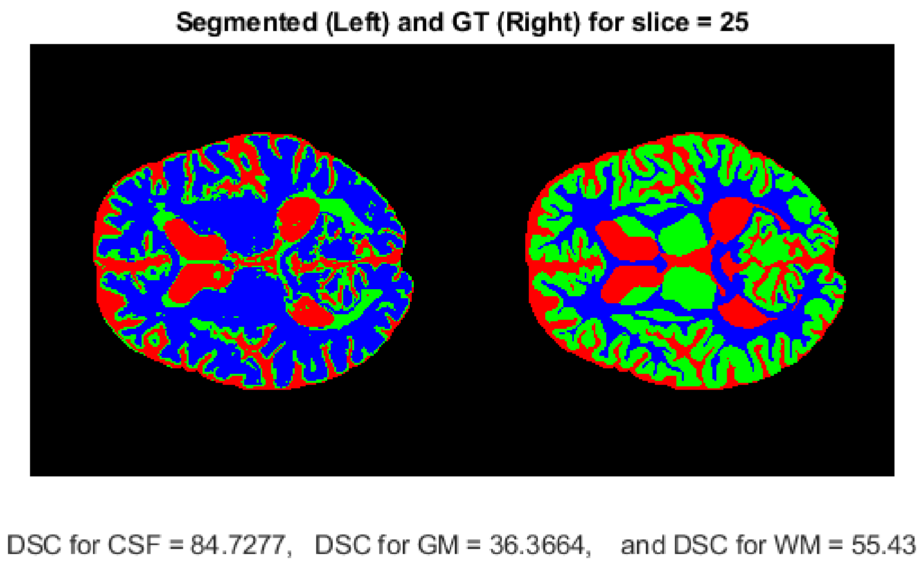

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF = 84.7277, DSC for GM = 36.3664, and DSC for WM = 55.43

Figure 3: Results for Slice 25, patient 2, 2D method

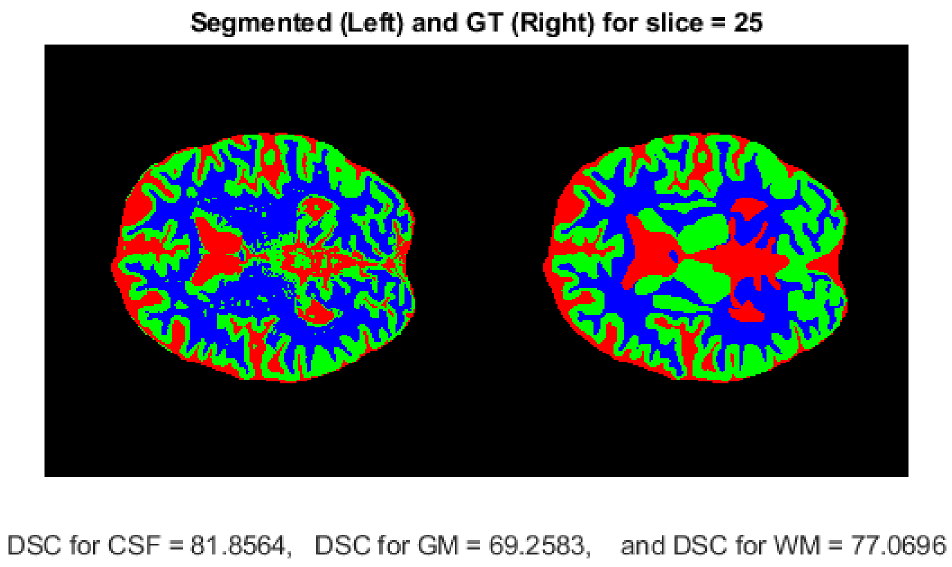

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF = 81.8564, DSC for GM = 69.2583, and DSC for WM = 77.0696

