11-19-2018

Detection of Autism using Magnetic Resonance Imaging data and Graph Convolutional Neural Networks

Saloni Mahendra Jain
sj7856@rit.edu

Detection of Autism using Magnetic Resonance Imaging data and Graph Convolutional Neural Networks

By
Saloni Mahendra Jain
November 19, 2018

A Thesis Submitted in Partial Fulfillment of the Requirements for the
Degree of Master of Science
in Computer Engineering from
Rochester Institute of Technology

Approved by:

Dr. Raymond Ptucha, Assistant Professor — Date
Thesis Advisor, Department of Computer Engineering

Dr. Shanchieh Jay Yang, Professor — Date
Committee Member, Department of Computer Engineering

Dr. Clark Hochgraf, Associate Professor — Date
Committee Member, Department of Electrical, Computer & Telecom Engr. Technology

R·I·T | Kate Gleason College of Engineering

Department of Computer Engineering

Acknowledgements

I am very grateful to Dr. Ptucha for his constant encouragement and guidance throughout my Masters education. I would like to extend my gratitude towards Miguel Dominguez for getting me involved in graphs and guiding me with my research. I would also like to thank my family, friends and fellow Machine Intelligence Laboratory members for their support.

I dedicate this work to my family and friends, their unconditional love and support has helped me throughout my journey.

Abstract

Autism or Autism Spectrum Disorder (ASD) is a development disability which generally begins during childhood and may last throughout the lifetime of an individual. It is generally associated with difficulty in communication and social interaction along with repetitive behavior. One out of every 59 children in the United States is diagnosed with ASD [11] and almost 1% of the world population has ASD [12]. ASD can be difficult to diagnose as there is no definite medical test to diagnose this disorder. The aim of this thesis is to extract features from resting state functional Magnetic Resonance Imaging (rsfMRI) data as well as some personal information provided about each subject to train variations of a Graph Convolutional Neural Network to detect if a subject is Autistic or Neurotypical. The time series information as well as the connectivity information of specific parts of the brain are the features used for analysis. The thesis converts fMRI data into a graphical representation where the vertex represents a part of the brain and the edge represents the connectivity between two parts of the brain. New adjacency matrix filters were added to the Graph CNN model and the model was altered to add a time dimension.

Table of Contents

List of Figures

List of Tables

Acronyms

ABIDE

: Autism Brain Imaging Data Exchange

MRI

: Magnetic Resonance Imaging

ASD

: Autism Spectrum Disorder

sMRI

: Structural Magnetic Resonance Imaging

fMRI

: Functional Magnetic Resonance Imaging

dMRI

: Diffusion Magnetic Resonance Imaging

rsfMRI

: Resting state Magnetic Resonance Imaging

HO

: Harvard Oxford

CC200

: Craddock 200

CDC

Center of Disease Control

BOLD

Blood Oxygen Level Dependent

CNN

Convolutional Neural Network

E2E

Edge to Edge

E2N

Edge to Node

N2G

Node to Graph

MLP

Multilayer Perceptron

PCP

Preprocessed Connectomes Project

ASOD

Autism Diagnostic Observation Schedule

fiq

Full scale IQ

piq

: Performance IQ

viq

: Verbal IQ

Chapter 1

Introduction

The Center of Disease Control and Prevention (CDC) has recorded the number of subjects diagnosed with ASD to have increased by 30 percent with the number of subjects diagnosed jumping from one in 88 children to one in 59 children over the period of ten years [11]. In today's world where technology has advanced by leaps and bounds, there is still no definite test for ASD. The only proven method to diagnose ASD is by observing the subject over a period and analyzing their behavior and development. This makes it difficult to diagnose ASD in a timely and effective manner. The symptoms of ASD can sometimes be observed as early as 18 months of age or earlier [11]. However, most of the subjects do not receive their final diagnosis till they are much older. In case of disorders like ASD, the earlier it is diagnosed, higher are the chances of effectively helping the subject by reducing the symptoms.

