CS590 – MEDICAL IMAGING PROJECT

Dept. of Computer Science

Bishop's University

Sherbrooke, Canada

Topic- Cortical Thickness Challenge

Submitted By:

Ishan Gulati (002295885)

Aayushi Pachorkar(002287964)

Professor:

Dr. Russell Butler

OBJECTIVE:

The goal is to develop an algorithm for estimating cortical thickness map from a raw T1-weighted image and segmenting gray and white matter.

INTRODUCTION:

- **Cortical thickness map** is the thickness of the gray matter of the human cortex brain at every point. It is defined as the distance between the white matter surface and the pial surface. It is as 3D distance measured from the white--gray matter boundary in the tissue classified brain volume to the cortical surface (gray--CSF boundary). The thickness of the cortex is used for identifying affected brain regions by disease for assessing treatment and studying brain development and ageing.
- Cortical thickness measures the width of gray matter of the human cortex and can be calculated from T1-weighted magnetic resonance images (MRI).
- To ensure the accuracy of cortical thickness measures, we compute them in the 3D image volume at a voxel level using gray matter segmentation of the image rather than using the vertices in the surface mesh.

Brain Extraction

Brain extraction is a critical preprocessing step in the analysis of neuroimaging studies conducted with magnetic resonance imaging (MRI) and influences the accuracy of downstream analyses.

Skull Stripping of MR Brain Images

Skull stripping is designed to eliminate non-brain tissues from MR brain images for many clinical applications and analyses, its accuracy and speed are considered as the key factors in the brain image segmentation and analysis. The accurate and automated skull stripping methods help to improve the speed and accuracy of prognostic and diagnostic procedures in medical applications.

- **Mathematical morphology-based method for Skull Stripping**

  these methods use the morphological erosion and dilation operations to separate the skull from the brain region. It requires a combination of thresholding and edge detection methods to find the initial ROI (region of interest). It consists of histogram-based thresholding and morphological operations. Based on the brain anatomical knowledge, it discriminates between the desired and undesired structures. This method is implemented using a sequence of conventional and novel morphological operations, using 2D and 3D operations

Libraries, Packages and Functions used :

- **NumPy as np**
- **from scipy import ndimage as ndi**
- **matplotlib.pyplot as plt**
- **import nibabel as nib**
- **from nilearn import plotting**
- **Import pylab as plt**
- **from skimage.util import montage**
- **from skimage.transform import rotate**
- **from nilearn.masking import apply_mask**
- **from skimage import morphology**
- **from skimage.segmentation import find_boundaries**
- **from skimage.segmentation import mark_boundaries**
- **from math import sqrt**
- **distance_transform_etd from scipy.ndimage**

STEPS PERFORMED FOR THE PROJECT :

- Image Loading
- Skull Stripping
- White Matter Extraction
- Gray Matter Extraction
- White Matter and Gray Matter Boundaries
- Cortical Thickness from White Matter boundary to Pial Surface

(We have implemented 2 different approach- distance transform etd and K-d Tree(as suggested by Professor Russell))

IMPLEMENTATION:

- Here, we have tried 2 implementations in this project - **K-d tree Algorithm** and **distance_transform_etd from scipy.ndimage.**

1. K-d Tree Algorithm :

We have calculated distances from every grey matter voxel to nearest white boundary point and updated the value of every voxel to that distance which we are getting from K-d Tree. Using kdtree.hpp ,kdtree.cpp libraries we have created a tree of white matter boundary points and iterating every grey matter voxel. For every grey matter voxel we are checking for the nearest white boundary point and getting the minimum distance and then we will assign that distance value to that point.

2. distance_transform_etd from scipy.ndimage

We have used this approach for calculating exact Euclidean distance between the pixels and is used to get the thickness of gray matter. Distance is calculated from every grey matter voxel to nearest white boundary point and updated the value of every voxel to that distance which we are getting from distance_transform_etd.

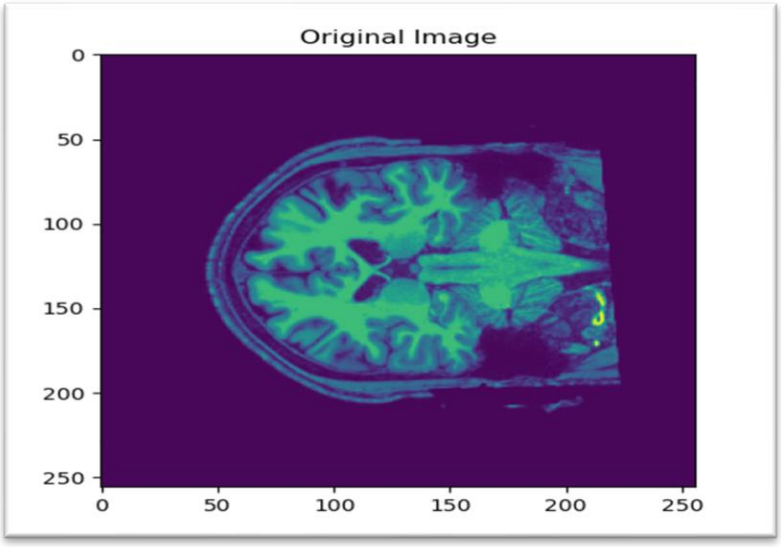
Output Conditions

- ***If gray matter point is closest to the white matter then its value is 1 or 1.414 depending upon the point if it is in a straight line or diagonal respectively.***
- ***If a gray matter point is far from white matter point then its value will increase like we have seen in the output image eg 2 or 4 or greater than 4.***