Figure 4: Results for Slice 25, patient 3, 2D method

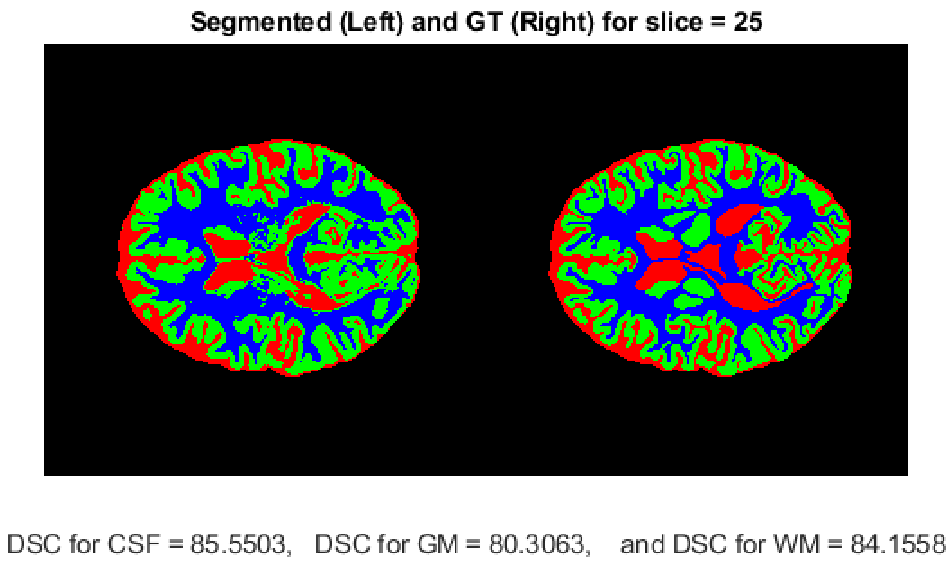

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF = 85.5503, DSC for GM = 80.3063, and DSC for WM = 84.1558

Figure 5: Results for Slice 25, patient 4, 2D method

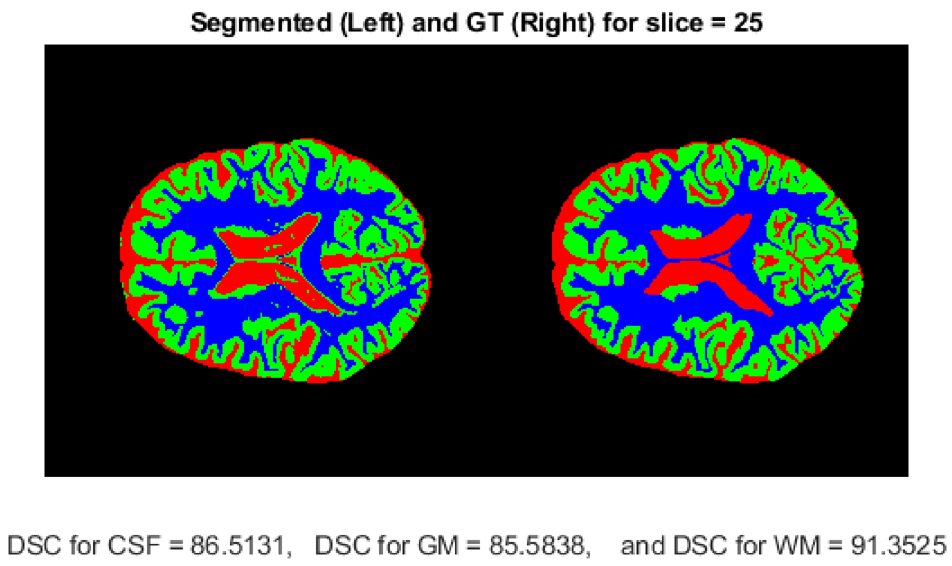

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF = 86.5131, DSC for GM = 85.5838, and DSC for WM = 91.3525

