MASTER'S PROJECT

(MEDICAL IMAGING- CS 590)

Dept. of Computer Science

Bishop's University

Sherbrooke, Canada

PROFESSOR: DR. RUSSELL BUTLER

TOPIC: CORTICAL THICKNESS CHALLENGE

SUBMITTED BY:

- ANJALI SINGH- 002322376
- NEHA SHEWALE- 002319810

OBJECTIVE

The goal is to create an algorithm for estimating cortical thickness maps and segmenting grey and white matter from raw T1-weighted images.

OVERVIEW

With image segmentation presently being the most often utilised application in neuroimage analysis, deep learning is a potential method for medical image analysis. A methodology for producing voxel-based cortical thickness (VBCT) maps from anatomical MRI is called VBM (Voxel-Based Morphometry). This method generates maps in which each voxel of the grey matter is given a thickness value, compares VBCT measures across many brain imaging, and compares the volume or concentration of the local grey matter. On a voxel-by-voxel basis, distinct variations in cortical thickness between participants or alterations in cortical thickness over time can be compared. The grey matter density of the human cortex brain is represented by a map of cortical thickness. The distance between the white matter surface and the pial surface is how it is described. The white-gray matter barrier in the tissue-classified brain volume is measured as a 3D distance from the cortical surface (gray—CSF boundary). When determining which parts of the brain are impacted by a condition, measuring therapy effectiveness, and researching brain ageing and development, the cortex's thickness is used. T1-weighted magnetic resonance scans can be used to calculate the human cortex's cortical thickness, which assesses the width of the grey matter there (MRI). Instead of using the vertices in the surface mesh, we compute cortical thickness estimates in the 3D image volume at a voxel level using grey matter segmentation of the picture.

BRAIN EXTRACTION

Brain extraction is an important preprocessing step in the analysis of magnetic resonance imaging (MRI) neuroimaging studies that influences the accuracy of downstream analyses.

Skull stripping: is a technique used to remove non-brain tissues from MR brain images for a variety of clinical applications and analyses; its accuracy and speed are regarded as critical factors in brain image segmentation and analysis. Accurate and automated skull stripping methods contribute to faster and more accurate prognostic and diagnostic procedures in medical applications. To separate the skull from the brain region, these methods employ morphological erosion and dilation operations. To determine the initial ROI, a combination of thresholding and edge detection methods must be used (region of interest). It consists of thresholding and morphological operations based on histograms. It distinguishes between desired and undesired brain structures based on anatomical knowledge. This method is implemented through a series of conventional and novel morphological operations in 2D and 3D.

APPROACH

- A distance mapping is a derived representation of a digital image, also known as a distance map or distance field. The term chosen is determined by one's perspective on the object in question: whether the original image is transformed into another representation or simply endowed with an additional map or field.
- Distance fields can also be signed, which is useful when determining whether a point is inside or outside of a shape.
- Distancemapping algorithms are useful in data-intensive applications for clustering and visualisation because they replace expensive distance calculations with sum-of-square calculations. Clustering in large databases with expensive distance metrics may become feasible as a result of this.
- They serve as the foundation for many popular and effective machine learning algorithms, such as supervised learning's k-nearest neighbours and unsupervised learning's k-means clustering.
- Depending on the type of data, different distance measures must be selected and used. As a result, it is critical to understand how to implement and calculate a variety of popular distance measures, as well as the intuitions for the resulting scores.
- The Distance Transformation algorithm operates on a binary image that contains both featured and non-featured pixels. It generates a distance map or distance matrix, with each cell corresponding to a pixel in the input image and containing a value indicating the distance to the nearest featured pixel.