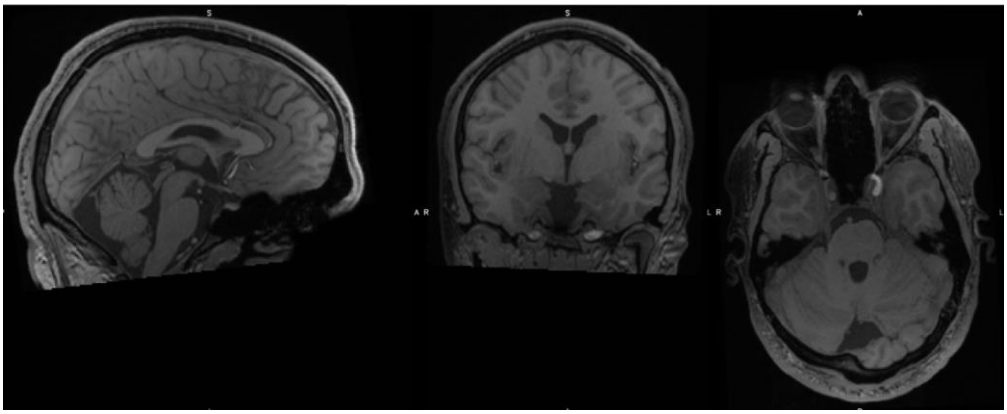
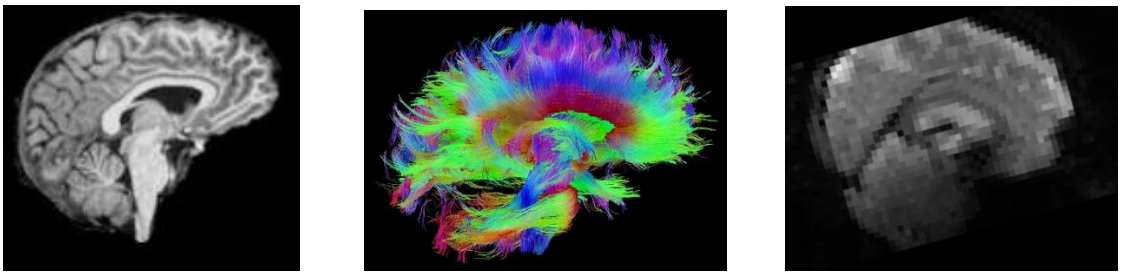


Figure 1. The left most image shows the Sagittal view, the middle image shows the Coronal view and the right most image shows the Axial view of a slice of the brain captured in the structural Magnetic Resonance Imaging data.



(a) (b) (c)

Figure 2 (a). The sagittal view of a sMRI scan [24].
Figure 2 (b). The sagittal view of a dMRI scan [24].
Figure 2 (c). The sagittal view of an fMRI scan [24].

The MRI data are brain scans which contain slices of the brain from three different views: axial, sagittal and coronal. There are three basic categories of data: structural Magnetic Resonance Imaging (sMRI), functional Magnetic Resonance Imaging (fMRI), diffusion Magnetic Resonance Imaging (dMRI). How do Magnetic Resonance Imaging (MRI) data help in diagnosis? FMRI data uses Blood Oxygenated Level Dependence (BOLD) effect to produce the fMRI scans. The raw fMRI information undergoes pre-processing to reduce the effect of any artifacts present in the data and to reduce it to a canonical form before extracting any features. After pre-processing, the resultant output is used to create a graphical representation of the data and extract features to perform classification using a deep learning model.

This master's thesis focuses on pre-processing the fMRI data present in the 'ABIDE I' dataset [11] and classifying the subjects into two categories, autistic and neurotypical, using deep learning models, specifically the Graph CNN model. The ABIDE I dataset contains information about 1114 subject obtained from 19 different sites from all around the world. It consists of 521 subjects with Autism and 593 subjects without Autism, with the age of the subjects ranging from 5 to 64 years [11]. This thesis explores the ability of Graph CNNs to perform detection of ASD with the help of features extracted from the dataset or the modifications made to the Graph CNN model created.

