CS 736

Assignment 4

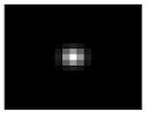
Amey Patil - 160050006 , Aditya Jadhav - 160050010

8th April 2018

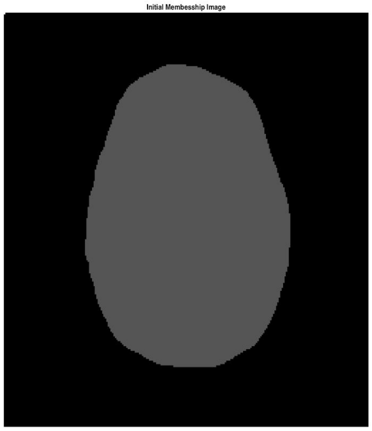
Question 1 - Segmenting Brain MRI image using Fuzzy-C-means

A) The optimal value of the free parameter q = 1.9
The sum of absolute value of Residual Image is the correct measure of optimum value of q. The sum of absolute value of residual image, i.e min {sum(abs(R)) }, this is the measure of the difference between the observed image data with the expected pixel values using class mean, membership and bias field.

B) The neighborhood



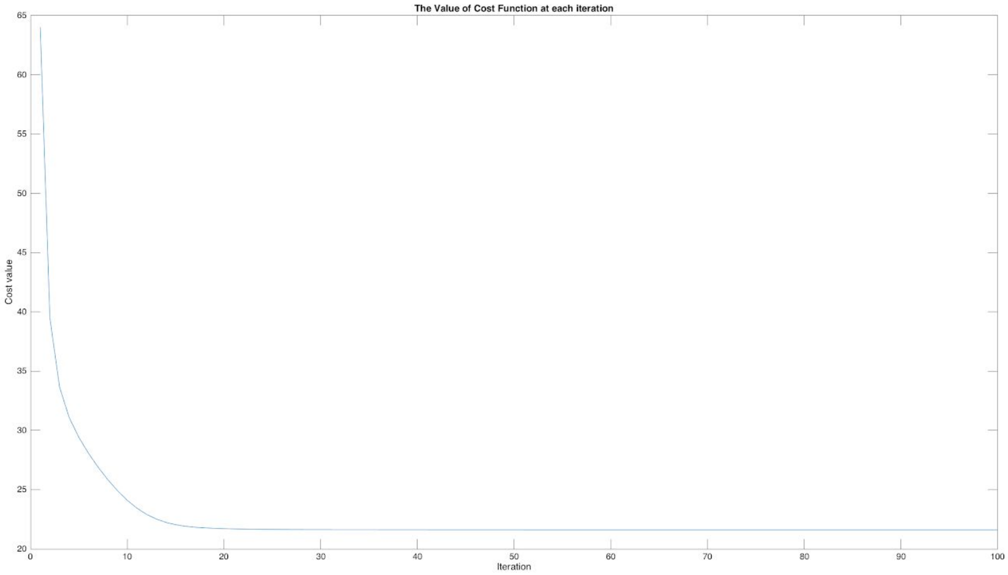
C) Initial estimate of Membership values : I used uniform membership for all pixel values inside brain. I.e U(i,j) = ⅓. This initialization of membership doesn't matters, as in the first step in Optimization we are updating membership values anyways using initial class mean values obtained by kmeans initialization.



D) Initial estimate of Class Mean values : [0.4571 , 0.6361 , 0.2525]
I used kmeans hard classification algorithm to initialize the class mean values for all pixel values in the image.

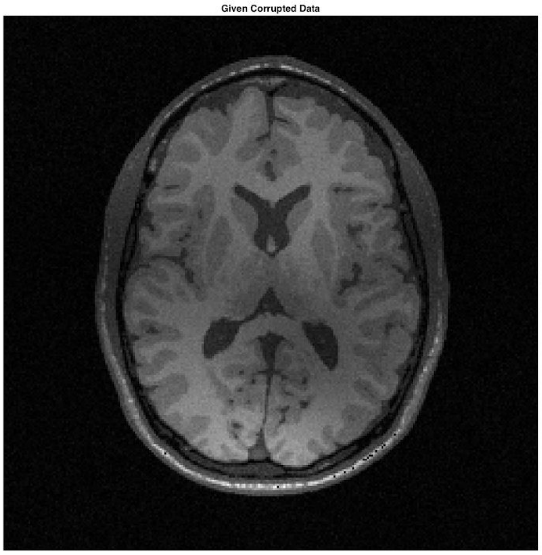
E) The value of objective function at each iteration : The cost value of the objective function is non increasing. Initial Cost = 66.0386, Final Cost value = 22.4135