IMPLEMENTATION

- Loading libraries and image to work on

```
import nibabel as nib
from scipy import ndimage as ndi
import matplotlib.pyplot as plt

import numpy as np
from nilearn import plotting
from nilearn import plotting

import pylab as plt
from nilearn import image as nli

from skimage import morphology
from skimage import measure

from skimage.util import montage
from skimage.transform import rotate
from nilearn.masking import apply_mask


from skimage.segmentation import mark_boundaries
```

```
#Loading dataset
data_frame=nib.load("raw_t1_subject_02.nii")
plot_data=data_frame.get_fdata()
```

- Finding threshold of image datapoints

```
thresh = np.percentile(plot_data,90)
thresh
```

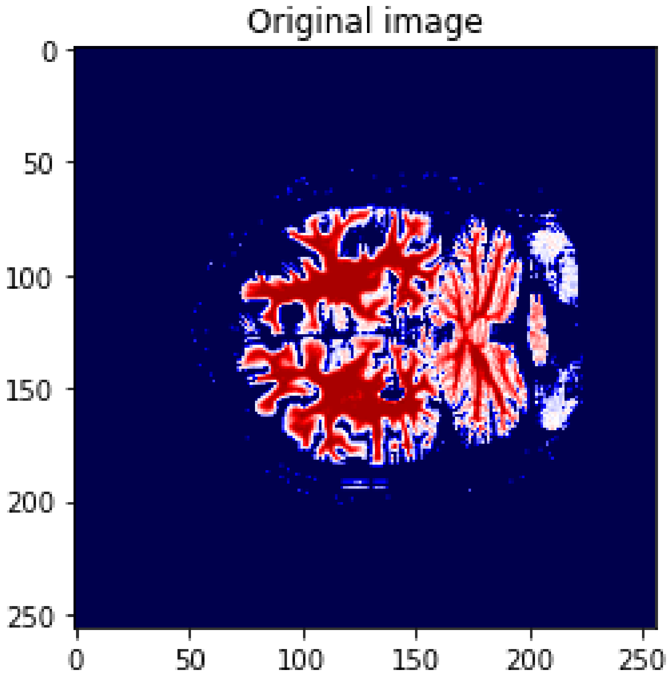
```
45.0
```

```
a=plot_data.copy()
a[a>thresh]=thresh
a.max()
```

```
45.0
```

```
b=plot_data.copy()
b[b<thresh]=0
```

```
plt.imshow(b[:,:,80],cmap='seismic')
plt.title('Original image')
```

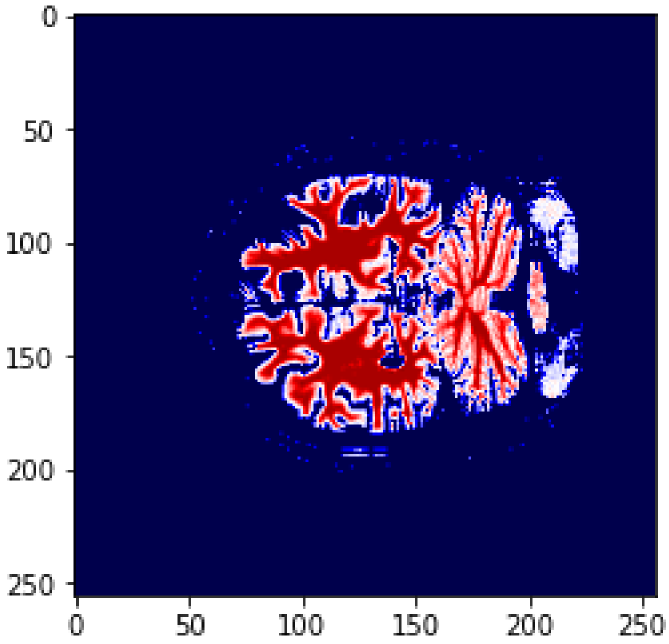


- Creating an instance of array with evenly spaced values.

```
b.shape
```

```
(256, 256, 256)
```

