CORTICAL THICKNESS CHALLENGE

Table of Contents

[Introduction 3](#_Toc109065820)

[Importing libraries 3](#_Toc109065821)

[White matter segmentation 4](#_Toc109065822)

[Image segmentation 7](#_Toc109065823)

[Thickness estimation 9](#_Toc109065824)

[Storing output files 10](#_Toc109065825)

[Conclusion 10](#_Toc109065826)

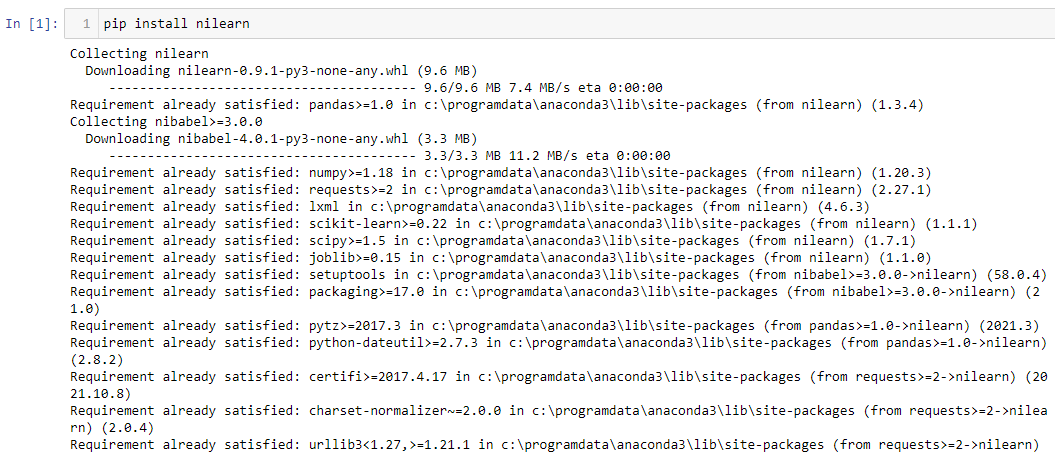
[References 12](#_Toc109065827)

# Introduction

The nifty images that is consists of the data related to the cortical thickness challenge should be loaded into python by the use of Nibabel libraries. The first time the user should probably want to segment the white matter. The process gives the starting point related to the thickness estimation algorithm. The purpose of the white matter segmentation is used to make decisions on the estimating thickness. The process is done by finding the vector related to the white matter surface and casting the vector outwards. This project is also done by using some heuristics to decide where the vector meets the pial surface that consists of the cortical thickness measured by the given vector length. The algorithm based on the topic of cortical thickness challenge is constructed by an algorithm that is mainly developed through the critical thickness map from a raw T1 weighted image. The map is formed by measuring the gray matter on the brain of the gray matter of the brain that is located at every point. The process of this system is used for defining the distance between the white matter surface and the pial surface. The output of the file is based on the python script that is used to represent the structure is mainly based on the three-dimensional structure and its output

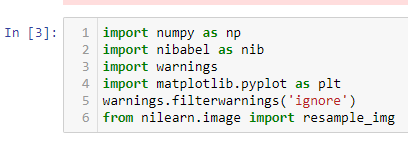
# Importing libraries

In this project, the nifty images into python should be loaded by using the nibble library files. The packages of the NiBabel library are mainly used to read and write for accessing various file formats related to standard neuroimaging. As per the view of Guo and Scheinost,(2022), this file format is used to provide read and write any access that is related to some medical and neuroimaging file format ***“GIFTI, NIFTI1, NIFTI2, CIFTI2, MINC1”***, etc as well as the Philips REC. It is used in various types of image format classes for full or selective access that is related to the header information. This is helpful to access the image data that is helpful to access image data that is available via Numpy arrays.



**Figure 1: Installing nibabel library**

(Source: Created by the learner)



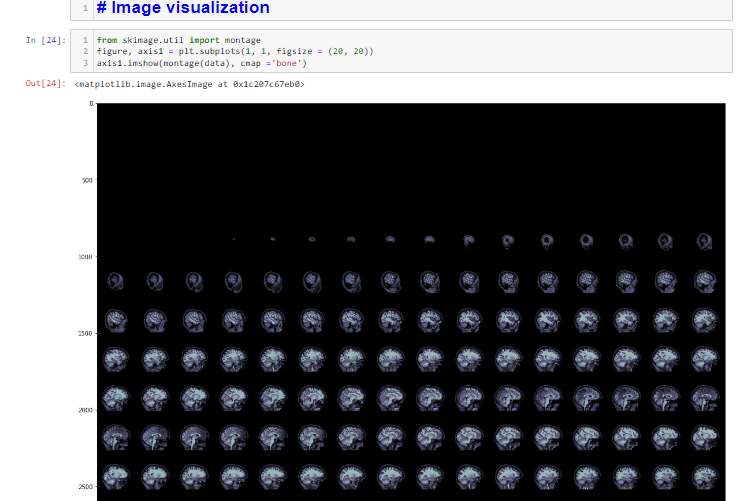
**Figure 2: Importing required libraries**

(Source: Retrieved from Jupiter notebook)

The following figures represent the libraries and packages imported for image segmentation and processing purpose. Libraries such as “nibabel, resample\_img” are imported for image segmentation purposes.

