CS 736

Assignment - 3

By: Ank Kumar Gupta (22B0623) & Veenus (22B0704)

Overview

- **Objective:** Brain MRI Segmentation:FCM, Artifact Modeling, HMRF-GMM-EM

- **Tasks:**
  1. Segmenting a Brain Magnetic Resonance (MR) Image (Use of modified fuzzy-c-means (FCM) algorithm)
  2. Segmenting a Brain Magnetic Resonance (MR) Image (Use of expectation maximization (EM) optimization algorithm that relies on a Gaussian mixture model (GMM) for intensities and a Markov random field (MRF) model on the labels)

Task 1

Objective

Brain MR Image Segmentation (CFS, GM & WM) using FCM Modified Algorithm

Description

- Finding optimal value of class means, memberships and bias field.
- Finding optimal value of beta parameter.
- Constructing the Optimal class-membership image estimates, Optimal bias-field image estimate, Bias removed image & Residual image
- Finding the optimal estimates for the class means
- Discussion of whether the formulation discussed in class leads to a unique solution, if not, then proposing a scheme with implementation

Task 1

Approach

- Loaded the bias- and noise-corrupted magnitude MR image and its corresponding binary brain image mask.
- Ensured all processing to the pixels within the brain (using the mask).
- Initialized the bias field b as a constant image (e.g., all ones) and $q = 2$.
- Initialized fuzzy memberships u_{nk} based on intensity thresholds (i.e. using the 33rd and 66th percentiles to roughly separate CSF, GM, and WM).
- Computed initial class means c_k as a weighted average of the bias-corrected intensities within the brain.
- Create a 7×7 Gaussian kernel to serve as the neighborhood mask for bias field smoothing.

Task 1

Approach

- Updated class means, memberships and bias field for each iteration.
- Also computed the objective function at each iteration to monitor convergence, where the used objective function is as below:

$$J = \sum_{n \in \text{brain}} \sum_{k=1}^{K} u_{nk}^{q} (Y_n - b_n\, c_k)^2$$

- Construct the bias-removed image A ($A_n := \sum_k u_{nk} c_k$) and the residual image R ($R_n := Y_n - A_n b_n$).
- Constructed all the required images (original, membership maps, bias-removed, and residual).
- Tried to find out optimal q by manual tuning and visualization.

Result

1(a): Class Means Update at each iteration to get optimality:

$$c_k = \frac{\sum_{n \in \text{brain}} u_{nk}^q \, b_n \, Y_n}{\sum_{n \in \text{brain}} u_{nk}^q \, b_n^2}$$

1(b): Membership Update at each iteration to get optimality:

$$u_{nk} = \frac{1}{\sum_{j=1}^{K} \left(\frac{(Y_n - b_n c_k)^2}{(Y_n - b_n c_j)^2 + \epsilon} \right)^{\frac{1}{q-1}}}$$

1(c): Bias Field Update at each iteration to get optimality:

$$b_n = \frac{\sum_{k=1}^{K} u_{nk}^q c_k Y_n}{\sum_{k=1}^{K} u_{nk}^q c_k^2}$$

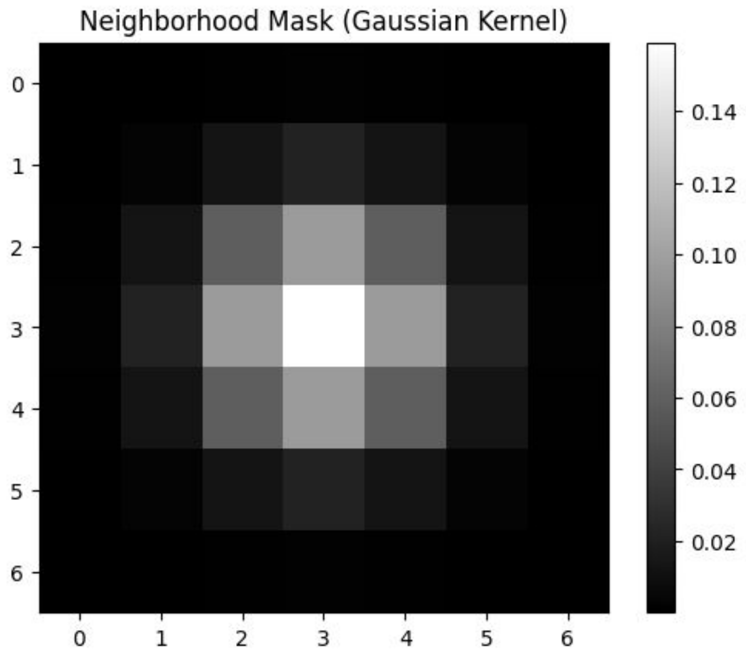
Result

1(a) Chosen Value of q: 2

- Chose the value as 2, as it seem offering a middle ground with moderate crispness and moderate cluster compactness.
- If chosen a value 1.5 or near to that, segmentation were very crisp but the separation between clusters were not so good.
- If chosen a value of 2.5 or more, the segmentation started to become more and more fuzzy resulting less distinct boundaries but cluster seemed well separated.