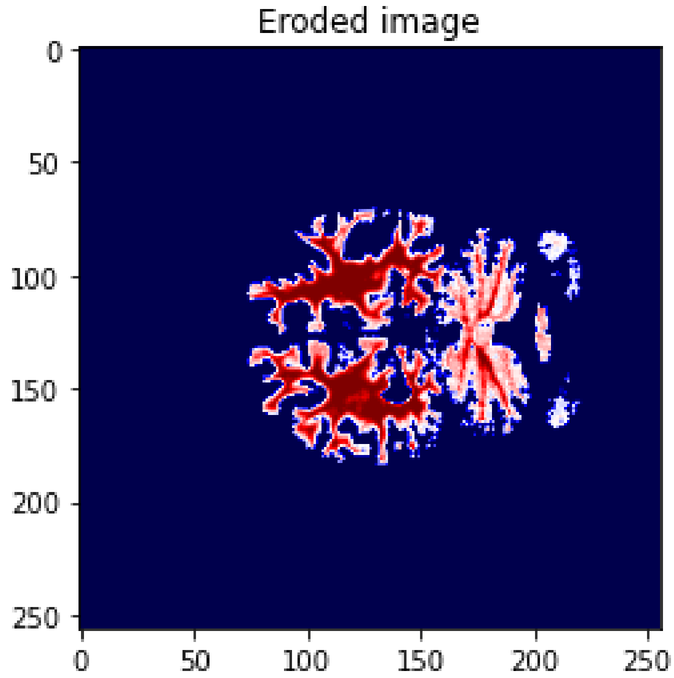
```
for i in np.arange(b.shape[0]):
    for j in np.arange(b.shape[1]):
        for k in np.arange(b.shape[2]):
            if(b[i,j,k]>50 and b[i,j,k]<100):
                if(b[i+3,j,k]<40 and b[i-3,j,k]<40):
                    b[i,j,k]=0
                elif(b[i,j+3,k]<40 and b[i,j-3,k]<40):
                    b[i,j,k]=0
            elif(b[i,j,k]<50):
                b[i,j,k]=0
```



- We will reduce the prominence of an image's features by erasing the foreground object's boundaries using erosion function.

```
erosion_i = morphology.erosion(b)
```

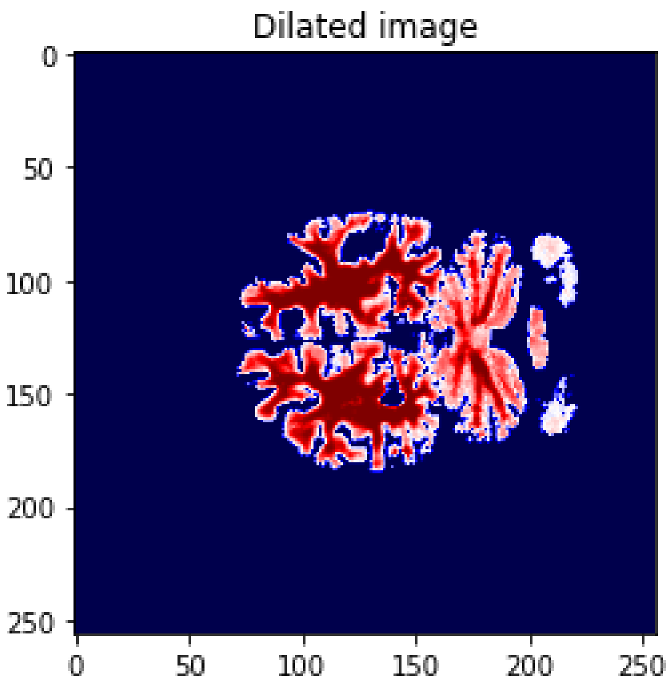
```
plt.imshow(erosion_i[:,:,80],cmap='seismic')
plt.title('Eroded image')
```



- We will use dilation feature to expand the object's surface area and to highlight features.

```
dilation_i = morphology.dilation(erosion_i)
```

```
plt.imshow(dilation_i[:,:,80],cmap='seismic')
plt.title('Dilated image')
```



- We will use thresholding function. Thresholding is a sort of image segmentation in which we modify a picture's pixel composition to facilitate analysis. Through the process of thresholding, we turn a colour or grayscale image into a binary image, or one that is only black and white.

```
grey = dilation_i-v
```

```
grey_b = grey[np.where(grey>0)]
```

```
np.min(grey_b)
```

```
50.0
```