Chapter 2

Background

2.1 Graph

A graph is a structure which depicts the objects as vertices and the relationship between two objects as the edge connecting them. This structure can be used to extract features or analyze the pattern present in the data to perform classification or prediction.

An image is a gridded structure where all the vertices have the same fixed number of edges and each edge represents the same relationship between them. Performing analysis on a graph is difficult as it does not have the same gridded structure as an image [13]. A graph which is non-gridded in nature will have different number of edges connected to the vertices and the connection would not represent the same relationship between them. A CNN uses the same filter to analyze the data present in the entire image. If the relationship between the vertices are different in a graph then the same filter will be unable to extract meaningful features. Therefore, the traditional Convolution Neural Network (CNN) cannot be used to extract features or learn the properties of the graph.

2.2 Graph Convolutional Neural Network

Utilizing the ideas of a CNN on graphs is not as simple as performing standard convolutions and pooling operations on the input graphs. This is due to the fact that standard convolution operations work only on gridded inputs. The modifications required to make convolutions operate on general graphs will be discussed in this section. Consider a graph where the matrix representing all the vertices is represented by $V \in R^{NXF}$. The vertex matrix V

contains *N* vertices and each vertex has *F* features. The adjacency matrix *A* contains the connectivity information of all the vertices present in the graph. The values in the adjacency matrix represent if there is an edge present between two vertices as well as the value of the edge connecting them. Therefore, the adjacency matrix can be represented as, $A \in R^{NXN}$. The *N* represents the number of vertices present in the graph. The performance of the Graph CNN is permutation invariant, meaning that it is direction agnostic.

The aim of a graph convolutional neural network is to learn a function of the features of the input graph. The input graph contains feature information as well as the connectivity information of each vertex. There are two types of graph convolutional neural networks: spectral and spatial graph convolution neural networks. Spectral graph convolutional neural networks are based on spectral graph theory [32]. In this network the signal of the input graph is first converted to its Laplacian form using the degree and adjacency matrix as shown in (1).

$$L = D - A \tag{1}$$

In (1), *D* is the degree matrix calculated by performing row wise summation of the elements in the adjacency matrix and storing along the diagonal of an otherwise zero matrix, and *A* is the adjacency matrix containing the connectivity information. The model utilizes the spectral properties of the graph Laplacian [33] to perform filtering and pooling operations. The spectral graph convolutional neural network uses a filter defined in the Fourier space to filter the graph transformed into the spectral domain.

$$x * h = U^T.(Ux \odot h) \tag{2}$$

U is the Eigen basis of *L* which is the spectrally transformed graph signal *x*. *h* is the filter in (2) which is used to perform the filtering operations on graphs. The Eigen basis calculated for a graph Laplacian is unique for each input graph therefore, the spectral filters learned for one graph will not generalize for other input graphs. The Kipf and Welling graph convolution model is an example of a spectral graph convolution [31].

