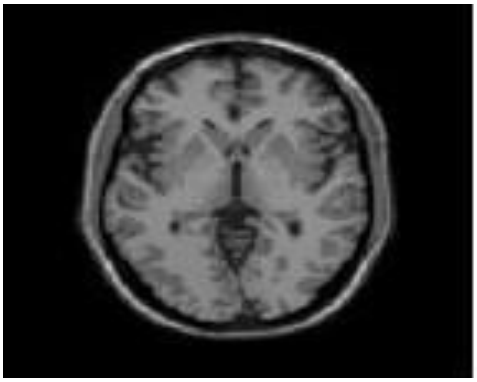
Task 4

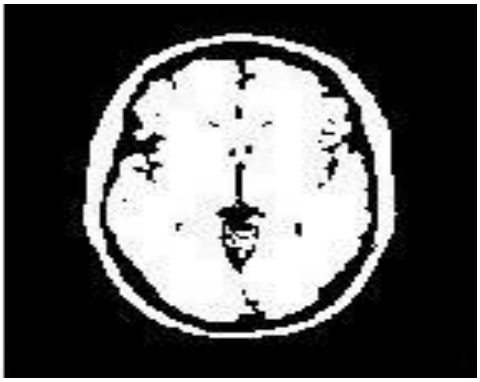
1) A-Optimal Thresolding

function [IO] = Optimal_Thresh(Imj,n)

- This is the main function which is used in Global and local Thresholding as we input the image either colored or grayscale
- We assume the background as the corners of the image and get their mean
- And the rest of the image is the object then get their mean
- And get the threshold which is the average of the 2 means
- Then update the mean of the object as image intensity smaller than the average of the 2 means
- And the updated mean of background as image intensity is greater than the average of the 2 means
- Then calculate the average of the 2 updated means for n iterations
- After no. of iterations make a new zero matrix in which image intensity greater than the final average will equal 255 which is the background and the object will appear in the zeros

For Grayscale image:





For RGB image:

function [Inew] = Optimal_RGB(I)

- We call the main function which is Optimal_Thresh for every matrix in the third dimension which means:
- We take the R-matrix in the image and apply the thresholding and put it on output image as an R-output_image
- The same we do with G-matrix and B-matrix
- And the 3 output images we put it in (Inew) as 3D image like the original one

For Local thresholding :

- We divided the image into subimages
- Then make the same algorithm with every subimage
- And put the output together.

For Grayscale image:

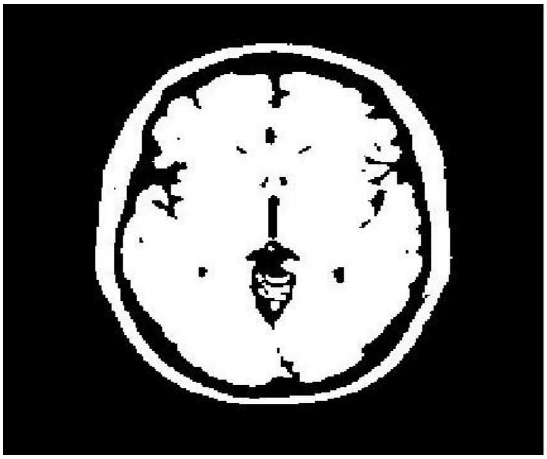
For RGB image:

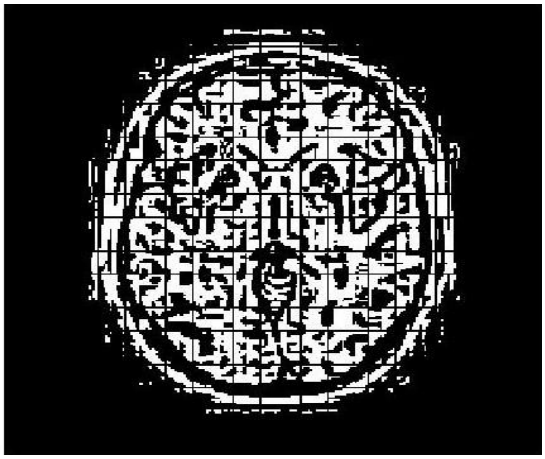
- We did the same with local thresholding as we explained in RGB image of global
- i.e. We took R-matrix and G-matrix and B-Matrix and applied thresholding for every subimage

B- Optimal Thresholding

function [IOtsu] = Otsu(I)

