**3D-Convolutional-Network-for-Alzheimer's-Detection**

This repository consists of an attempt to detect and diagnose Alzheimer's using 3D MRI T1 weighted scans from the ADNI database.It contains a data preprocessing pipeline to make the data suitable for feeding to a 3D Convnet or Voxnet followed by a Deep Neural Network definition and an exploration into all the utilities that could be required for such a task.

**Prerequisites**

* [Python3 (Os,Matplotlib,Panda,Numpy)](https://www.python.org/)
* [Scikit-Image](http://scikit-image.org/)
* [Nibabel](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/nipy.org/nibabel)
* [Nipype](http://nipy.org/packages/nipype/index.html)
* [FSL(FMRIB Software Library v5.0)](https://fsl.fmrib.ox.ac.uk/fsl/fslwiki)
* [Tensorflow](https://www.tensorflow.org/)
* [Tflearn](http://tflearn.org/)

**Data Preprocessing**

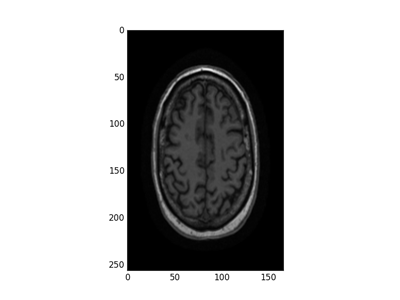
The ideal form in which the brain MRI image data can be sent for training is when it is skull-stripped, resized to a common size and labeled for all the different labels in the classification task.

**Data loading**

The first step is to load the data into Numpy arrays for futrher manipulation.The python library **Nibabel** has been used to access the MRI scan using its image object.The data attribute of this object is used to acquire the image in the form of a numpy array.

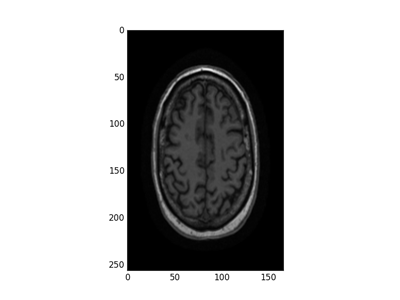
**Visualisation**

There is no predefined function in python packages to view 3D images using 3D axis however slice by slice visualisation can be done using **matplotlib**.Slice by slice visualisation can take place without manual input of the depth value using the method given by Juan Nunez-Iglesias in his blog [here](https://www.datacamp.com/community/tutorials/matplotlib-3d-volumetric-data).The same method is adopted above however the slice values are increased by tapping 'Q' and decreased by tapping 'A'.The gif below shows this form of visualisation.

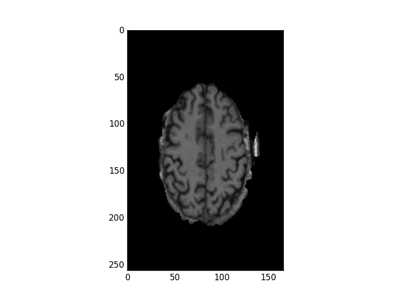
[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/brainscan/normal.gif)

**Skull Stripping**

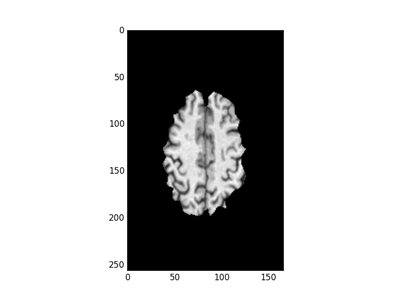
**FSL** is a library used for analysis and manipulation of MRI brain imaging data.**Nipype** provides an interface to use the **FSL** library via python code.Thus this is used to skull strip the images in the given code.The *frac* parameter is used to pass values for the fractional intensity threshold.A smaller value will give a much better estimate of the brain at the cost of lesser stripping.A sample image has been displayed below after skull stripping it with different *frac* values.

[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/brainscan/normal.gif)

frac=0.0

[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/brainscan/ss02.gif)

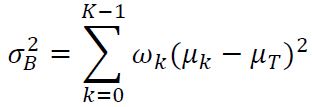
frac=0.2

[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/brainscan/ss05.gif)

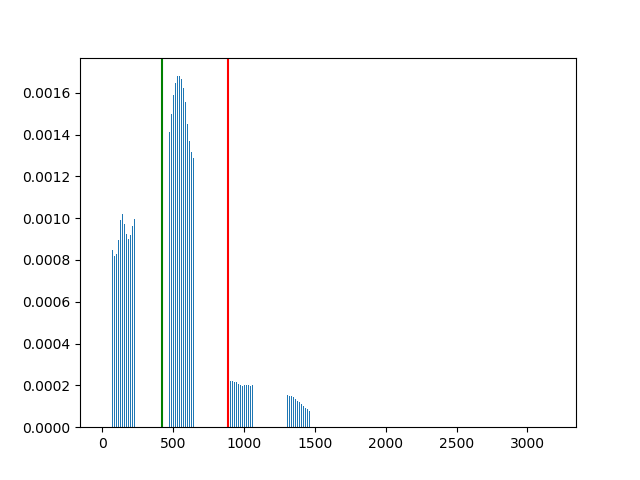
frac=0.5

**Histogram Thresholding & Segmentation**

An approximation of the amount of grey matter,white matter and CSF(Cerebrospinal Fluid) can be found using **Multi Otsu Histogram Thresholding** where the maximum variance is given by the formula:

[](https://camo.githubusercontent.com/fa18226b0f182bd6289c205c783b42bdbbeb7af6/68747470733a2f2f692e737461636b2e696d6775722e636f6d2f786e474b6d2e6a7067)

There's no predefined function for this in python packages therefore the code for it has been written firsthand in python. A sample of the thresholds derived in a skull stripped images are given in the histogram below.

[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/histogram/Figure_1.png)

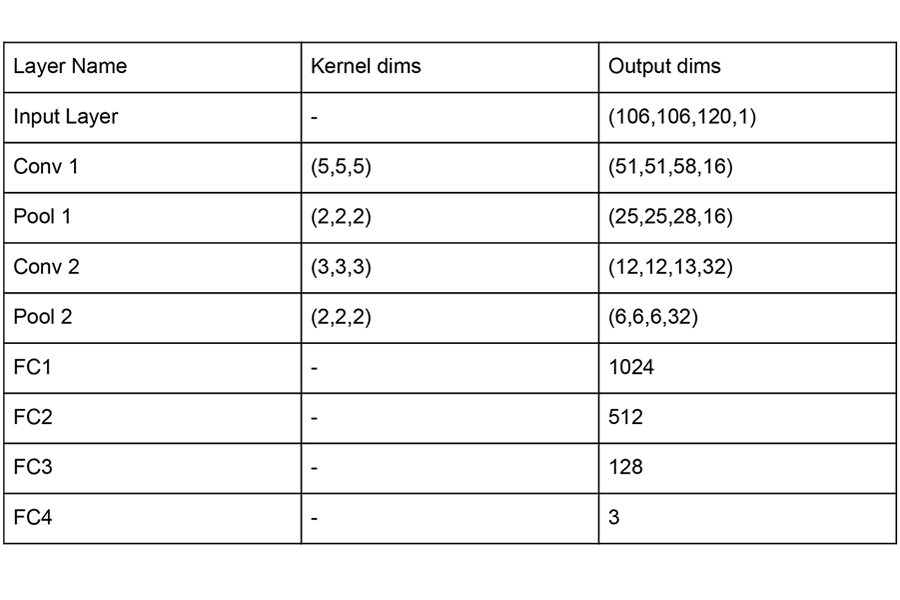
This is a form of global thresholding however better approximations can be made using adaptive and dynamic forms of thresholding where parts of the image are segmented at a time.

**Final Touches**

All the images are resized to the same dimensions using the predefined skimage transform.resize function. The values of all the pixels are then normalised so that faster training would occur after which the images are grouped with their labels and saved as numpy objects.

**3D CNN**

A 3D convolutional neural network has been defined using *Tflearn* which basically serves to provide wrapper functions for the tensorflow framework thus making it easier to create the network.The network uses mini batch gradient descent with batch normalisation for each activation layer.It uses dropout and L2 regularisation to tackle high variance and is optimised by the adam optimiser.It is designed for a 3 classification task with the classes as AD (Alzheimer's Disease), MCI(Mild Cognitive Impairment) and NL(normal). The layers for the network have been defined as per the table given below.The total number of parameters in this network is 7,670,960.

[](https://github.com/RishalAggarwal/3D-Convnet-for-Alzheimer-s-Detection/blob/master/3D%20Convolutional%20Network%20for%20Alzheimer's%20Detection/cnn%20table.png)

However the above network doesn't learn the deep features well enough therefore for experimentation purposes a Resnet network with 3D convolutions was used in hope of better feature learning due to it's shortcut connections. The Renet showed better performance but it wasn't significant.

**What's Next**

Training a 3D CNN for an end to end task like this is practically possible yet extremely difficult. Through such a procedure if the CNN is very Deep it's likely to overfit and if it's too shallow it's likely to underfit, it will only be able to mark boundaries satisfactorily if enough data is fed and region of interest localisation is done for training. This is largely due to the complexity of the problem.Alternatively the most successful approach has been to train the network on an 3D autoencoder as mentioned in thes paper: [Predicting Alzheimer’s disease: a neuroimaging study with 3D convolutional neural networks](https://arxiv.org/pdf/1502.02506.pdf).All this has been done on an 8GB RAM CPU laptop along with google colaboratory for network training.