Result

1(b) The neighbourhood mask w_{ij}



- Used a gaussian kernel of size 7*7 because it seemed a good balance between performance (Capturing sufficient local neighborhood information) and computational cost.
- Used interpolation='nearest' because it ensures that each pixel in the neighborhood mask is displayed clearly without any interpolation-induced blurring.

Result

1(c) The initial estimate for the membership values:

- Used an intensity based thresholding (as 33 and 66 percentile) as:
    - Lowest Intensity (<= 33 percentile): CFS
    - Intermediate Intensity (> 33 percentile & <= 66 percentile): GM
    - Highest Intensity (> 66 percentile): WM
- To deal with pixels on the boundary, assigned equal membership i.e. ⅓ ensuring the sum is 1.
- Finally, Normalized the membership values so that for each pixel the sum across all classes equals 1.

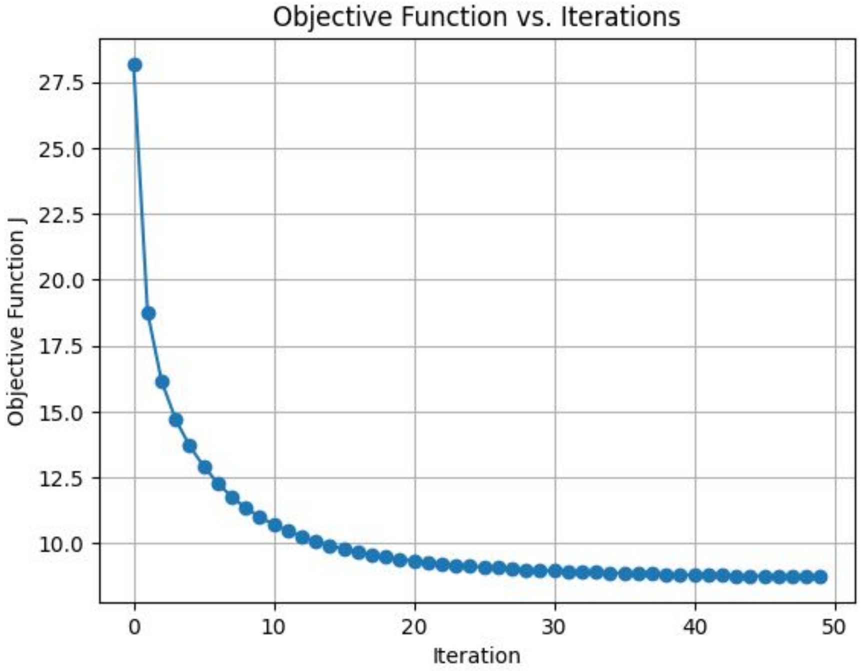
Result

1(d) Initializing the class means

- As MR images exhibit distinct intensity ranges for different tissue types, by computing the weighted average of the intensities based on initial memberships, the resulting class means can capture these inherent intensity differences.
- Therefore, initial class mean c_k for class k is computed by taking a weighted sum of the observed intensities (bias-corrected by the current bias estimate b_n) over all pixels inside the brain.
- Finally, The denominator normalizes the contributions, which ensures that the estimated class mean reflects the average intensity for pixels that predominantly belong to that class.