Figure 6: Results for Slice 25, patient 5, 2D method

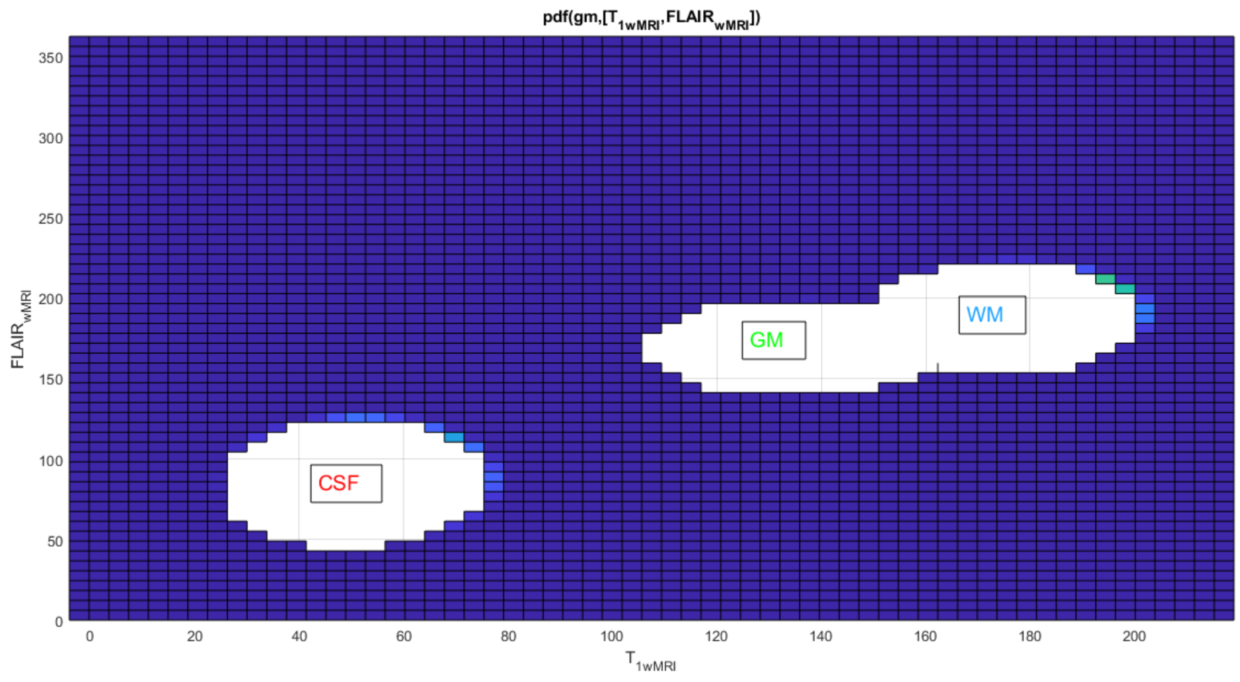


Figure 7: Joint Mixture Models for T1 and FLAIR for patient 5 (slice number=25)

Figure 2 shows the Dice Similarity Coefficient values for the three regions. As can be seen from the figure, the algorithm did a great job in classifying the pixels. Quantitatively and qualitatively, the results are acceptable. Figure 3 shows that the results are less precise than patient 1, especially for the gray matter. One reason is that some regions overlap (mainly, gray matter and white matter) in a sense which makes it difficult for the algorithm to decide. To show the evidence for that, Figure 7 show the joint mixture models for T1 and FLAIR for patient 2. Figure 4 demonstrates that the algorithm did a bit better than patient2. Figure 5 and Figure 6 shows relatively good results in 2D implementation slice by slice in one shot.

4.0.2 Results for 3D Implementation

For 3D, all the slices were taken in one round and reshaped as one vector (N x 2). The vector was fed to the algorithm which dealt with it as a single picture. After the algorithm has finished, the result for slice 25 was taken and judged.