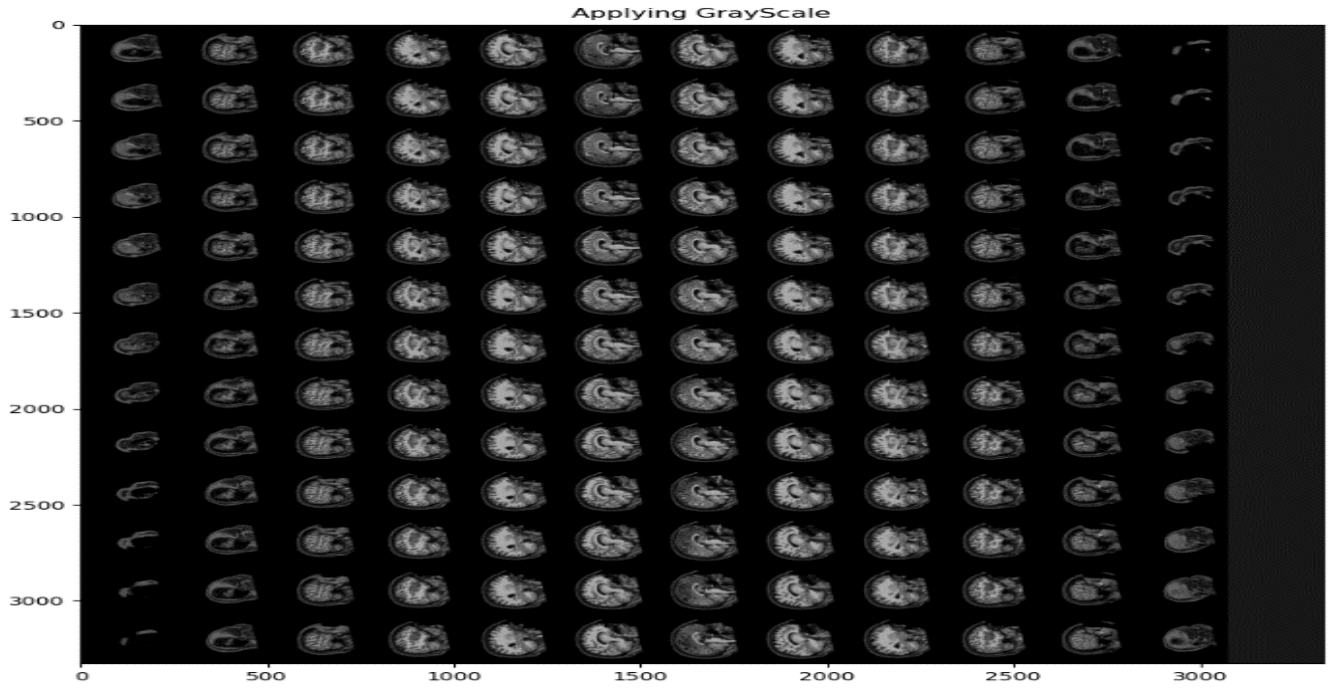
SCREENSHOTS- For the implementation mentioned above

Original Image –

This is an Original Image of brain on which ***cortical thickness mapping and segmentation of Gray and White Matter has to be performed.***