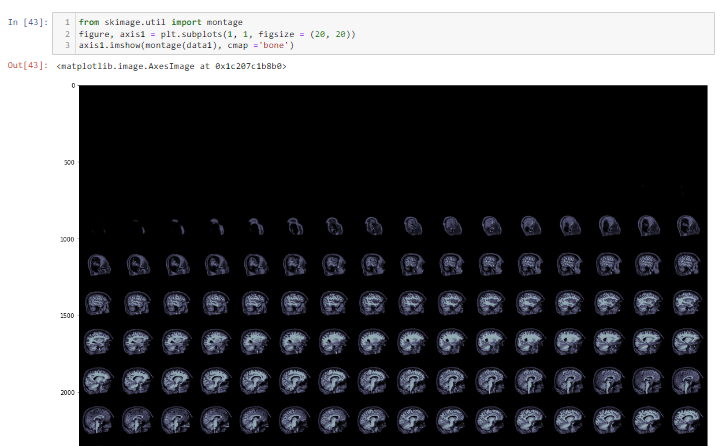
# White matter segmentation

The white matter segmentation is conducted by visualization the images president in the dataset. As per the view of Hai *et al.* (2022), here library named montage is used to visualize the montage data through subplots. This process allowed evaluating the differences between the white matter samples present within the datasets.



**Figure 3: Visualization of white matter**

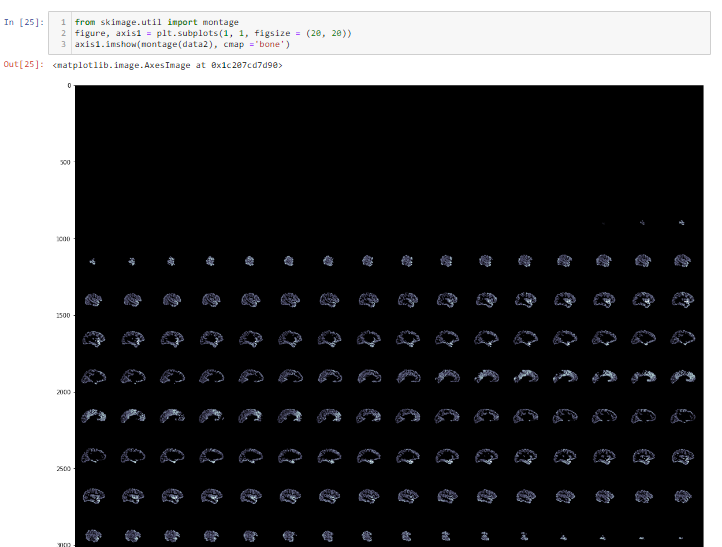
(Source: Created by the learner)



**Figure 4: Montage visualization**

(Source: Retrieved from jupyter notebook)

The following visualization is created by leveraging the ***“dataset named data1 and configuring cmap=bone”.*** Significant difference can be observed in the two different visualizations.



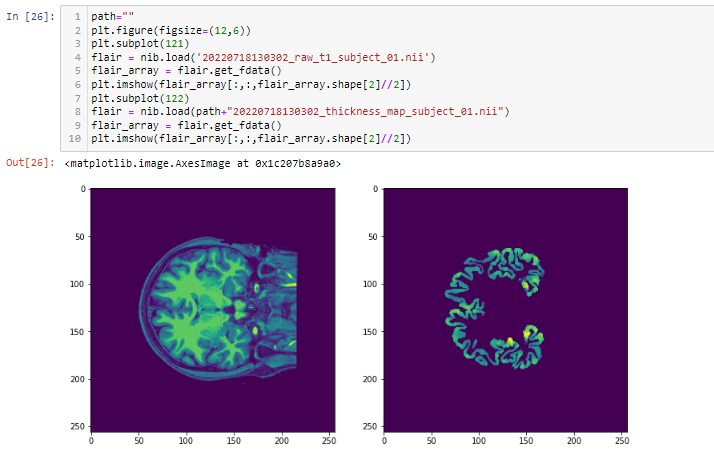
**Figure 5: Visualization on data2 dataset**

(Source: Collated from jupyter notebook)

The following image is the vitalization created based on the dataset2, based on this visualization it can be stated that the following visualization represents the earlier stages of white matter development and have less clear visibility as compared to other two visualizations.