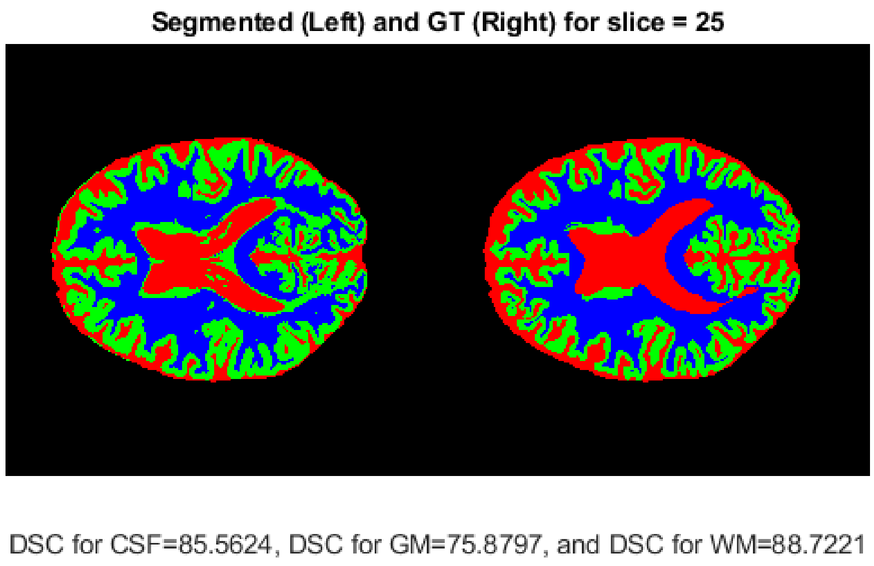

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF=85.5624, DSC for GM=75.8797, and DSC for WM=88.7221

Figure 8: Results for Slice 25, patient 1, 3D method

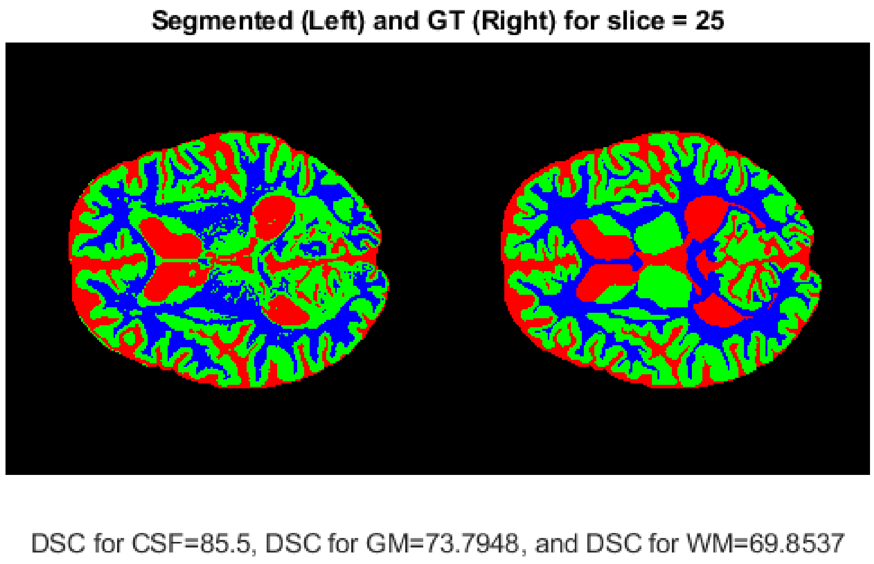

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF=85.5, DSC for GM=73.7948, and DSC for WM=69.8537