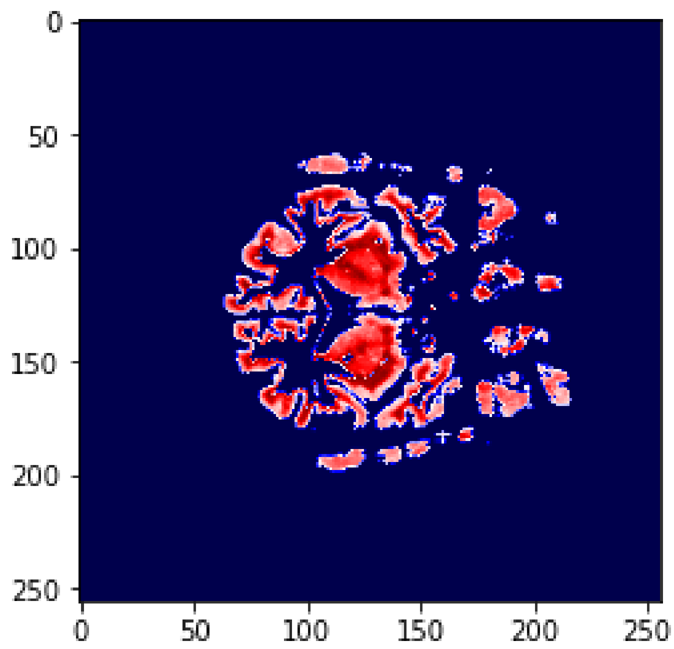
```
grey.shape
```

```
(256, 256, 256)
```

```
b_g = grey_b[np.where(grey_b==np.min(grey_b))]
```