# Image segmentation

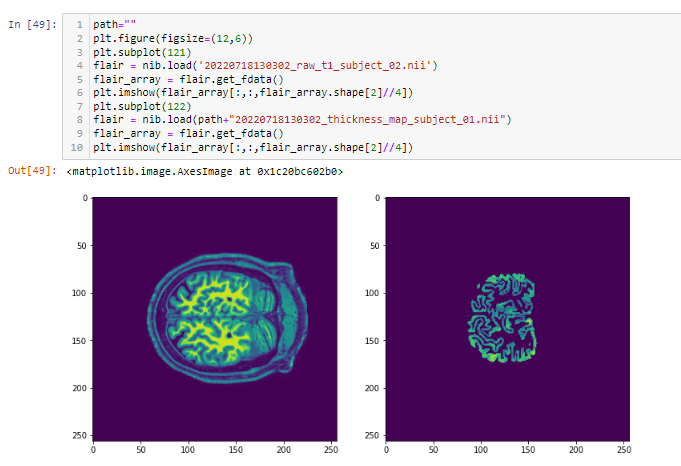
Image segmentation is performed on the provided datasets by configuring the “subject\_01 and map\_subject\_01 dataset into the flair”. As per the view of Lin and Zhang (2019), the flair is configured with a 2/2 array for the following image segmentation purpose.



**Figure 6: Image segmentation**

(Source: Crated by the learner)

The above image can be considered to observe the results obtained from the image segmentation process.



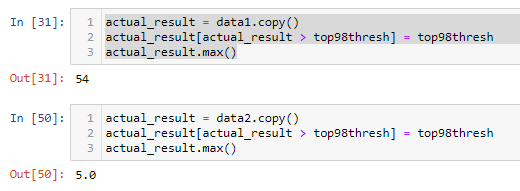
**Figure 7: 2nd image segmentation**

(Source: Crated by the learner)

The above mentioned image is the results obtained from the image segmentation which is conducted on “subject\_01”.As per the view of Kim and Seong (2022), here minimal difference can be observed by evaluating the two image segmentation results. The white matter shown in the second segmentation is more composed as compared to the previous segmentation.

# Thickness estimation

The thickness estimation of the white matter has been conducted by evaluating the min and max values of the dataset. Moreover, the percentile of data1 dataset is configured as “top98thresh” to configure its thickness.



**Figure 8: Actual results**

(Source: Created by the learner)

The corresponding figure represents the results obtained by configuring “top98thresh” within the method of actual results. Actual results from the two different datasets named “data1 and data2” are calculated at 54 and 5.0 respectively. Based on thus results it can be stated that the dataset1 images have relatively more thickness as compared to the images present in dataset2. Calculating the actuarial results was a challenging task as the file type is not suitable for this type of performance analysis.

# Storing output files

The output files that are generated through the image segmentation process are stored in empty files by configuring the empty files with ***“nib.Nifti1Header()”.*** This configuration is done as the desert files are provided in .nil format.

# Conclusion

In the whole process data sets are to be imported to the library files by the help of the NumPy library files. This is done by the nifty image files into the python by using the nibabel libraries. At first, the process is mainly enclosed by the segmentation process of white matter. The process is done for the purpose related to estimating thickness by different methods. The first method is to find a vector orthogonal to the white matter surface. The second process is to cast the vector that is related to the white matter. Surface outwards. This is done by using some heuristic for the purpose of deciding where

The vector meets the pial surface which is the essential tool for measuring the vector length by the cortical thickness. To calculate the whole data set the python script and thickness map for subject 02 that is containing a folder of single zip folder along with a report that is surrounded by indicating steps that are taken in this field.

# References

Guo, X., Wang, J., Rosenblatt, M., Dadashkarimi, J., Sun, H., Pollatou, A., Dufford, A. and Scheinost, D., 2022. MINDS Lab SLCN Challenge.

Hai, T., Swansburg, R., Kahl, C.K., Frank, H., Stone, K., Lemay, J.F. and MacMaster, F.P., 2022. Right, Superior Frontal Gyrus Cortical Thickness in Pediatric ADHD. *Journal of Attention Disorders*, p.10870547221110918.

Kim, S., Kim, S.W., Noh, Y., Lee, P.H., Na, D.L., Seo, S.W. and Seong, J.K., 2022. Harmonization of Multicenter Cortical Thickness Data by Linear Mixed Effect Model. *Frontiers in Aging Neuroscience*, *14*.

Lin, Y., Li, M., Zhou, Y., Deng, W., Ma, X., Wang, Q., Guo, W., Li, Y., Jiang, L., Hu, X. and Zhang, N., 2019. Age-related reduction in cortical thickness in first-episode treatment-naïve patients with schizophrenia. *Neuroscience Bulletin*, *35*(4), pp.688-696.

Suh, J.S., Schneider, M.A., Minozzi, L., MacQueen, G.M., Strother, S.C., Kennedy, S.H. and Frey, B.N., 2019. Cortical thickness in major depressive disorder: a systematic review and meta-analysis. *Progress in Neuro-Psychopharmacology and Biological Psychiatry*, *88*, pp.287-302.