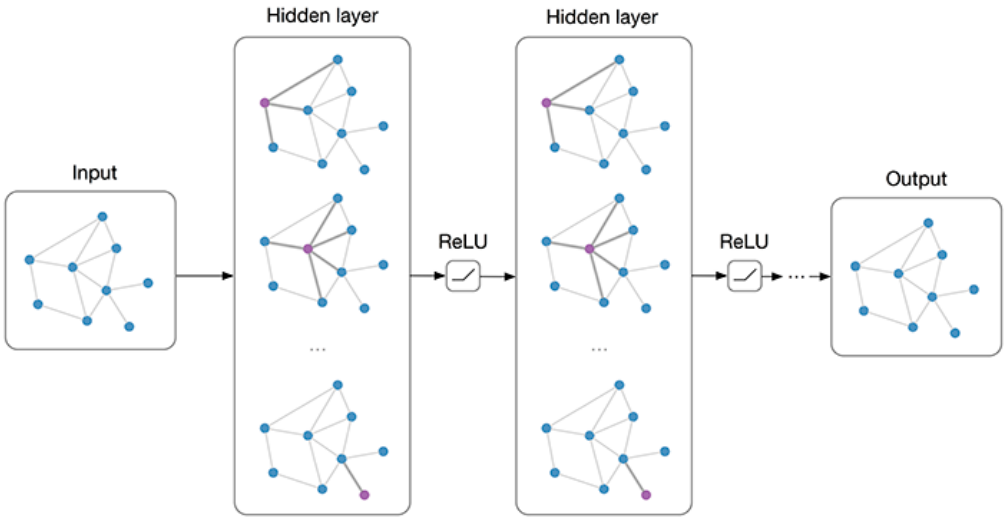


Figure 3. The graph convolution model created by Kipf and Wellington [31].

Spatial graph convolutional neural networks of 3D objects utilize the position of each vertex in the 3D space to create a function that represents the input graph. These neural networks use the position of the vertices in the 3D coordinate axis to calculate the distance of each vertex from all the other vertices present in the graph. If the 3D object were a surface scan, such as LiDAR, a mesh could be fit to all points. In other instances, such as fMRI brain scans, it is possible all points are connected to one another in which case the distance is used to decide if two vertices are directly connected by an edge or not. The connectivity information is stored in the adjacency matrix of the graph. When an adjacency matrix contains zeros (meaning that two vertices are not directly connected), the adjacency matrix also contains the information regarding the k-hop neighbors of all the vertices. The k-hop neighbor information represents the number of hops required by a vertex to reach some other vertex present in the graph. The number of hops can be decided by a variable k defined by the user. The adjacency matrices ranging from 1 to the k^{th} power and k isotropic filters are used to create a k^{th} order polynomial filter. This polynomial filter is then applied either on the Laplacian, L of the input graph or the adjacency matrix, A to filter the graph information. This filtering operation analyzes the distance information without any sense of direction and is therefore an isotropic method of filtering. In

order to overcome this limitation, Petroski Such et al. [17] partition the adjacency matrix into tensors, where each tensor represents a particular direction, and then applies separate filters on each partition.

2.3 MRI (Magnetic Resonance Imaging)

There are three types of MRI data: functional MRI (fMRI), structural MRI (sMRI) and diffusion MRI (dMRI). FMRI data is a low-resolution data which captures the connectivity information between different parts of the brain over a period of time. SMRI data is a high resolution scan of the brain. The different groves and parts present in the brain can be clearly seen and is hence used to create an anatomical map of the brain. DMRI data contains information about the connectivity between different white matter regions in the brain. It is captured by analyzing the diffusion trajectory of water molecules in each voxel.

2.3.1 SMRI

The sMRI data provides high resolution information about the anatomy of the brain as the information it captures does not change in the duration of a few minutes. Hence a tradeoff between time and resolution is observed here. The sMRI data depicts different types of tissues present by measuring the amount of water present in them [31]. This data is obtained using the transverse relaxation time (T2) pulse sequence. This pulse depicts gray matter as the darker region and white matter as the lighter region in the scans by measuring the concentration of water in different regions of the brain. Each subject's sMRI data is used as a reference image to perform coregistration, normalization and segmentation of the fMRI data.

2.3.2 FMRI

What is functional Magnetic Resonance Imaging (fMRI) data? How does fMRI data help in diagnosis? FMRI data measures the flow of blood in the brain using the Blood Oxygenated Level Dependent (BOLD) imaging [25] over a short period of time. Neurons do not have internal reserves of oxygen and therefore when they are active they are provided with oxygen at a

higher rate than inactive neurons. The BOLD imaging uses this principle and create a contrast between the active and inactive parts of the brain in an fMRI scan. This is in turn used to depict functional connectivity between different regions of the brain. Multiple scans are collected in a very short interval of time to depict the change in flow of blood. The fMRI data like all the other MRI data contain scans with the slices of the brain from three different views: coronal, axial and sagittal planes. The information from each slice of each view is combined with the other slices and views to create the information for the entire brain of a subject.