Figure 9: Results for Slice 25, patient 2, 3D method

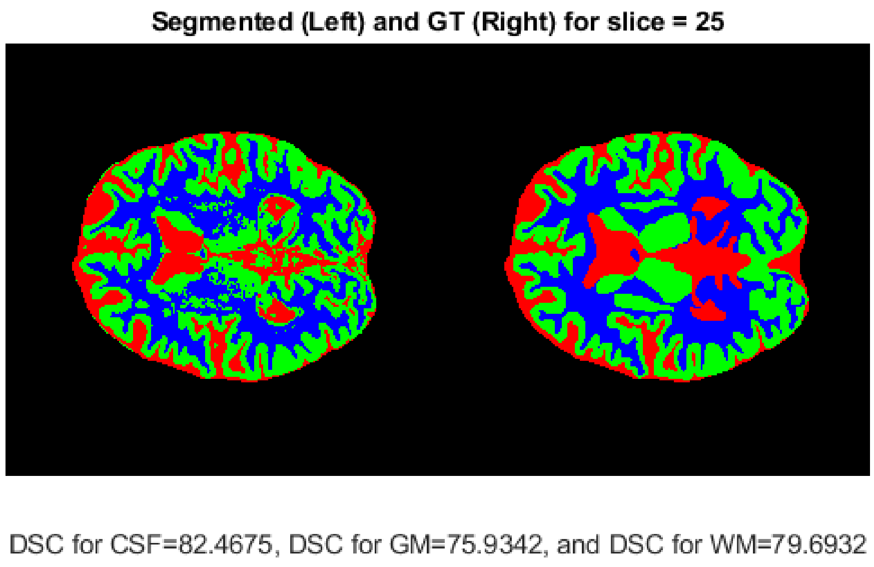
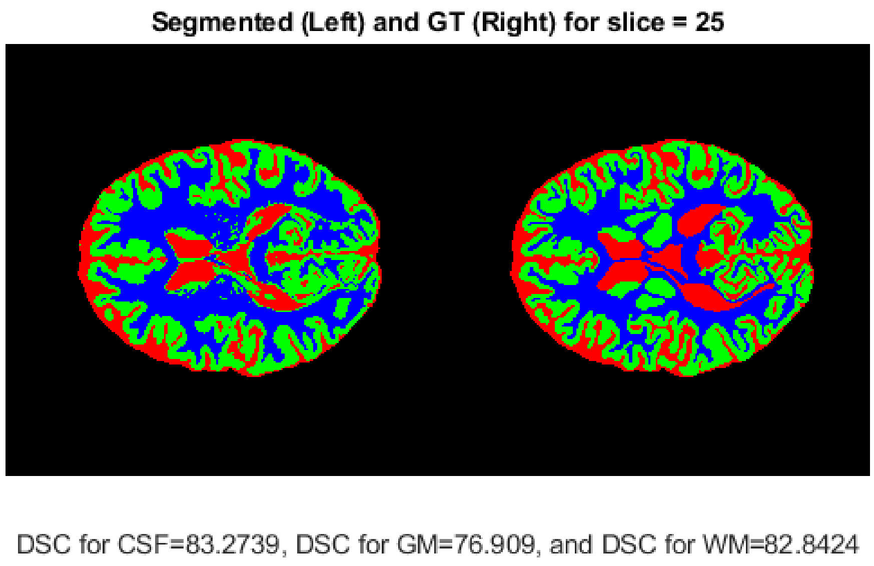
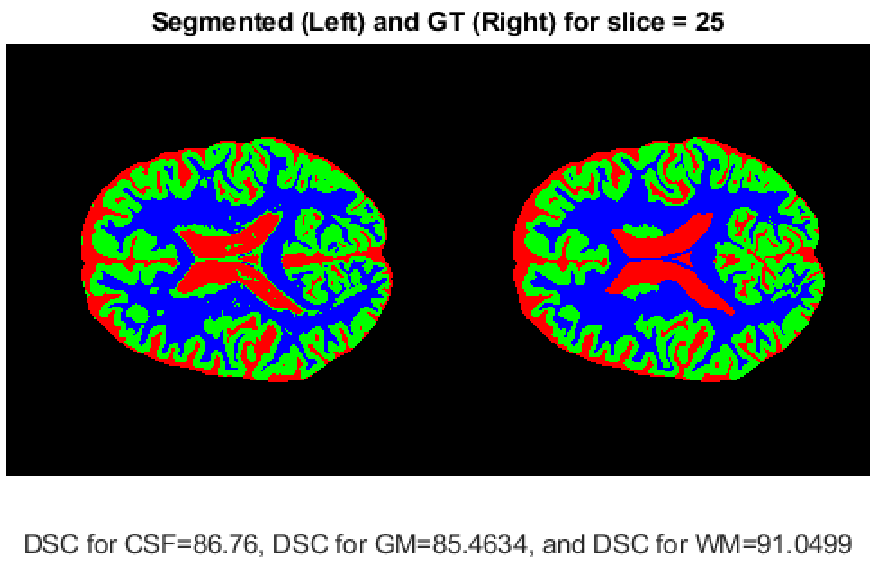

Segmented (Left) and GT (Right) for slice = 25
DSC for CSF=82.4675, DSC for GM=75.9342, and DSC for WM=79.6932

Figure 10: Results for Slice 25, patient 3, 3D method



DSC for CSF=83.2739, DSC for GM=76.909, and DSC for WM=82.8424

Figure 11: Results for Slice 25, patient 4, 3D method



DSC for CSF=86.76, DSC for GM=85.4634, and DSC for WM=91.0499

Figure 12: Results for Slice 25, patient 5, 3D method

As can be seen in Figures 8 to 12, results are pretty acceptable, in terms of patient 2, which had worst results in 2D, now results are much better. patient 5, was the easiest for the algorithm to segment. Patient 1, 3, 4 gave logical results as well.