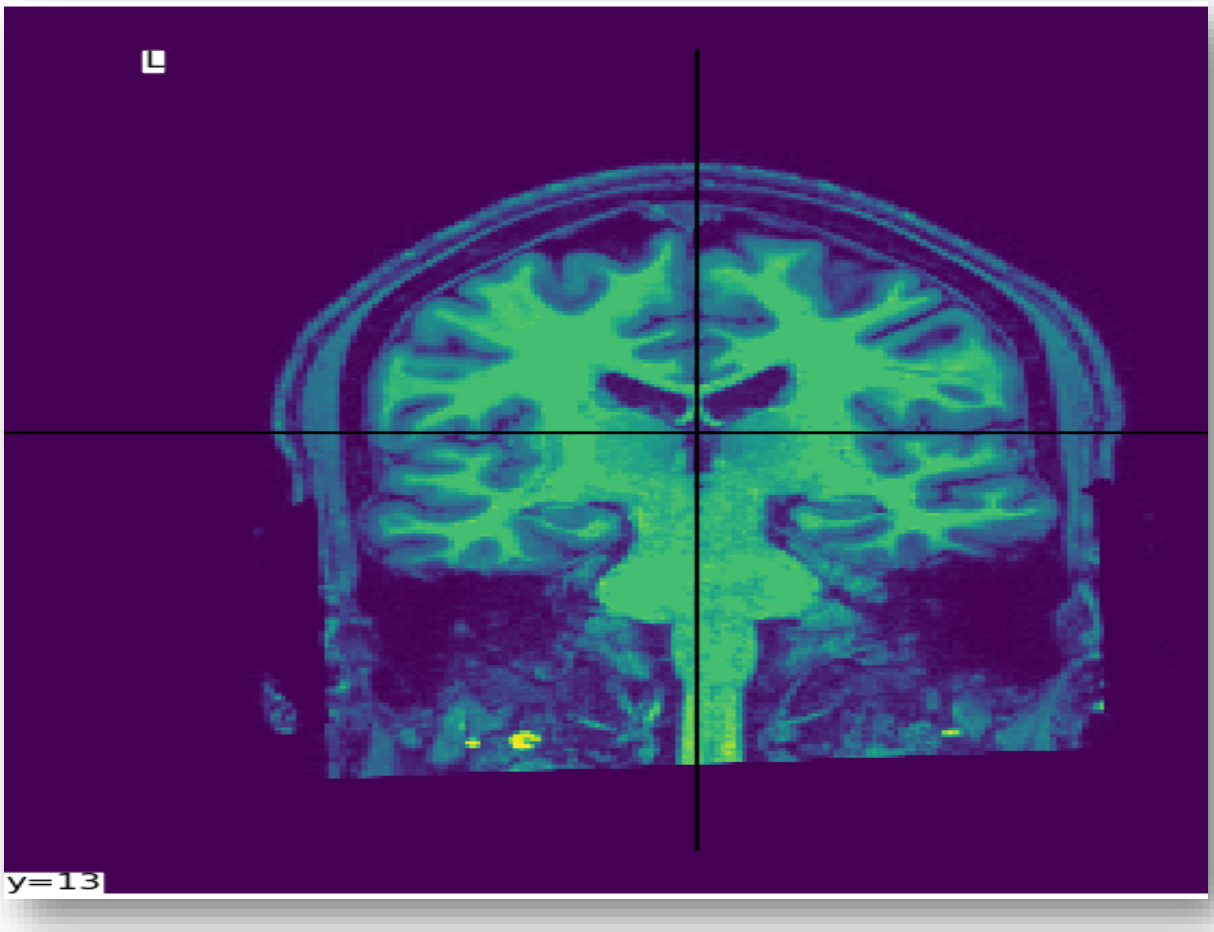


Applying Gray Scale to the plot data from above-

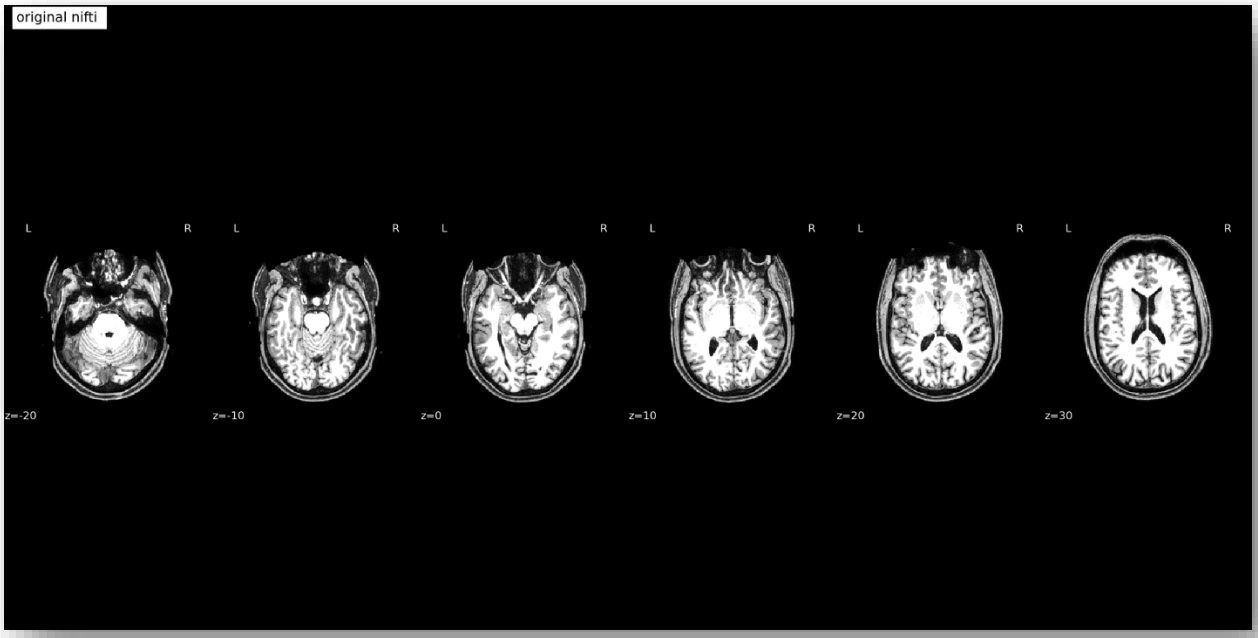


Plotting Mean of Subject-02

plotting.plot_img(mean_subject02, bg_img=mean_subject02)