There are two types of fMRI data: resting state fMRI and fMRI. The resting state fMRI data is collected when the subject is not performing any activity. This data is usually collected either before or after the subject performs a task. The rsfMRI reveals information about the residual connections still firing and the activity in the brain while the subject is in a complete resting state. The collection of fMRI data is carried out while a subject performs any task. The nature and type of the task performed by the subject depends on the objective of the scans. These scans provide us with information regarding the regions of the brain active as well as the connectivity between them while the subject performs the assigned task. Both the rsfMRI and fMRI data is collected using the, BOLD imaging method.

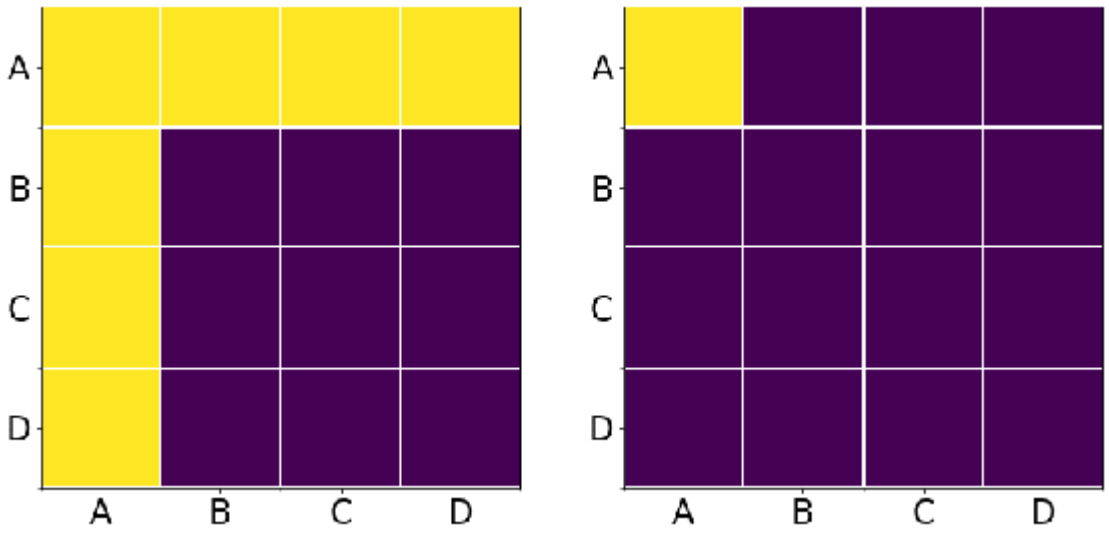
In this thesis research the rsfMRI data as well as the sMRI data from the Autism Brain Imaging Data Exchange (ABIDE I) dataset are used. The rsfMRI data provides information about the connectivity between different parts of the brain over a small duration of time while the subject is in resting state and the sMRI data provides a high resolution data which later separates the rsfMRI data into different region of interests (ROI).