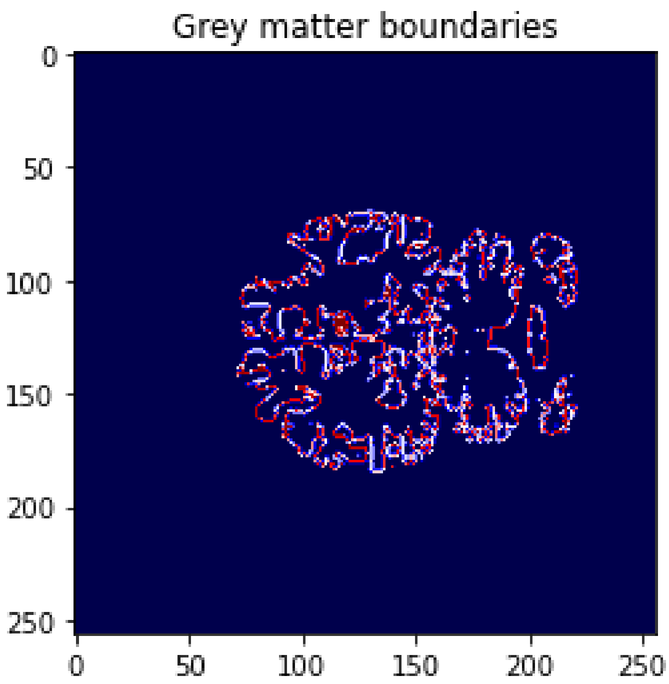
```
b_g.shape
```

```
(21036,)
```



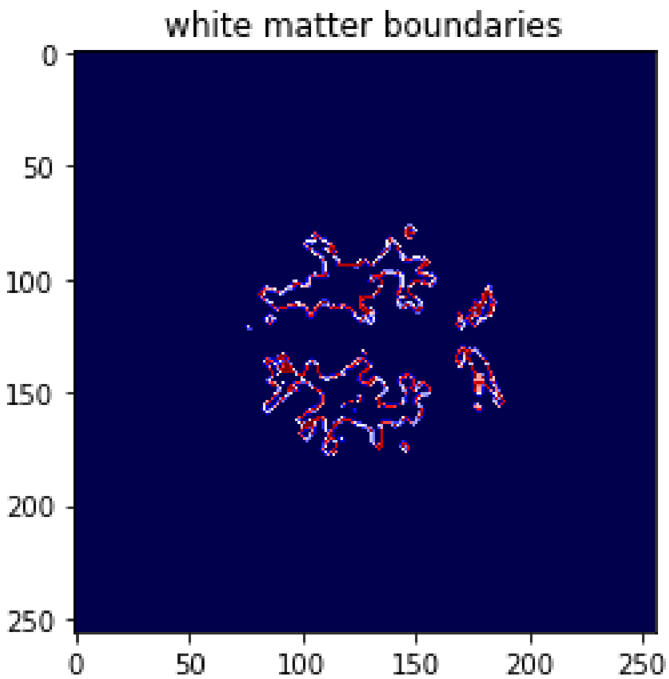
- First we will find grey matter boundary

```
grey_b=np.array([0]*256**3).reshape(256,256,256)
for i in np.arange(dilation_i.shape[0]):
    for j in np.arange(dilation_i.shape[1]):
        for k in np.arange(dilation_i.shape[2]):
            if (dilation_i[i,j,k]>0):
                if (dilation_i[i-1,j,k] == 0 or
                   dilation_i[i,j-1,k] == 0 or
                   dilation_i[i,j,k-1] == 0 or
                   dilation_i[i+1,j,k] == 0 or
                   dilation_i[i,j+1,k] == 0 or
                   dilation_i[i,j,k+1] == 0 ):

                    grey_b[i,j,k] = 1;


plt.imshow(grey_b[:,:,80], cmap='seismic')
plt.title('Grey matter boundaries')
```



- We will find white boundary of the image

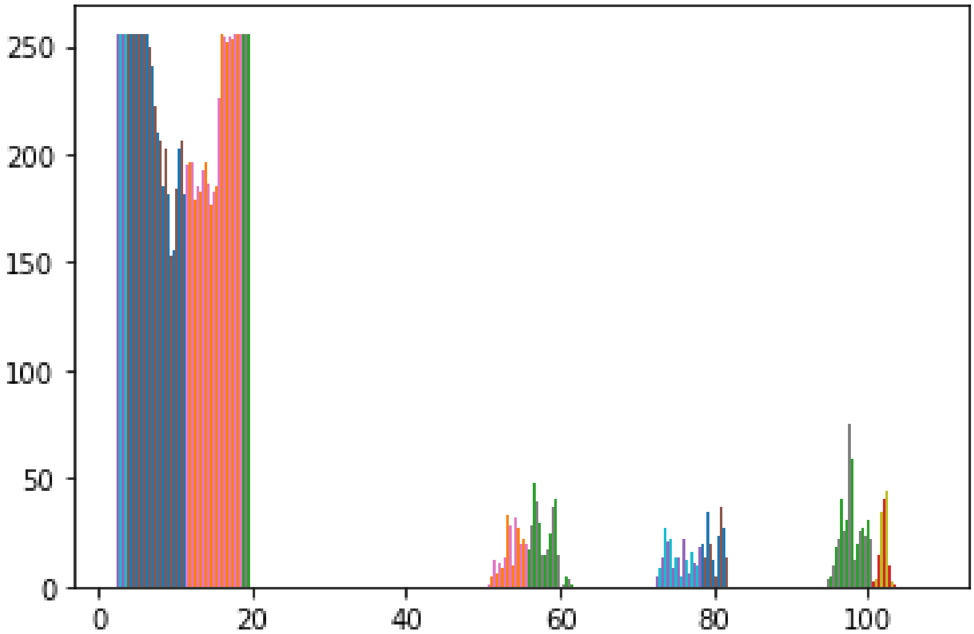
```
white_boundary = np.array([0]*256**3).reshape(256,256,256)

for x in np.arange(grey.shape[0]):
    for y in np.arange(grey.shape[1]):
        for z in np.arange(grey.shape[2]):
            if(grey[x,y,z]>0):
                if(v[x+1,y,z]>=100 or v[x,y+1,z]>=100 or v[x,y,z+1]>=100 or v[x-1,y,z]>=100 or v[x,y-1,z]>=100 or v[x,y,z-1]>=10
                    white_boundary[x,y,z]=1;


from skimage.measure import label

gry = label(grey)

gry=gry-white_boundary

plt.imshow(white_boundary[:,:,80],cmap='seismic')
plt.title('white matter boundaries')
```



- We will now observe a histogram representation of the eroded image.