The graph of Cost at each iteration is as follows



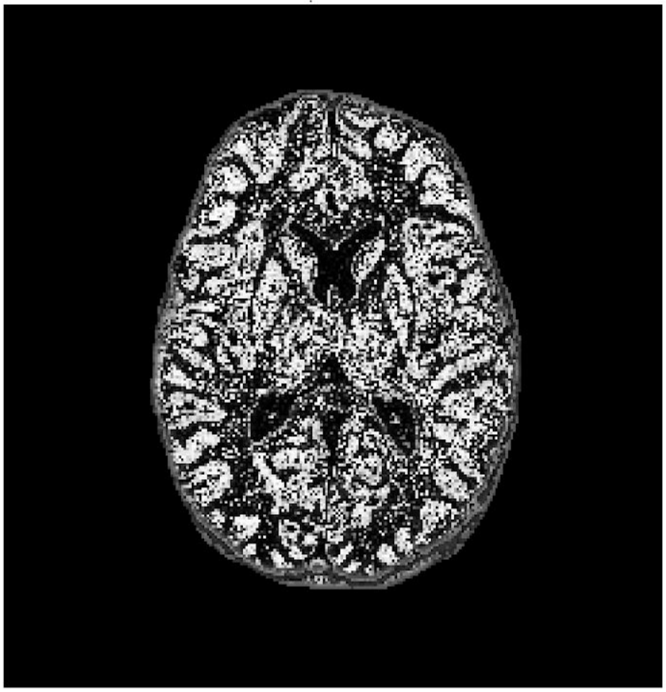
F) Images to be reported

a) Given Corrupted Image provided -

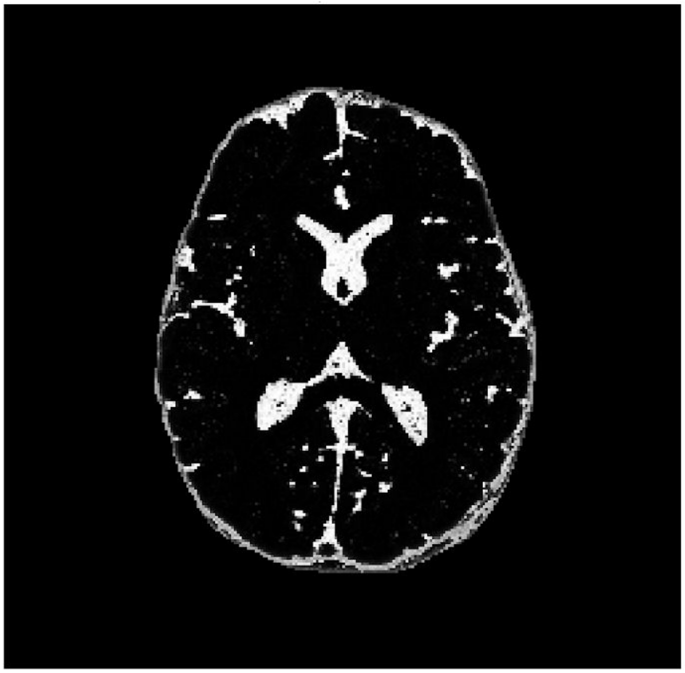


b) Optimal class-membership image estimate -

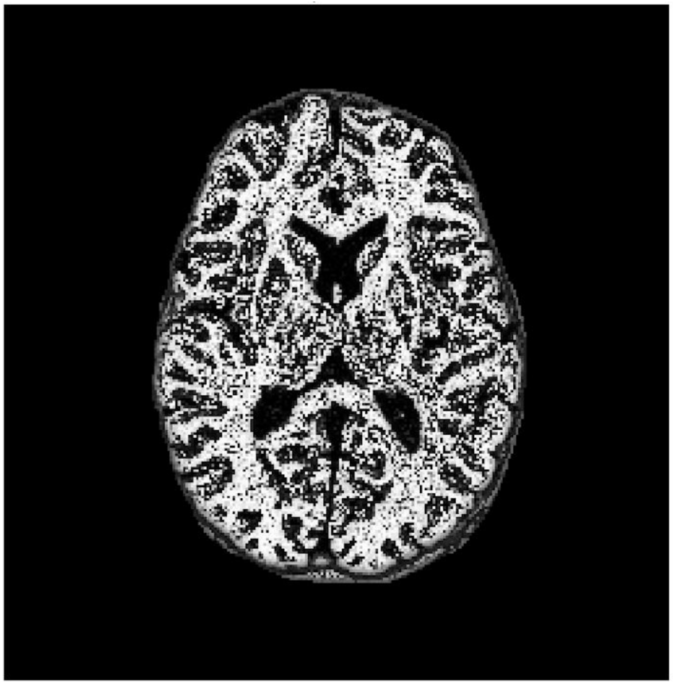
i) Class 1 -



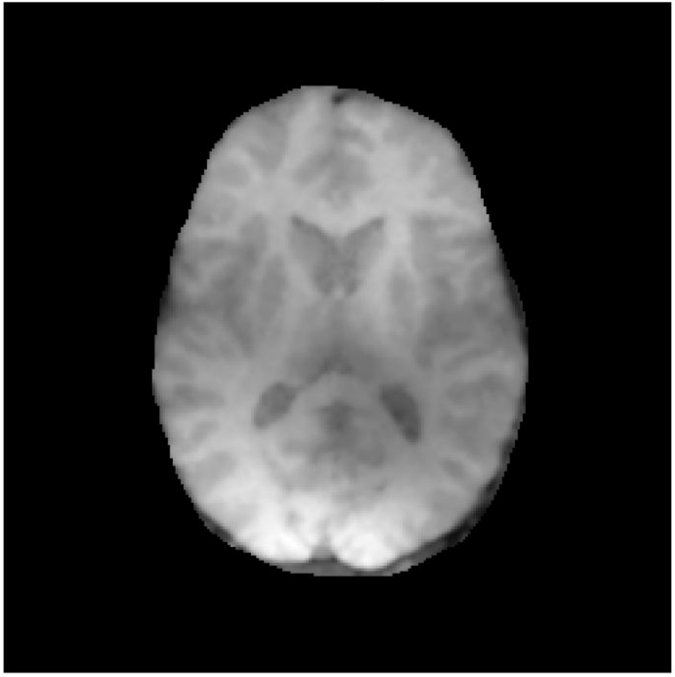
ii) Class 2 -



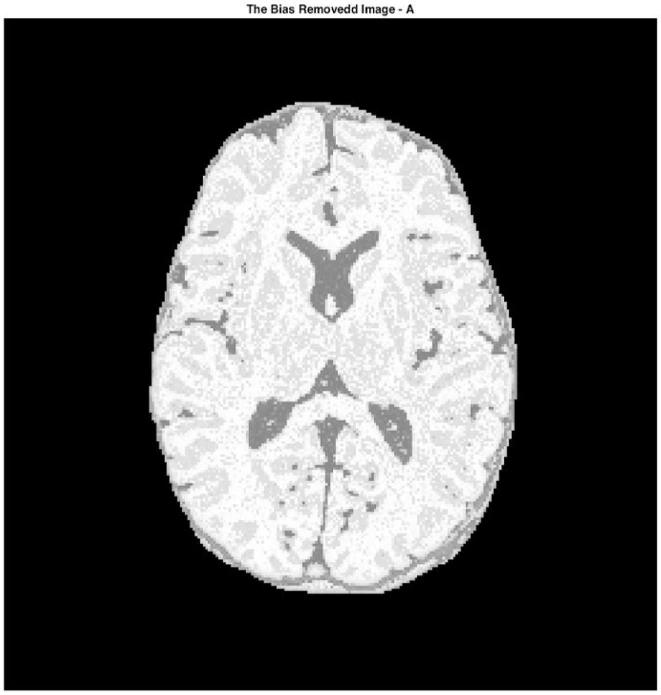
iii) Class 3 -



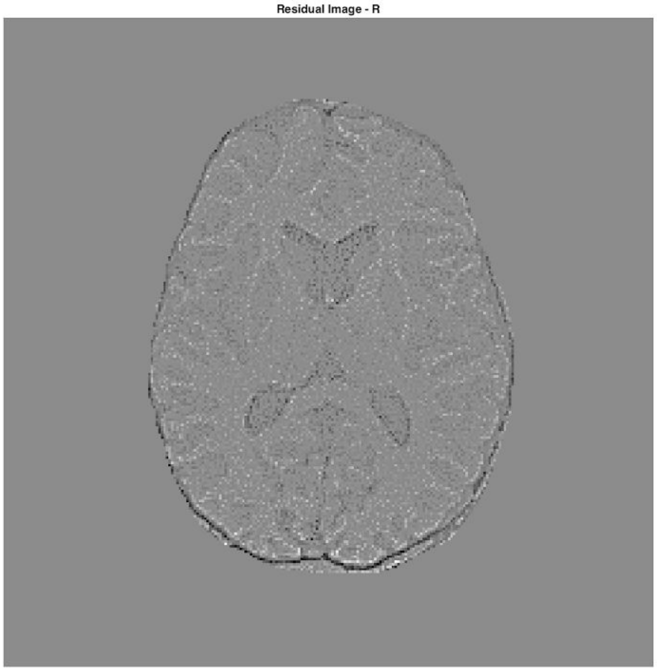
c) Optimal bias field image estimate -



d) Bias removed image (A) -


The Bias Removedd Image - A

e) Residual Image (R) -


Residual Image - R

G) The optimal estimation of Class Means : [0.5544 , 0.4874 , 0.3017]

Question 2 :