Chapter 3

Related work

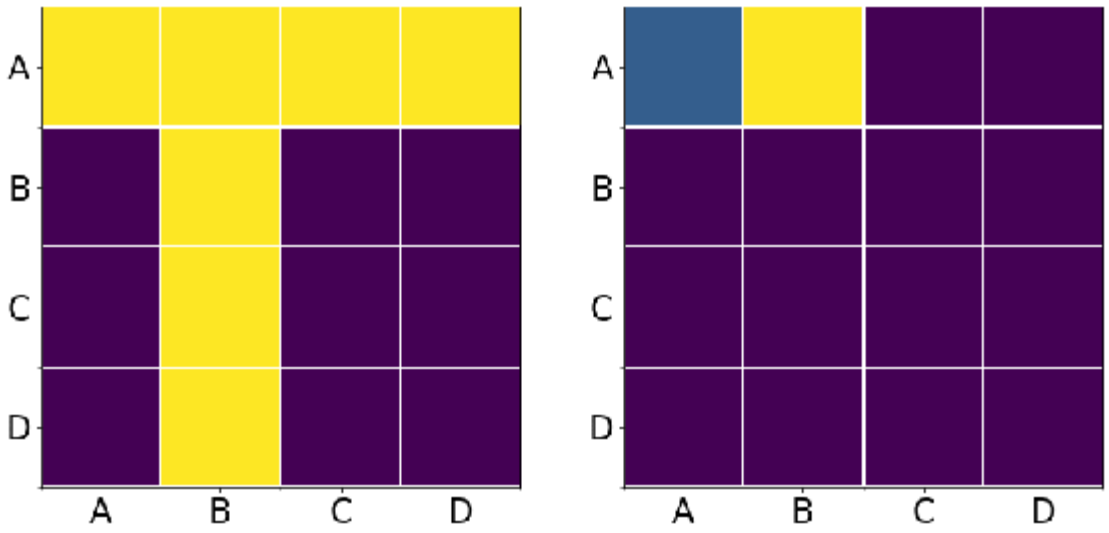
3.1 Convolutional Neural Networks for Brain Network

Kawahara et al. [16] proposed using a traditional CNN along with a few new filters added to predict cognitive and neuromotor outcomes of preterm infants. This paper introduced the use of edge to edge (E2E) filtering, edge to node (E2N) filtering as well as node to graph (N2G) convolutional filtering to extract features from the dataset and created a new network called BrainNetCNN [16]. These convolution layers perform simple convolution operations with filters of different shapes analyzing the connectivity in different ways.



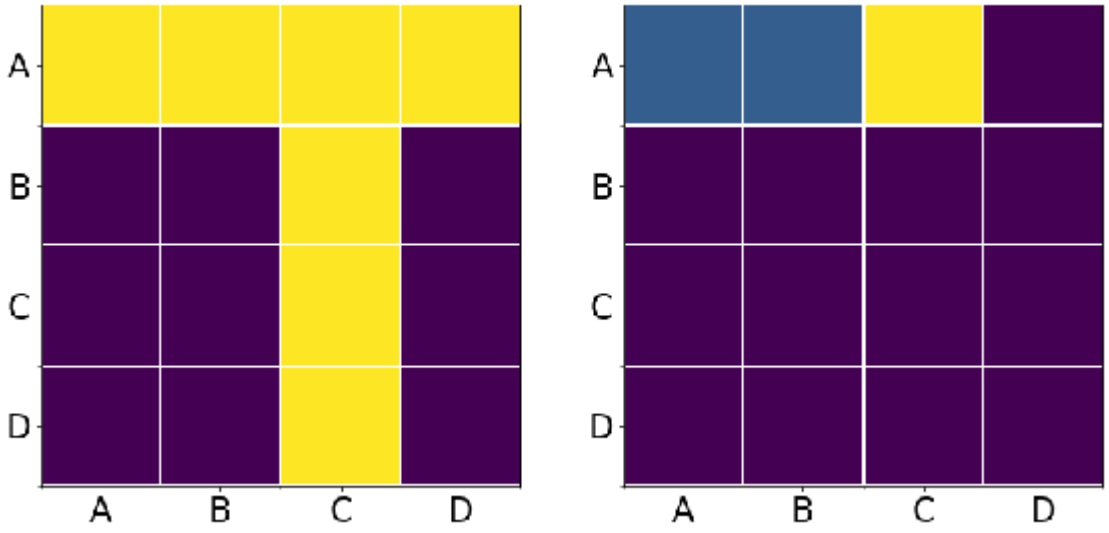
Input Output

(a)