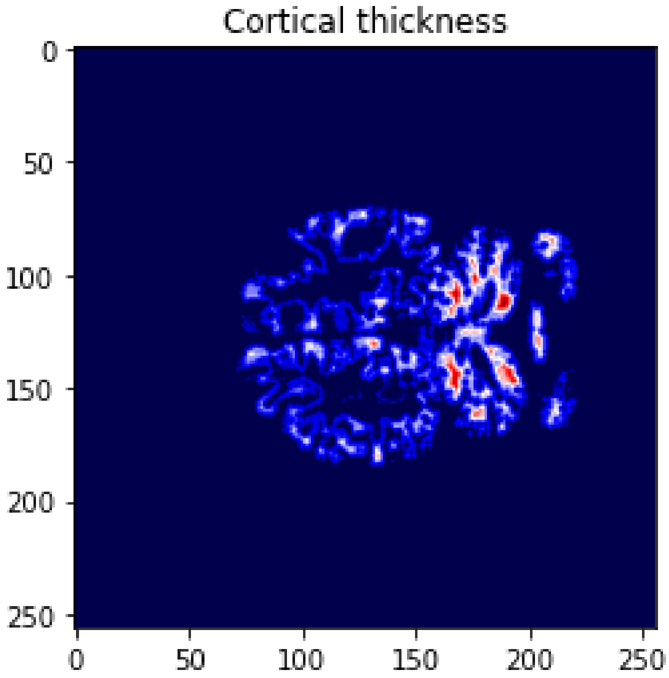
```
from skimage import segmentation

label_img=np.array([0]*256**2).reshape(256,256)
label_img=label_img.astype(int)
```

```
plt.hist(erosion_i[:,:,100],bins=5,linewidth=2.5)
```



- Now we will proceed to find the thickness of the cerebral cortex.

```
from scipy import ndimage
distanceThick = ndimage.distance_transform_edt(grey)
plt.imshow(distanceThick[:,:,80],cmap='seismic')
plt.title('Cortical thickness')

affine = np.eye(4)
final_file = nib.Nifti1Image(distanceThick,affine)
nib.save(final_file,"D:\Anjali\Semester 3\Masters Project\grey_thick.nii")
```


Cortical thickness
0
50
100
150
200
250
0
50
100
150
200
250

CHALLENGES FACED

- The brain and non-brain are divided in intensity-based approaches in accordance with the picture intensity. The watershed algorithm was used. The watershed method treats pixel values as a local topography starting from user-defined markers. The watershed method is built on capturing specific background and foreground information. Markers are then used to run watershed and determine the precise borders. But picture segmentation training takes a very long period.
- So, to separate the skull from the brain region, we performed morphological erosion and dilation techniques. These techniques call for the use of both thresholding and edge detection techniques.
- We tried our best but could not find a way to calculate nearest point in the skull. To overcome this, we have used Euclidean distance mapping algorithm. The Distance Transformation algorithm works on a binary image with both featured and non-featured pixels. It produces a distance map or distance matrix, with each cell representing a pixel in the input image and containing a value indicating the distance to the nearest featured pixel.

REFERENCES

- https://analyticsindiamag.com/what-are-the-different-image-thresholding-techniques-and-how-to-implement-them/
- https://www.mathworks.com/help/images/morphological-dilation-and-erosion.html#:~:text=Dilation%20adds%20pixels%20to%20the,used%20to%20process%20the%20image.
- https://scikit-image.org/docs/stable/api/skimage.segmentation.html
- https://link.springer.com/chapter/10.1007/978-0-387-68413-0_2
- https://www.pnas.org/doi/10.1073/pnas.200033797#:~:text=The%20human%20cerebral%20cortex%20is,thickness%20can%20be%20quite%20large.
- https://medicine.yale.edu/bioimaging/suite/manual/guide/extraction/#:~:text=The%20brain%20extraction%20tool%20is,brain%2C%20occupied%20by%20the%20CSF.
- https://nipy.org/nibabel/gettingstarted.html