Nifti Images – Original and Mean



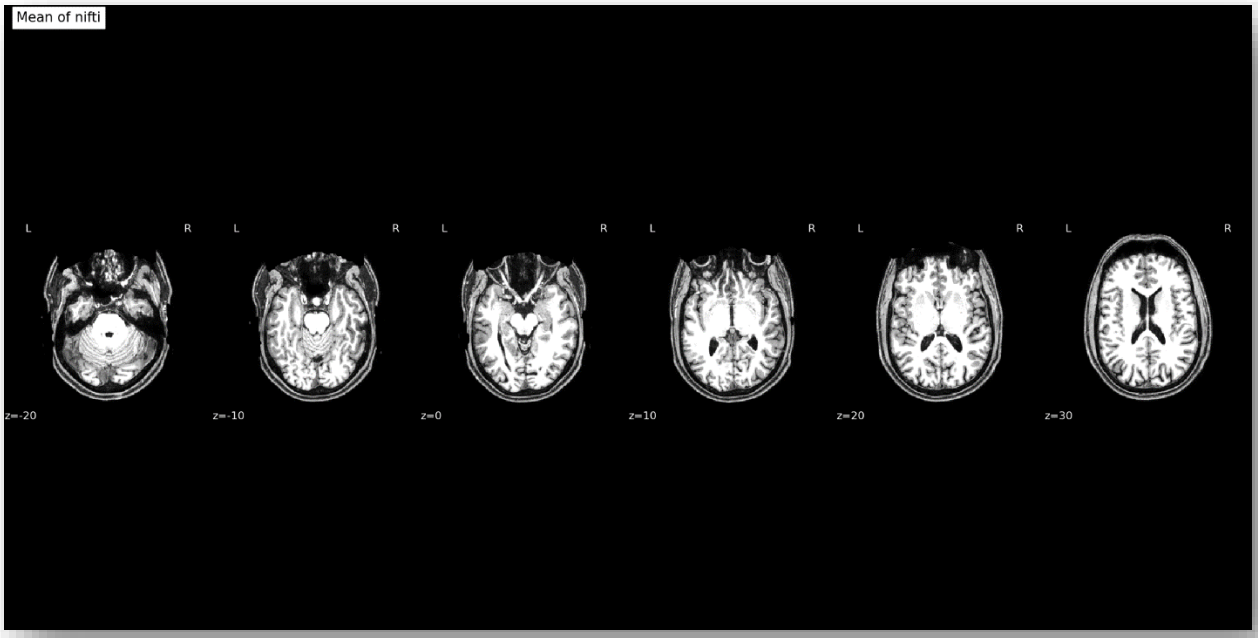
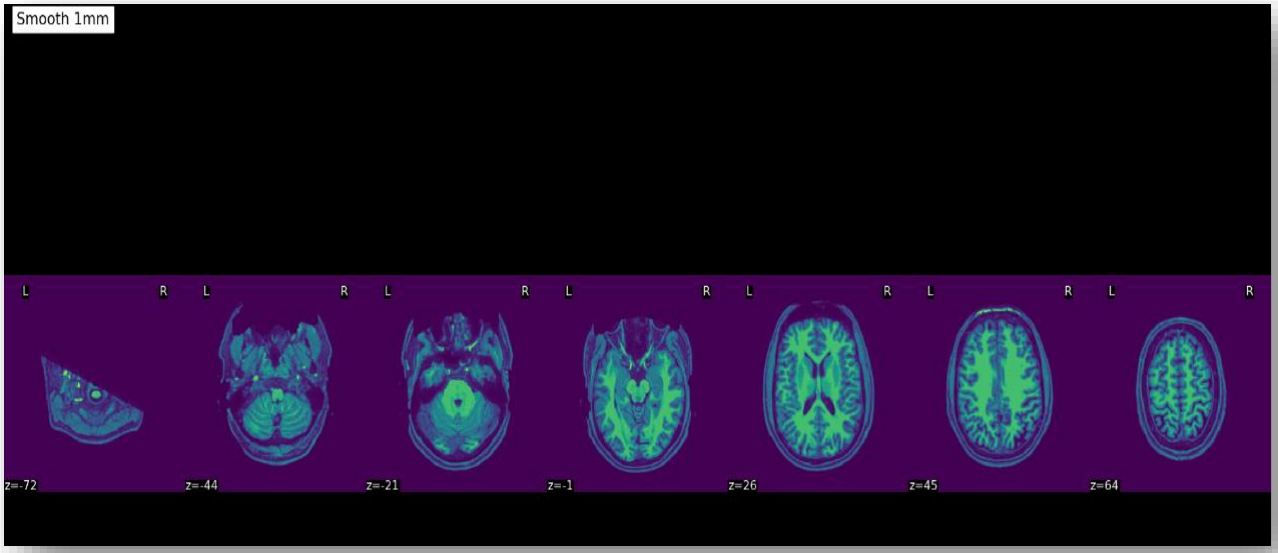
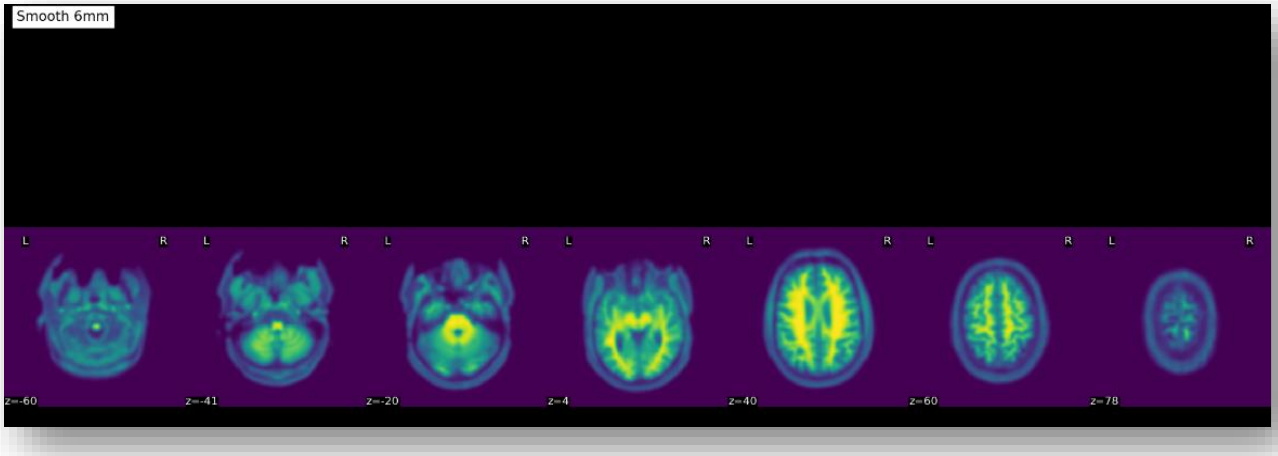
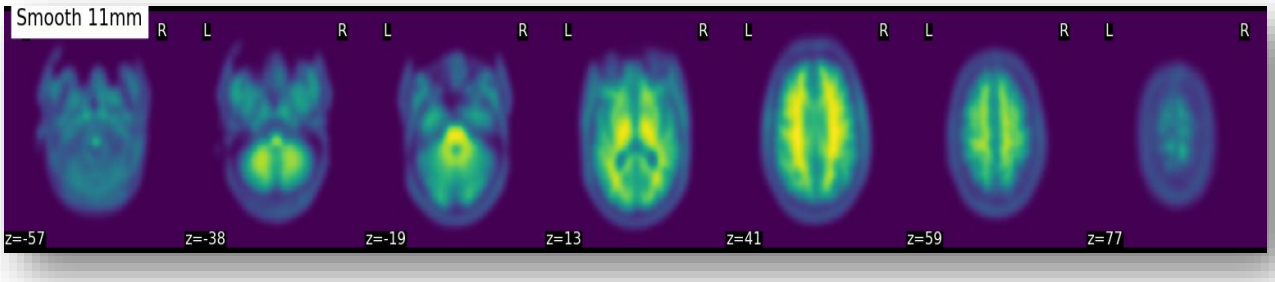


Image Smoothing-

Here, we are smoothing the image by decreasing the amount of noise :