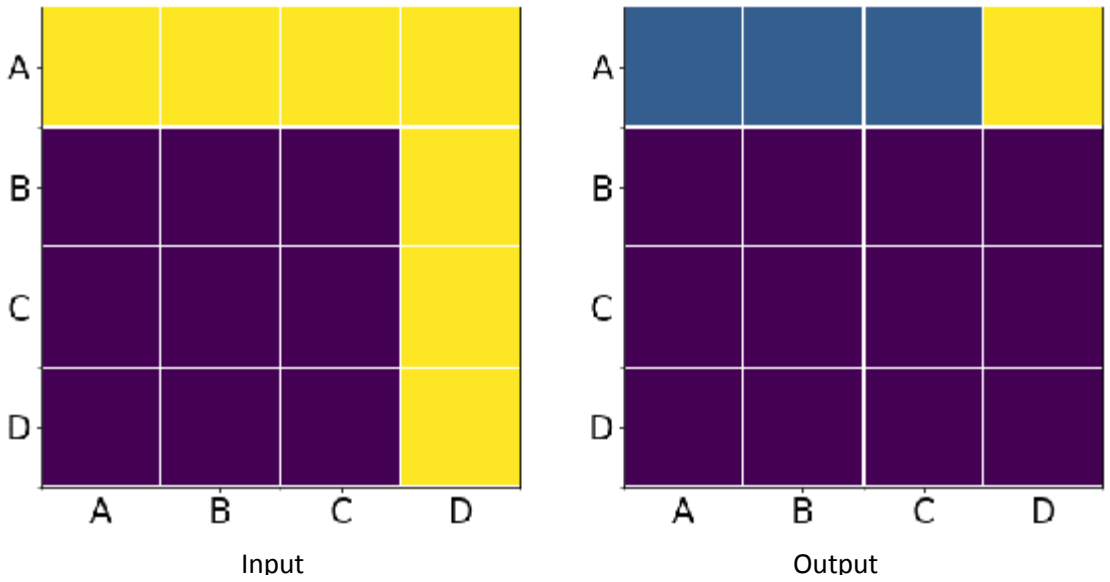
Input Output

(b)



Input Output

(c)



(d)

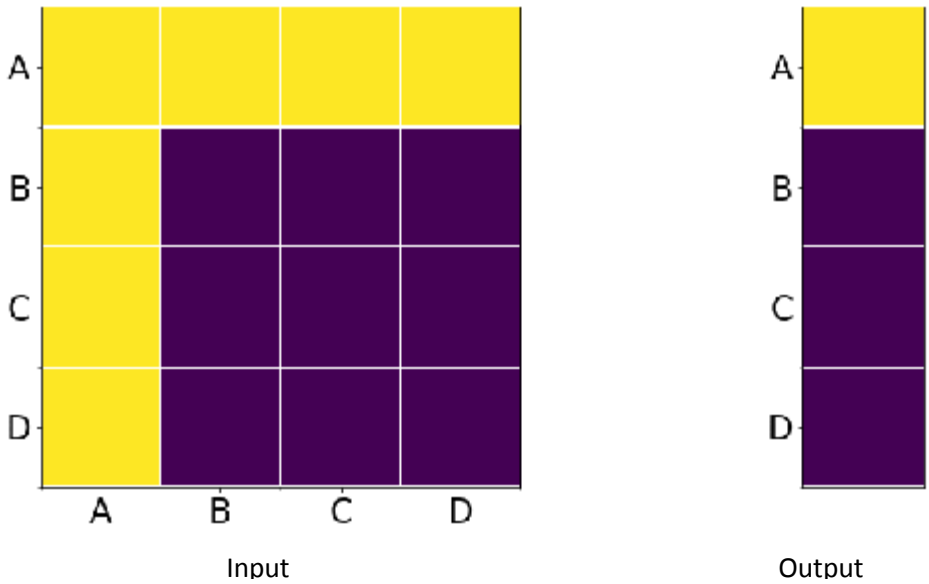
Figure 4. An adjacency matrix shown at four consecutive stages (a), (b), (c) and (d) of convolution. Two edge filters applied in yellow in the input matrix, the yellow cell in the output matrix shows the position of the corresponding output cell. Prior processed output values are in blue. The images depict how the filters move on the adjacency matrix in an edge to edge convolutional layer to analyze vertex A and create the first row of the output matrix [17].