- Main function for global threshold, first we get the histogram for the image which is used as the probability
- Then we divide the histogram into two classes by taking each intensity value in the histogram and stop to calculate the probability, mean of the class before and after this threshold and variance between these two classes to find the maximum variance at certain threshold which is taken as the threshold for the image
- For local thresholding we make the same algorithm for every subimage

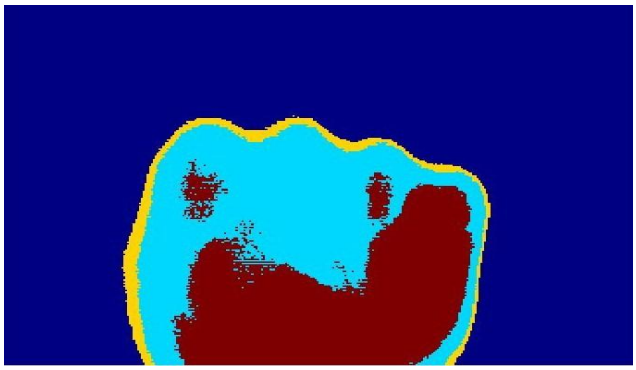




2)A- k-means :

function [Im] = KMeans(Im,n)

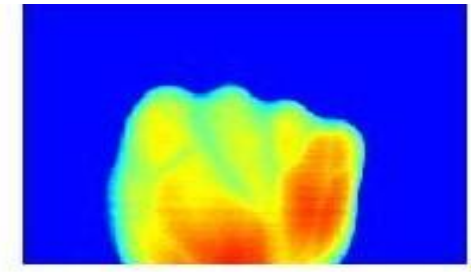
- By taking number of clusters n and choosing n random values which will be the means
- By subtracting the intensity values of the image by the mean of each cluster and finding the minimum value of each pixel to the cluster
- Assigning it to a matrix of this cluster with the same position until assigning all intensity values to every cluster
- Then recalculate the mean of the intensities of the whole cluster again for each cluster
- Which will be the new mean value till it does not change

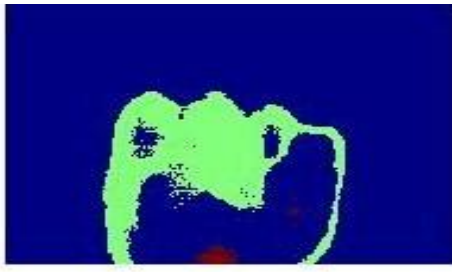


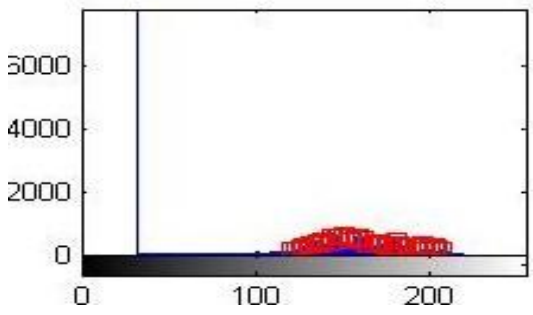
B- Agglomerative :

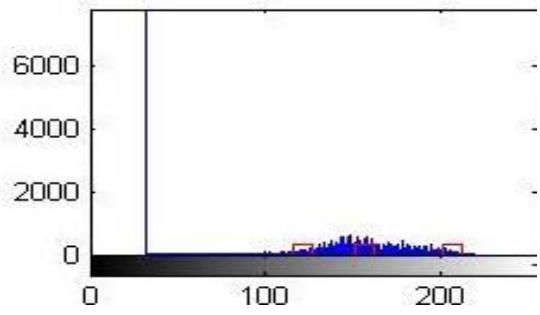
function [Im] = agglomerative(Imj)

- Get the histogram of the image and let all the intensity values are the peaks which will represent the clusters
- Calculate the minimum difference of intensity values between clusters and merge the clusters nearest to each other to become small number of clusters remaining and not close to each other
- Assign index of the same cluster to show the image in number of clusters only









C- Mean Shift for RGB image

function [Im] = agglomerative(Imj)

- By taking R-matrix and G and B and taking their value to make 3D histogram but in 17 bins to make the voting larger
- Get random numbers for R,G,B for number of clusters
- Take a window around this value
- To calculate of center of gravity at each point .. calculate the sum of values multiplied by the index window .. divide it by the length of these values
- Therefore the new value which is the centre of gravity is the new value in which we will take a window until the value does not change