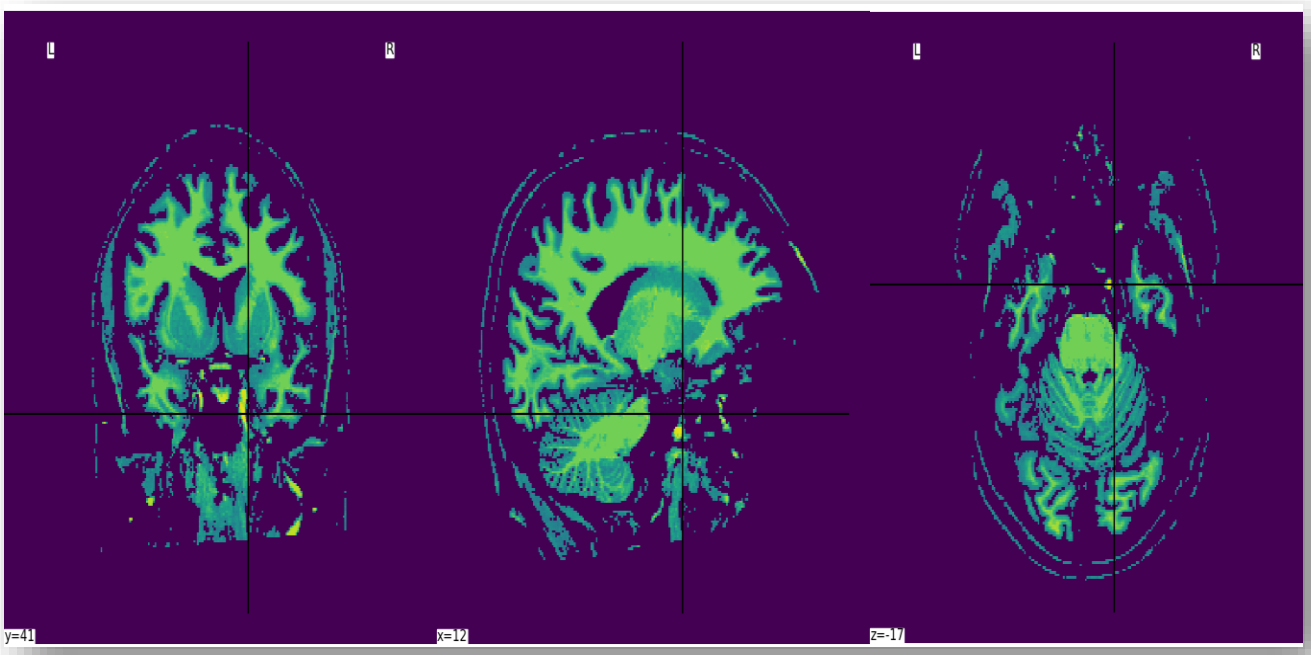




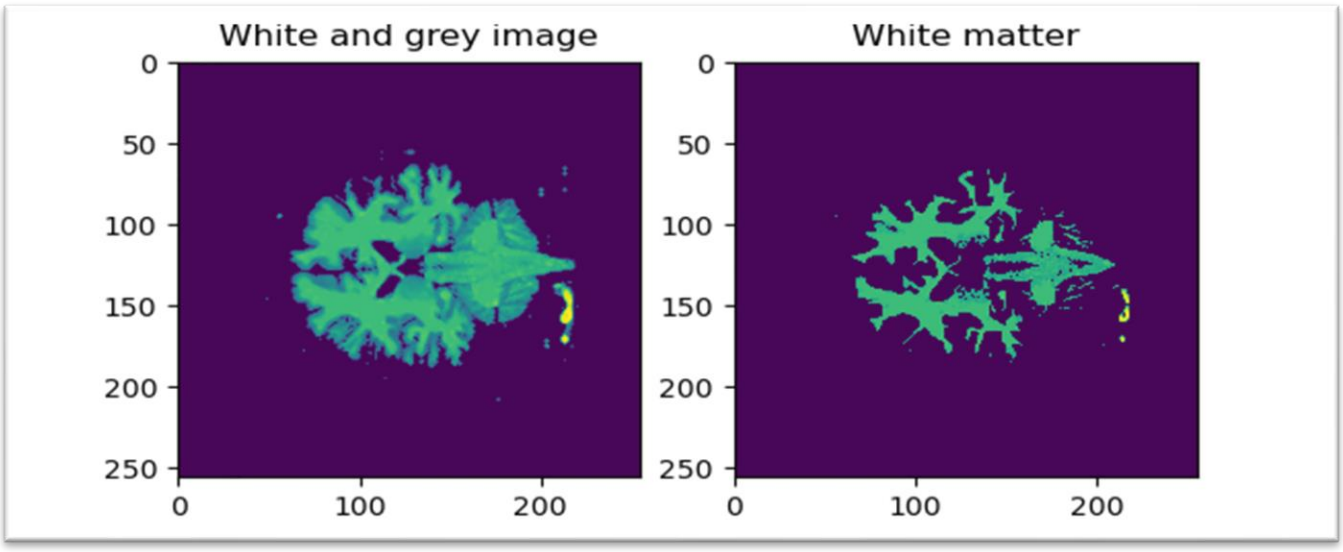


Thresholding Image-

Here, We are thresholding the image making segments based on image intensities meaning voxels which have intensities greater than this value will be kept. The given value should be within the range of minimum and maximum intensity of the input image. If string, it should finish with percent sign e.g "80%" and we threshold based on the score obtained using this percentile on the image data. The voxels which have intensities greater than this score will be kept.