4.0.3 Comparative Analysis between 2D and 3D Implementations and Conclusion

One among 2D and 3D implementations, *Which implementation does work better?*. To provide this answer, instead of considering one slice, we have taken the average DSC for each region and all slices and patients, their average convergence rate and computation times will be shown. To draw the conclusion of this sub-section, and by examining Figure 13 and Figure 14, the 3D implementation proved to be by far better than the 2D one. It is seen that for each tissue and for each patient, 3D implementation provides better avg. quantitative performance over 2D implementation, moreover, the convergence rate for the 3D implementation is faster than 2D implementation. In conclusion, 3D implementation has outperformed dramatically the 2D one in a relatively-short time, and EM gave decent results.

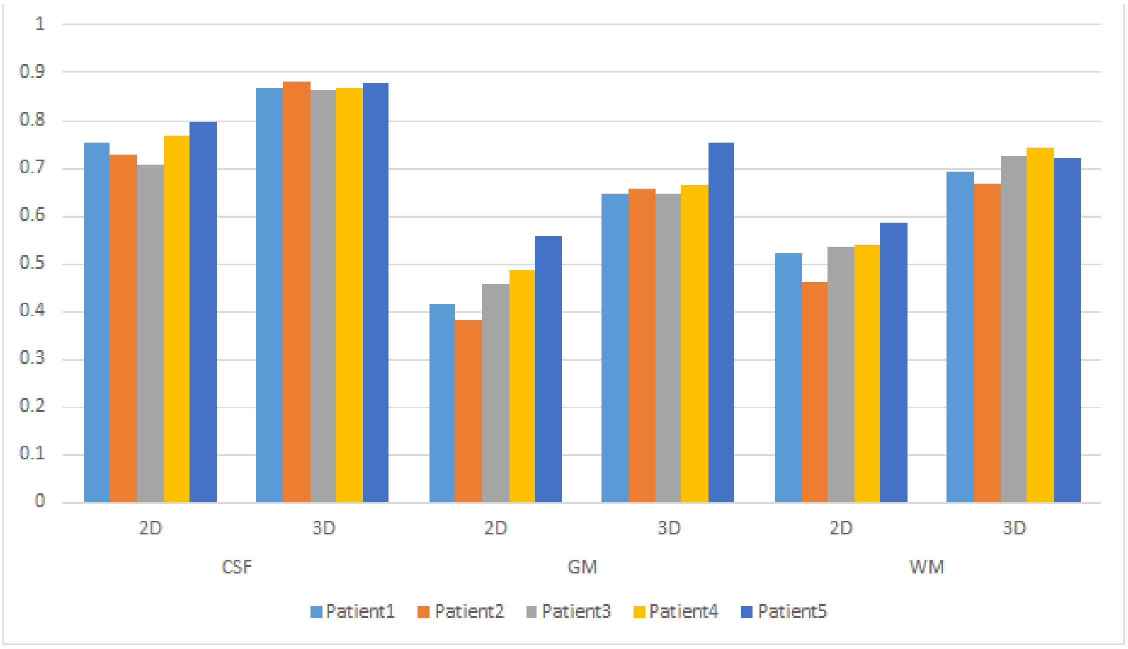
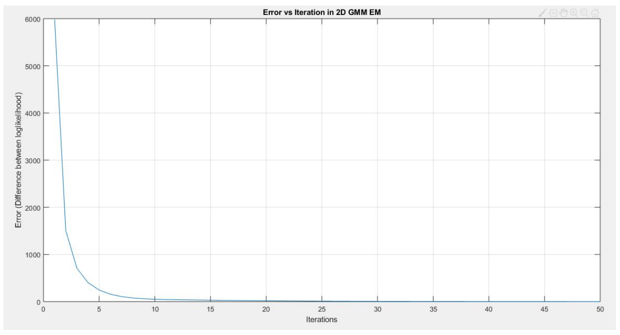
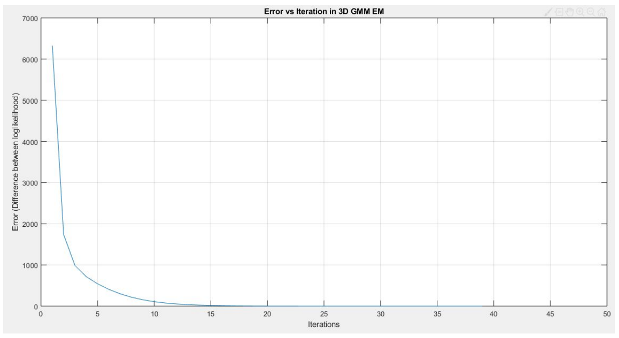