Result

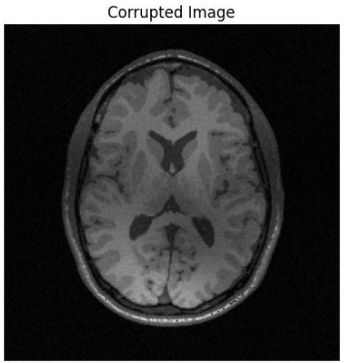
1(e) Values of Objective function at each iteration in modified FCM Algo:

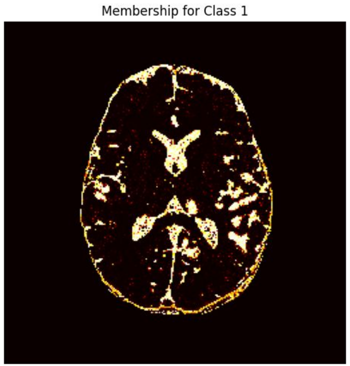


Value of Objective Function at Convergence:
8.7564

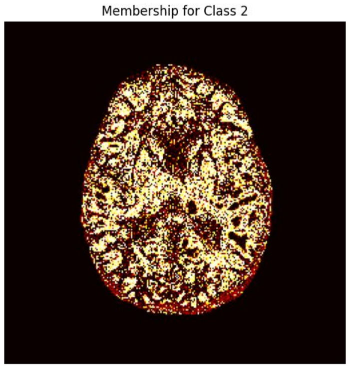
Result

1(f) Showcasing the images:

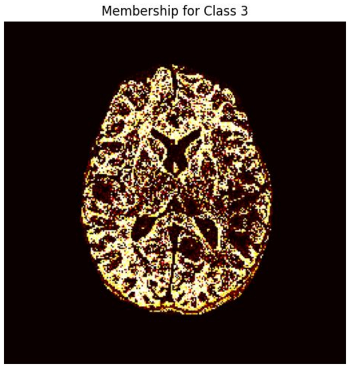




CSF



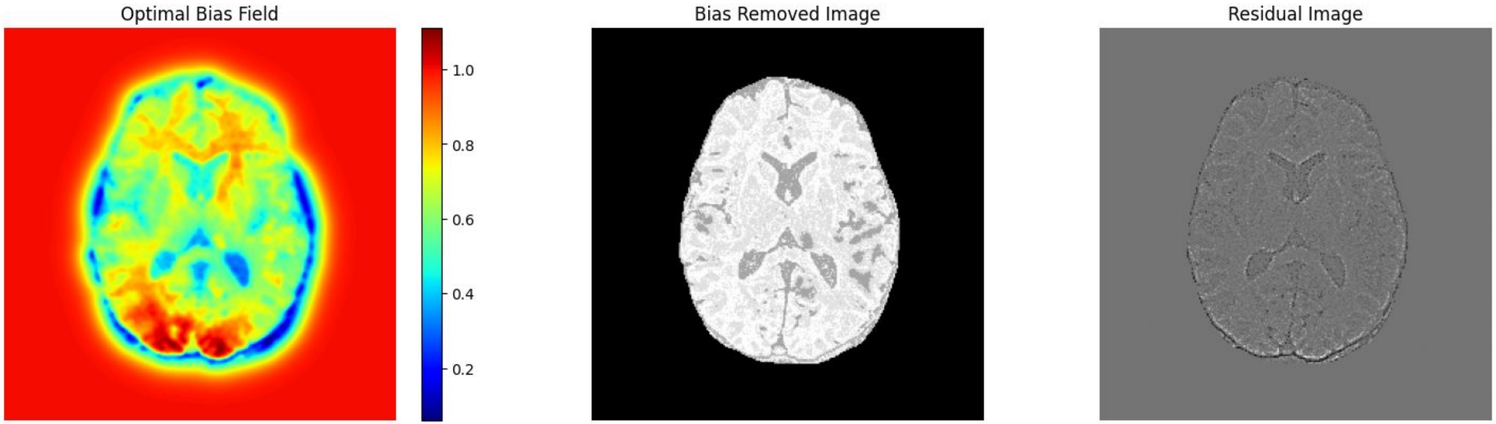
GM



WM

Result

1(f) Showcasing the images:



Result

1(g) Optimal Class Means Estimate:

- Cluster 1 Mean: 0.43084619
- Cluster 2 Mean: 0.58572463
- Cluster 3 Mean: 0.66383283

Result

(2) Whether the formulation discussed in class leads to a unique solution:

- No, It is because if you multiply the bias field by a constant α and simultaneously divide the class means by α, the product $b_n c_k$ (which really matters in the reconstruction) remains unchanged.
- The solution can be made unique by impose a constraint on the bias field. I used a constraint i.e. to force the mean value of the bias field over the brain region to be 1 as:

$$\frac{1}{N}\sum_{n\in\text{brain}} b_n = 1$$

Task 2

Objective

Brain MR Image Segmentation (CFS, GM & WM) using EM optimization relying on GMM and MRF

Description

- Writing code to find optimal values of membership, class means and class standard deviations.
- Writing code for optimal labelling, based on modified ICM algorithm.
- Show the following 5 images in the report (i) Corrupted image provided, (ii) Optimal class-membership image estimates for chosen β, (iii) Optimal label image estimate for chosen β, (iv) Optimal class-membership image estimates β = 0, i.e., NO MRF prior on labels, (v) Optimal label image estimate for β= 0, i.e., NO MRF prior on labels.
- Optimal estimates for the class means for chosen β.