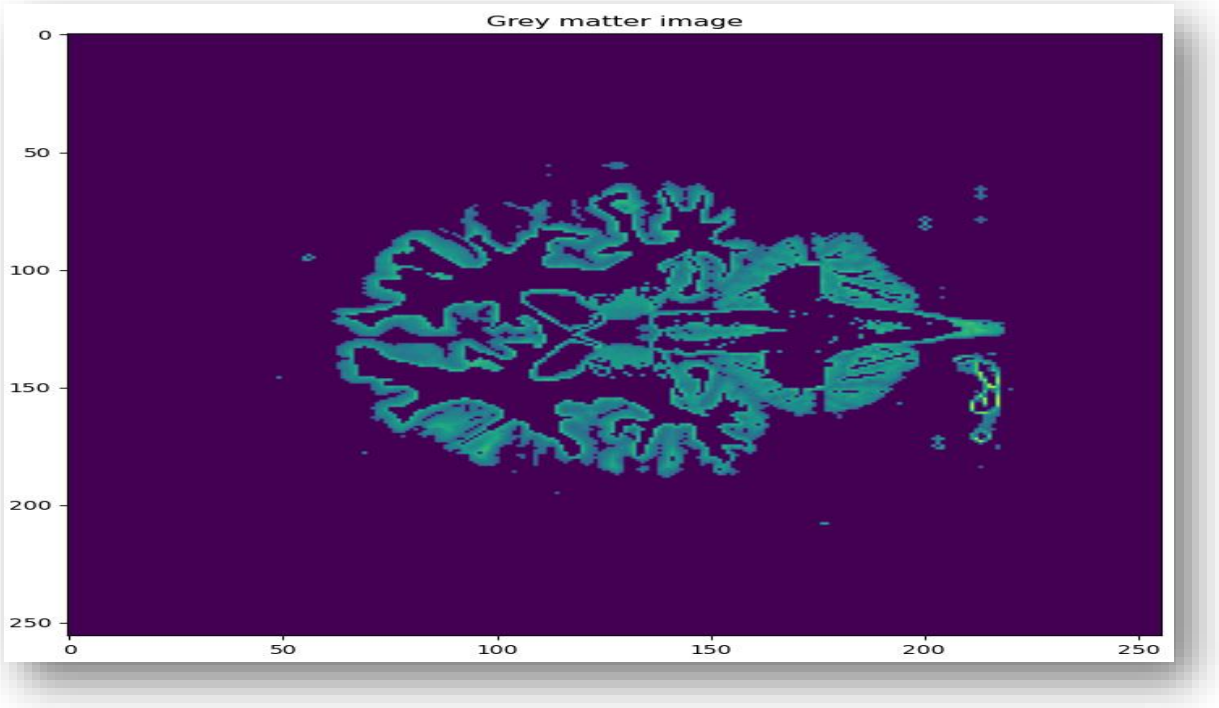


Plot for White and Gray Matter , White Matter -



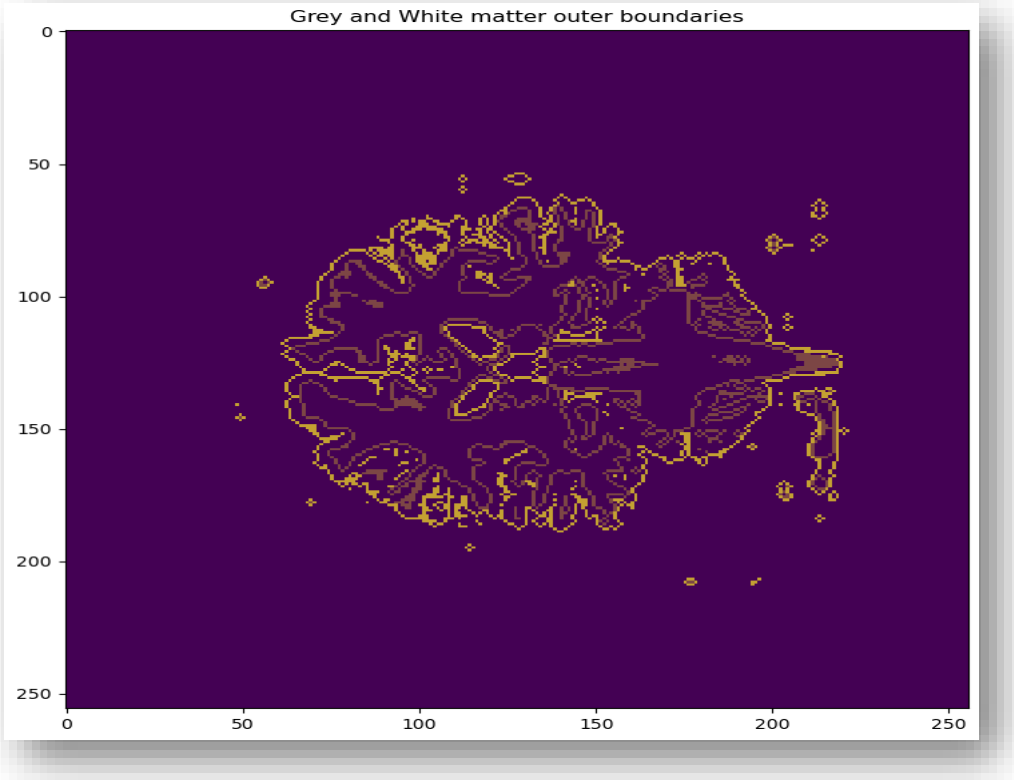
Gray Matter Image -

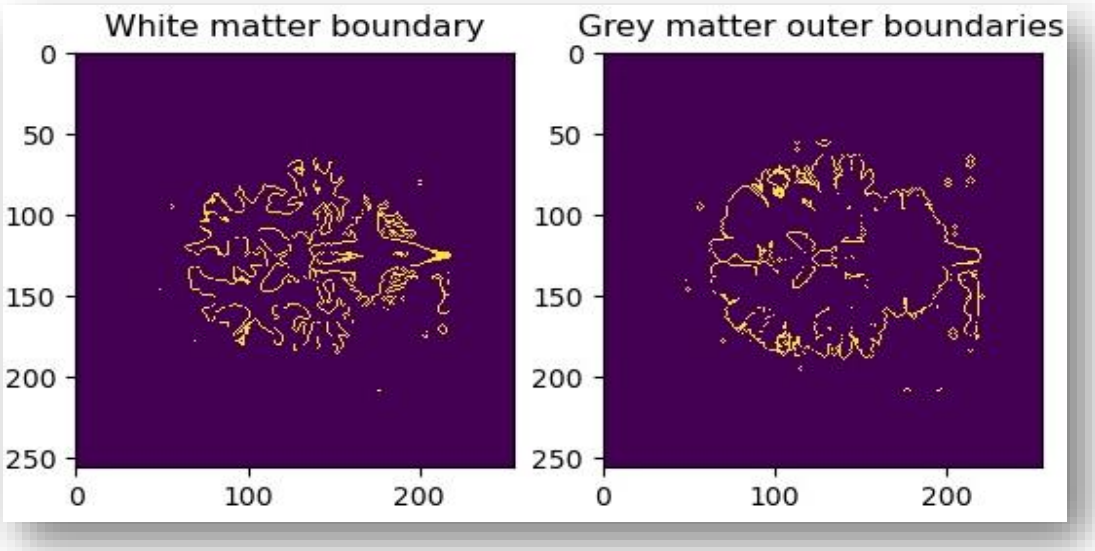
By subtracting both the images(White and Gray Image- White Matter) we have got Gray Matter image as output. We have also implemented dilation on brain segmented image then subtracted it.