To implement the algorithm for segmenting the brain in 3 segments, namely, (i) white matter, (ii) gray matter, and (iii) cerebrospinal fluid, using an 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.

Code to find the optimal value of the memberships : - UpdateGAM.m

Returns New X-MAP and Gamma Values (256 x 256 x 3)

Code to find the optimal value of the class means : - UpdateUk.m

Returns New Mean for Kth Class

Code to find the optimal value of the class standard deviations

Returns New Standard Deviation values for Kth class.

Code to find the optimal labeling, within every iteration :- UpdateGAM.m

Return New X-MAP and Gamma Values

a) **β** is basically a weight/scaling factor assigned to the prior. After some hit-and-trials we found that a smooth and realistic segmentation was achieved for chosen value for **β** = 1.

b) Initial estimate for the label image is done using K-Means algorithm (inbuilt MATLAB function). The motivation behind this is obvious as we want to segment the image within 3 classes, K-Means segmentation is a good estimate for initialization of parameters (Mean, Variance, Image Labels).

c) Initial estimates are -

Mean	=	[0.630468521133553 ; 0.510148456503999 ; 0.293141267964317]
Variance	=	[0.001283087177599,0.006554981800504,0.001708301799664]

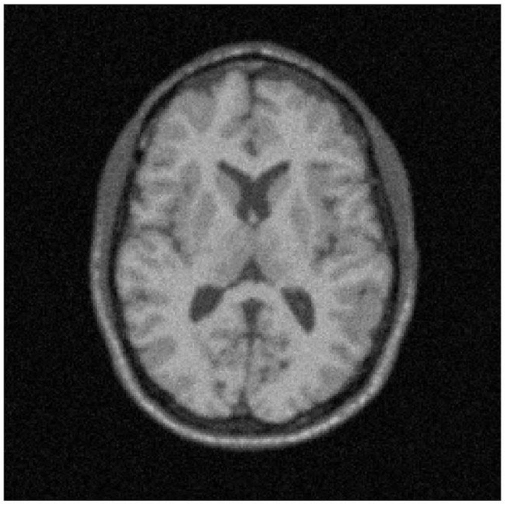
This initialization is done using K-Means (inbuilt MATLAB function). The motivation behind this is obvious as we want to segment the image within 3 classes, K-Means segmentation is a good estimate for initialization of parameters (Mean, Variance, Image Labels).

d) Posterior values within every iteration (Total 100 iterations).