Figure 13: Comparison between average DSC of 2D and 3D implementations for all regions, slices, and patients



(a) for 2D implementation



(b) for 3D implementation

Figure 14: Error (Difference between log likelihood) VS iterations for 2D (left) and 3D (right) implementation

5 Project Management

Working individually and working in a group are two completely new experiences. The aim of this lab group task is to make us use to with the work environment where we need to work with a big team in most cases. A very important part of group task is to carry continuous communication and keep every group member up-to-data about the process. Figure 15 below shows how we perform communication among group and distribution of time spend to complete the lab task. During lab hours we worked together. We carried out communication of our findings and problem faced through Facebook messenger group (called “Team Work”). We used Google-drive to share our functions and codes to each-other. Finally, report was written using share-latex so that everyone can contribute and simultaneously observe others activity. About time distribution approximately 70% time spend on understanding the problem and doing coding, 5% time is been spend for optimization of the code and 20% was spend on report writing.

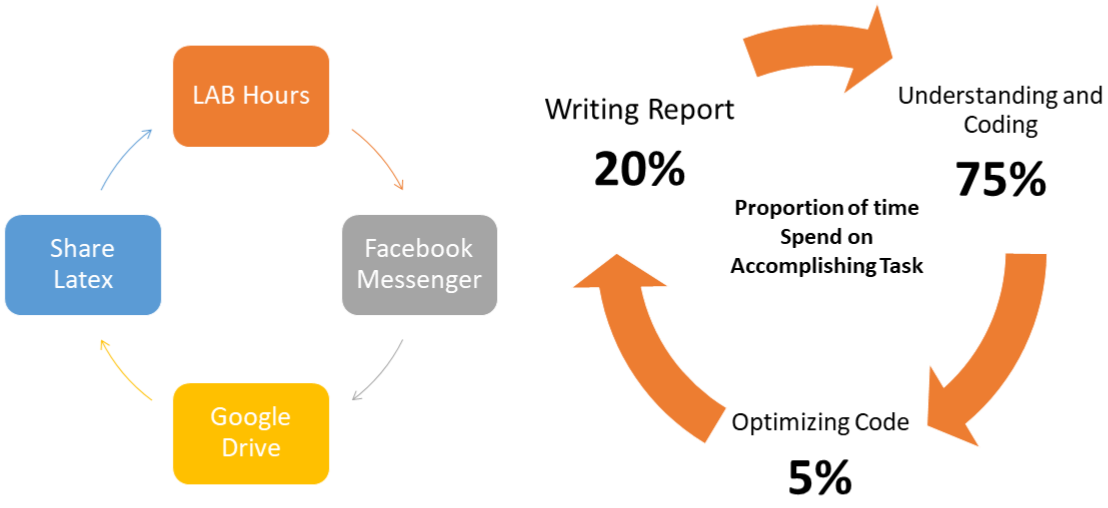


Figure 15: Project management flow.

References

[1] Class Lecture: Xavier Llado,“Image Segmentation,”pp. 44.