Gray and white matter outer boundaries -

These are the outer boundaries for gray and white matter

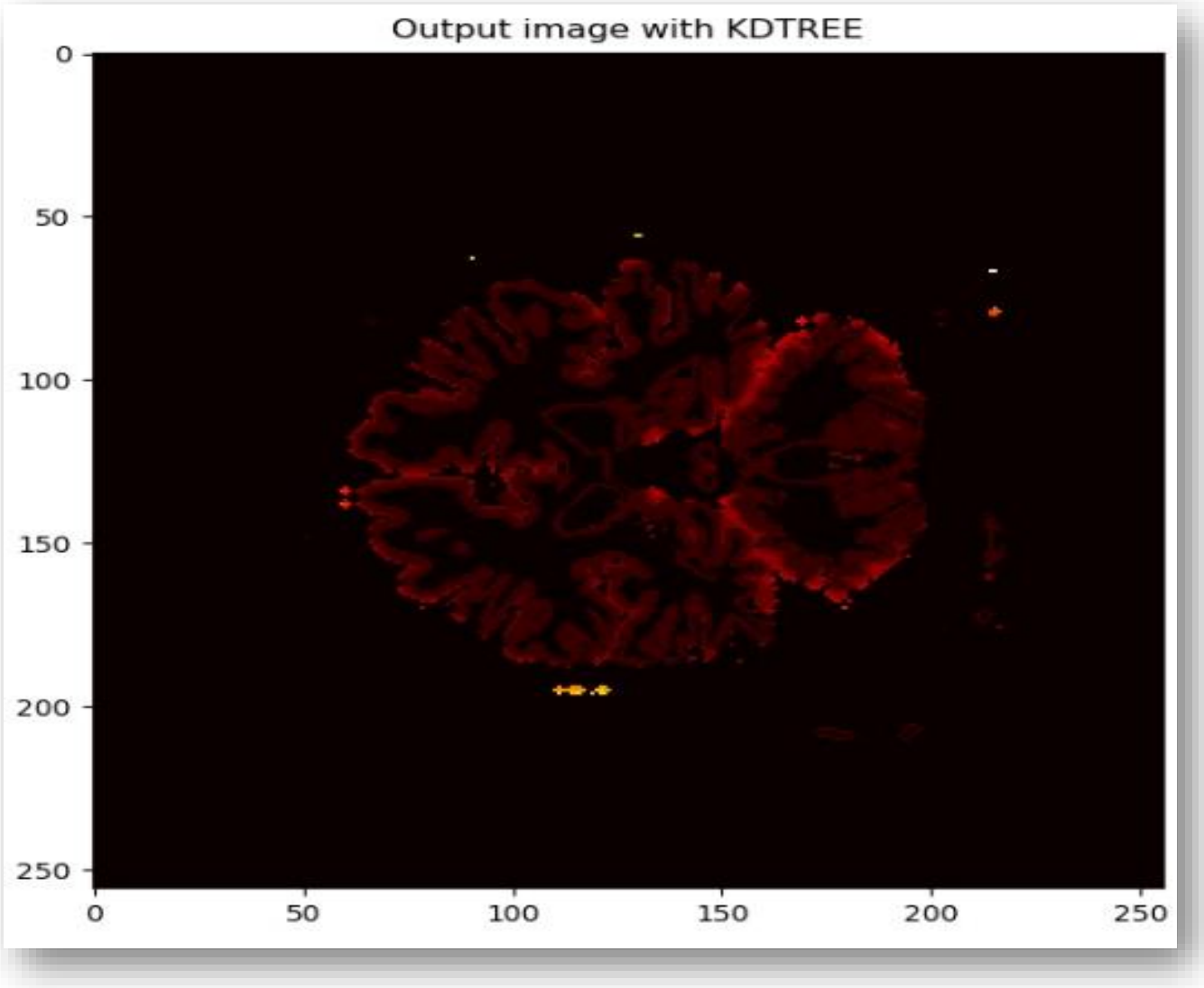




FINAL OUTPUT:

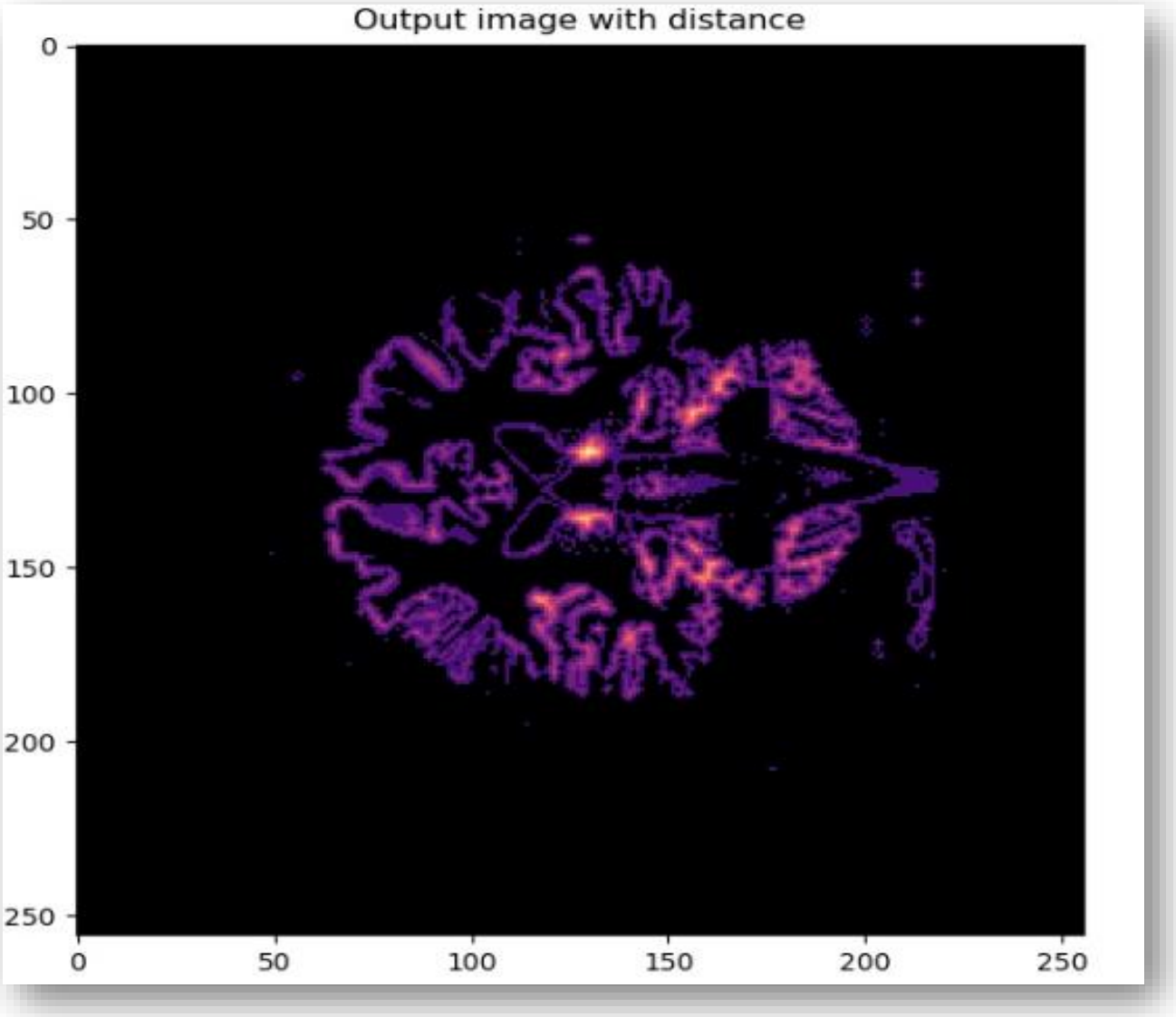
Approach 1- Final Output Image with K-d Tree