Task 2

Approach

- Load the corrupted MR image and its corresponding binary brain mask from the provided, also restricted it using the mask to use only brain part.
- Defined the neighborhood structure (4-connected) for the brain pixels and Built an adjacency list to later compute the MRF prior.
- Used GMM clustering to initialize labels.
- Created a one-hot (hard) membership matrix from the initial labels, which are refined in the iterative loop.
- For each brain pixel, computed the Gaussian likelihood for each class and normalized to update the membership probabilities w_{nk}.
- Using the current memberships, updated each class's mean and standard deviation using weighted averages and variances.

Task 2

Approach

- Computed the overall (unnormalized) log-posterior of the current label configuration, including both the Gaussian likelihood (for each pixel) and the pairwise MRF penalty.
- In the ICM update loop, for each pixel, evaluated each candidate label (using the Gaussian likelihood and the MRF prior from neighbors) and picked the one with the highest local posterior.
- Compared the overall log-posterior before and after the ICM update. Accepted if the new configuration improves.
- Plotted the required graphs.

Result

(a): Membership Update at each iteration to get optimality:

$$w_{nk} = \frac{\frac{1}{\sigma_k}\exp\left(-\frac{(Y_n-\mu_k)^2}{2\sigma_k^2}\right)}{\sum_{j=1}^{K}\frac{1}{\sigma_j}\exp\left(-\frac{(Y_n-\mu_j)^2}{2\sigma_j^2}\right)}$$

(b): Class Means Update at each iteration to get optimality:

$$\mu_k = \frac{\sum_{n=1}^{N} w_{nk} Y_n}{\sum_{n=1}^{N} w_{nk}}$$

(c): Standard Deviation or Var Update at each iteration to get optimality:

$$\sigma_k^2 = \frac{\sum_{n=1}^{N} w_{nk}(Y_n - \mu_k)^2}{\sum_{n=1}^{N} w_{nk}}$$

Result

(d): Optimal Labelling based on ICM optimization:

Expression for maximizing the posterior probability for labels:

$$\log p(x \mid Y) \propto \sum_{n=1}^{N} \log p(Y_n \mid x_n) + \sum_{(n,m)\in\mathcal{N}} \phi(x_n, x_m)$$

Where the likelihood term for a pixel n with label k is:

$$\log p(Y_n \mid x_n = k) \propto -\frac{1}{2}\log(2\pi\sigma_k^2) - \frac{(Y_n - \mu_k)^2}{2\sigma_k^2}$$

And the prior for a pair (n, m) is:

$$\phi(x_n, x_m) = \begin{cases} 0, & \text{if } x_n = x_m, \\ -\beta, & \text{if } x_n \neq x_m. \end{cases}$$

Result

- The code evaluates both of these quantities i.e. log-likelihood and prior for each pixel and candidate label.
- Then, for each pixel, the label that maximizes the sum, is selected.
- After updating all pixels (synchronously), the code computes the overall log-posterior before and after the update.
- The new label configuration is accepted only if it increases (or does not decrease) the log-posterior.
- It ensures that the modified ICM update will either improve or maintain the overall log-posterior probability.

Result

2(a) Chosen Value of β: 8

- After manually tuning the parameter β, and testing a range of values (e.g., β = 0, 0.5, 1.5, 5, 8, 10, 15), I found that setting β = 7 produced the most realistic segmentation.
- At β = 0 (no spatial smoothing), the segmentation was noisy with abrupt label changes.
- And the larger values tended to over smooth the segmentation, which blurred important tissue boundaries.
- Also, the objective function was very close to 0 for after ICM plot, which indicated a balance where both the likelihood (GMM fit) and the prior (MRF smoothness) are contributing effectively.

Result

2(b) Initial estimate for label image x:

- Used a Gaussian Mixture Model (GMM) Clustering Approach, which ensures that the initial segmentation is aligned with intensity distributions in the image.
- Performed hard assignment by initializing x by assigning each pixel to the class k that maximizes the Gaussian likelihood:

$$x_i = \arg\max_k P(Y_i|\mu_k, \sigma_k) = \arg\max_k \left[-\frac{1}{2}\log(2\pi\sigma_k^2) - \frac{(Y_i - \mu_k)^2}{2\sigma_k^2}\right]$$

- It provided a reasonable starting segmentation without considering spatial smoothness.