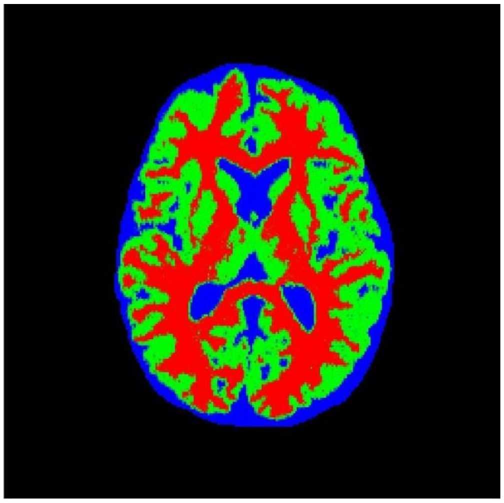
```
64243.2668944582   64647.0281936900   64838.7084582667   64918.9479991437
65023.4743586810   64995.1789957265   65065.2979332403   64983.9791382280
65089.6297941367   65031.4185139167   65123.6870053513   65087.5027898884
65137.0378044830   65088.0291570951   65155.6073671607   65113.6480795708
65162.6811420393   65117.2690739511   65159.1375950460   65104.6073235819
65157.9139742969   65112.1214017754   65165.1488533281   65113.9516340149
65166.5333821631   65114.1521573226   65170.7068142253   65119.1545386871
65170.8656453125   65119.4215494272   65171.0342282257   65119.5072813306
65171.0745719648   65119.5255011093   65171.0826809379   65119.5290606636
65171.0842424200   65119.5297408693   65171.0845396858   65119.5298701098
65171.0845961212   65119.5298946368   65171.0846068311   65119.5298992915
65171.0846088638   65119.5299001748   65171.0846092494   65119.5299003427
65171.0846093230   65119.5299003744   65171.0846093368   65119.5299003805
65171.0846093394   65119.5299003818   65171.0846093399   65119.5299003819
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
65171.0846093400   65119.5299003820   65171.0846093400   65119.5299003820
```

We can clearly see that the Log Likelihood values are increasing within every iteration.

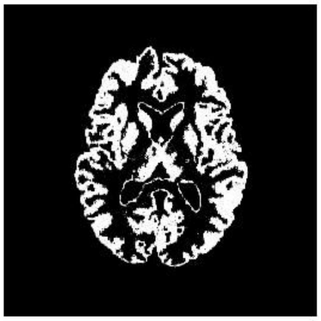
e) Showing the following 5 images -



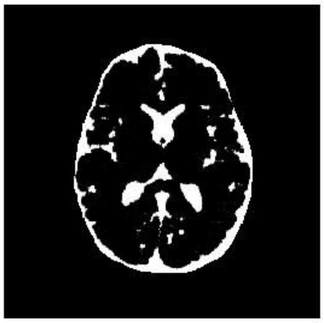
(Corrupted/Given Image)



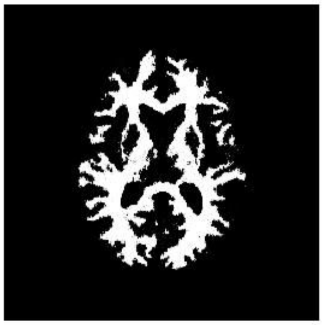
(Class Membership Image)



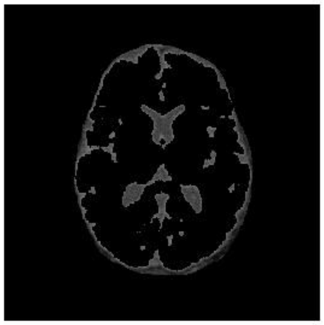
(Class 1)