By using colourmap as hot. We have calculated distances from every grey matter voxel to nearest white boundary point and updated the value of every voxel to that distance which we are getting from KD Tree. Using Sklearn.neighbors.KDTree libraries we have created a tree of white matter boundary points and iterating every grey matter voxel. For every grey matter voxel we are checking for the nearest white boundary point and getting the minimum distance and then we will assign that distance value to that point.



Approach 2- Output Image with distance_transform_etd from scipy.ndimage

Here we are getting distance value which is 4.24, this is for the pixel(129,117,110) coordinates.



CONCLUSION:

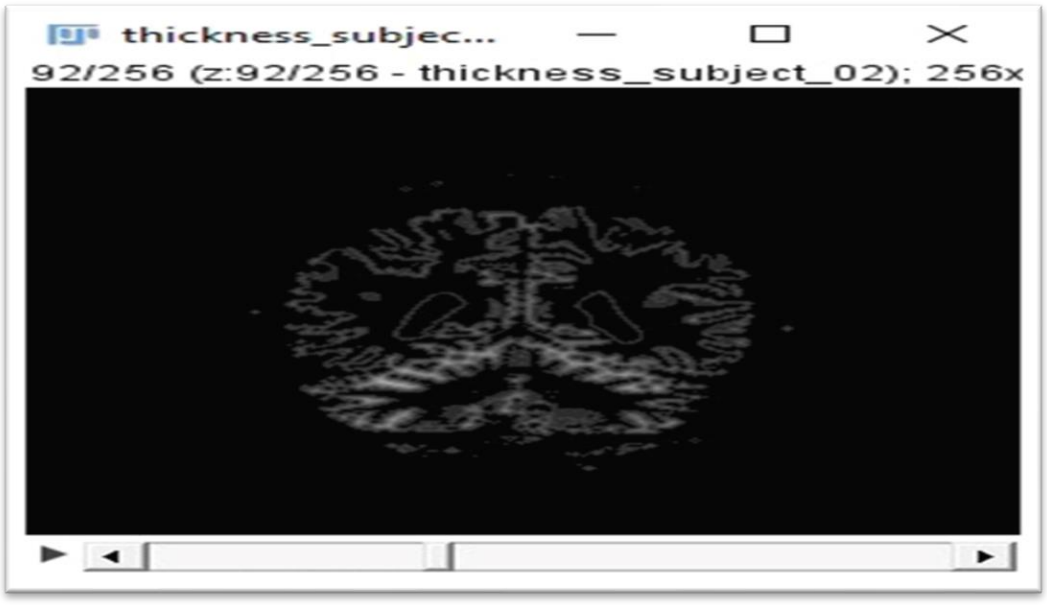
We have calculated distances from every grey matter voxel to nearest white boundary point and updated the value of every voxel to that distance which we are getting from K-d Tree and distance_transform_etd from scipy.ndimage we have calculated the Cortical Thickness of gray matter.

For approach 1 we have use Colourmap as hot and for approach 2 we have used Colourmap as magma.From the observation of the image above as the distance gradually increases from the white matter boundary ,lighter the colour is and as there is decrease in distance from the white matter boundary the colour gets darker.The fluctuations observed are based on distance from white matter boundaries to gray matter boundaries.

- **Using Fiji Tool**

  As we can see white matter has been successfully segmented if we take this segmentation out we are left with simple cortical thickness.

  Using Fiji tool we can see the Cortical Thickness in 3D and the value of z corresponding to the distance.



CHALLENGES FACED:

- We have tried to use Bresenham3D approach but could not get desired solution. The screenshots for the trial code is mentioned below. Also in the submitted file for code we have commented the code

```
#code
def Bresenham3D(x1, y1, z1, x2, y2, z2):
    ListOfPoints = []
    ListOfPoints.append((x1, y1, z1))
    dx = abs(x2 - x1)
    dy = abs(y2 - y1)
    dz = abs(z2 - z1)
    if (x2 > x1):
        xs = 1
    else:
        xs = -1
    if (y2 > y1):
        ys = 1
    else:
        ys = -1
    if (z2 > z1):
        zs = 1
    else:
        zs = -1

    # Driving axis is X-axis"
    if (dx >= dy and dx >= dz):
        p1 = 2 * dy - dx
        p2 = 2 * dz - dx
        while (x1 != x2):
            x1 += xs
            if (p1 >= 0):
                y1 += ys
                p1 -= 2 * dx
            if (p2 >= 0):
                z1 += zs
                p2 -= 2 * dx
            p1 += 2 * dy
            p2 += 2 * dz
            ListOfPoints.append((x1, y1, z1))

```

REFERENCES :

- **KDTREE – example and library. https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KDTree.html**
- **Thickness and distance- https://stackoverflow.com/questions/73181955/how-to-measure-thickness-of-a-binary-3d-image-in-python**
- **https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.distance_matrix.html**
- **Research Paper for brain segmentation – https://translational-medicine.biomedcentral.com/articles/10.1186/s12967-020-02317-9#:~:text=Cortical%20thickness%20measures%20the%20width,magnetic%20resonance%20images%20(MRI).**
- **Tamnes CK, Østby Y, Walhovd KB, Westlye LT, Due-Tønnessen P, Fjell AM. Intellectual abilities and white matter microstructure in development: A diffusion tensor imaging study. *Human Brain Mapping.* 2010;31:1609–1625.**