Result

2(c) Initial estimates of the Gaussian parameters , i.e., the class means and standard deviations:

- Fitted a Gaussian Mixture Model (GMM) to the observed pixel values of the image given, identifying k distinct clusters.
- Computed the mean of each cluster, ensures μ_k values are centered around the dominant intensities:

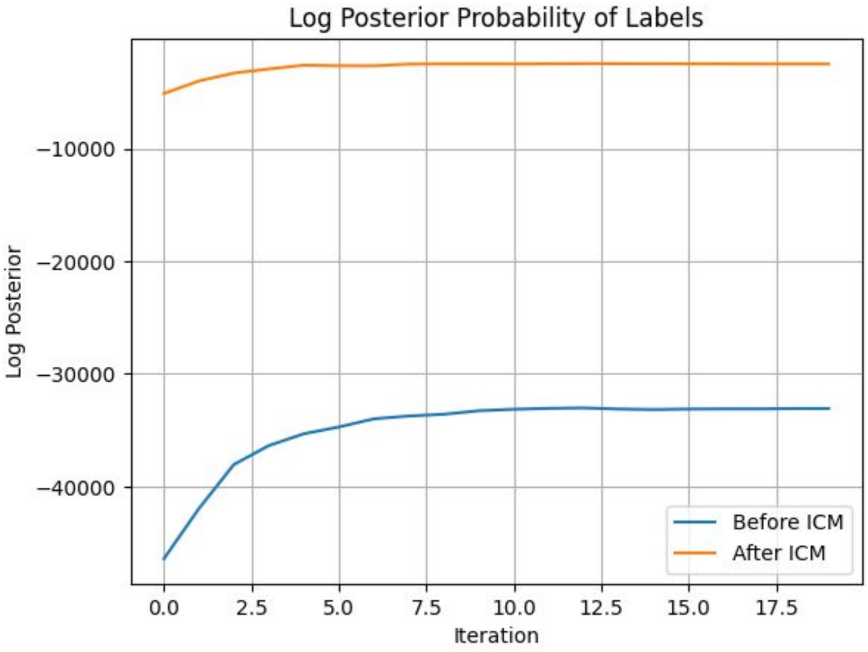
$$\mu_k = \frac{1}{N_k} \sum_{i \in C_k} Y_i$$

- Then computed the standard deviation of pixels in each cluster:

$$\sigma_k = \sqrt{\frac{1}{N_k} \sum_{i \in C_k} (Y_i - \mu_k)^2}$$

Result

2(d) Values of Log Posterior Probability for Labels (before & after ICM update

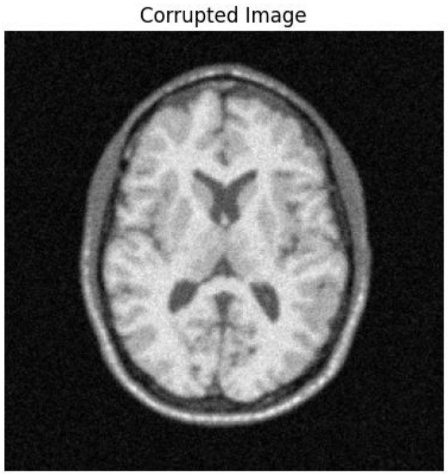


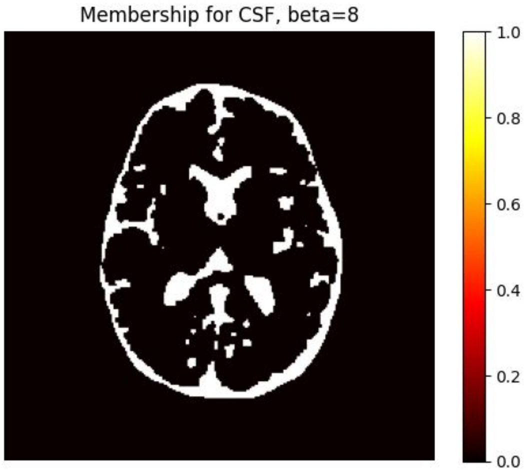
Converged Log Posterior for:

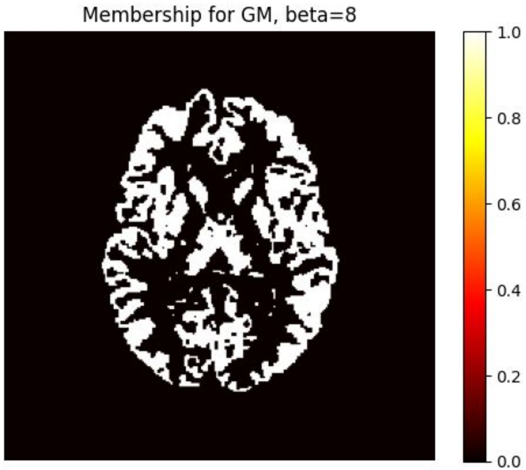
- Before ICM: -33077.25531230283
- After ICM: -2498.7970524290577

Result

2(e) Required Plots (β = 8):

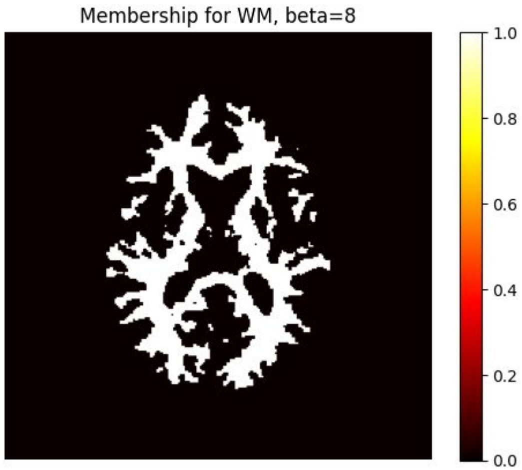


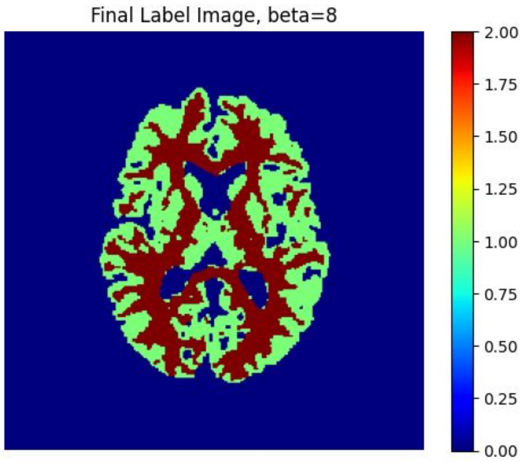




Result

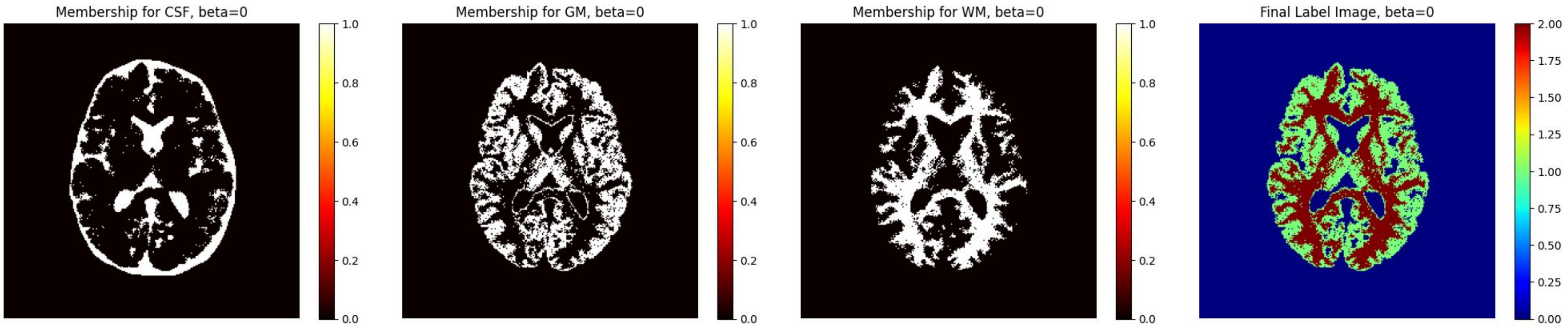
2(e) Required Plots (β = 8):





Result

2(e) Required Plots (β = 0):



Result

2(f) Optimal estimates for the class means for the chosen β = 8:

- Final Class Means:
  - Class 1 Mean: 0.37955903
  - Class 2 Mean: 0.53467993
  - Class 3 Mean: 0.63583703

Thank You!