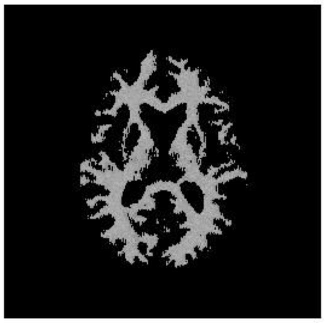
(Class 2)



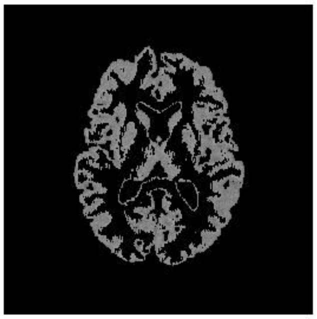
(Class 3)



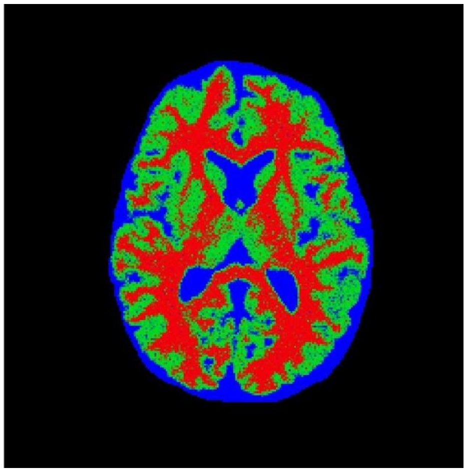
(Label Image 1)



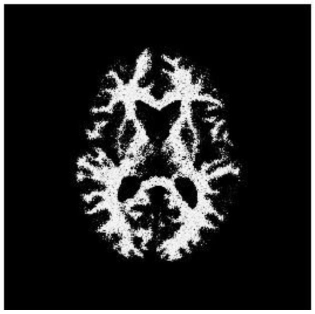
(Label Image 2)



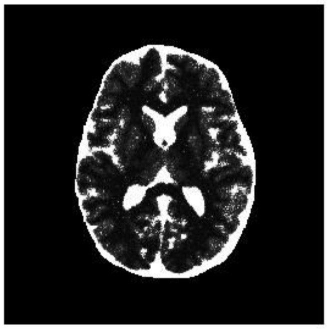
(Label Image 3)



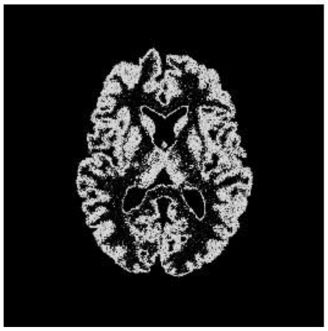
(Class Membership Image | **β** = 0)



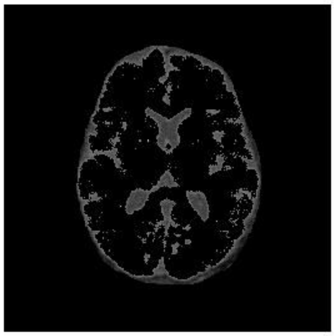
(Class 1 | | **β** = 0)



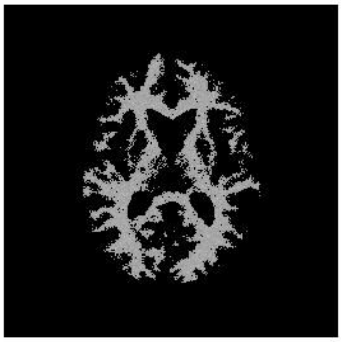
Class 1 | | **β** = 0)



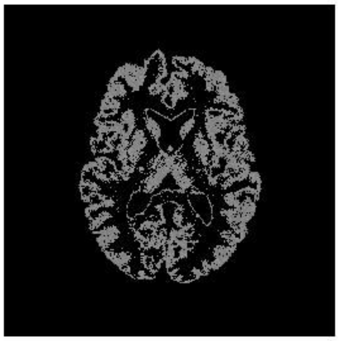
(Class 1 | | **β** = 0)



(Label Image 1 | **β** = 0)



(Label Image 2 | **β** = 0)



(Label Image 3 | **β** = 0)

f) Optimal Means = [0.633866935754221 , 0.318115134329006 , 0.525392182873736]