The edge to edge convolution layer applies filters over the neighboring edges of the two vertices in question and performs their weighted sum to get a value representing that edge. For example, consider a graph with four vertices: *A, B, C* and *D*. Figure 4 shows the corresponding adjacency matrix of this graph. In order to perform edge to edge convolution two filters of dimensions *N*×1 are used. Therefore, in this example we use filters of dimensions 4×1. First the edge convolution is performed on vertex *A*. The value present in the adjacency matrix at position (*A, A*) depicts the connection of vertex *A* with itself. Therefore, we apply the filters to the row and column containing the connectivity information of all the edges connected to vertex A. The output of the two filters is then summed to create a single value which will be paced at position (*A, A*) in the output matrix. Then the vertical filter is moved one position left to analyze the edges

of vertex A and vertex B as shown in Figure 4 (b). The same process as before is repeated to obtain the resultant output value of position (*A, B*). Once all the edges have been analyzed from vertex A's point of view, the horizontal filter is moved one position below and the vertical filter is moved to the first column of the adjacency matrix. This process continues until the two filters have traversed through the entire adjacency matrix, filtering all edges. The size of the input matrix does not change after performing the edge to edge convolution. Hence, this layer was added before the edge to node (E2N) and the node to graph (N2G) convolution layers. The filtering process in edge to edge convolution process is carried out by using (3).

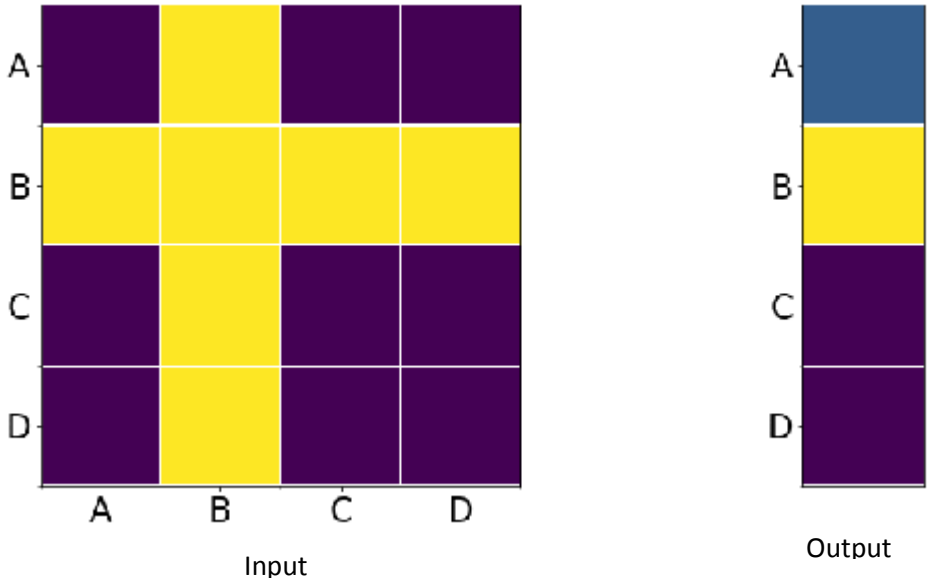
$$\mathrm{A}_{\mathrm{i,j}}^{\mathrm{l+1,n}} = \sum_{\mathrm{m=1}}^{\mathrm{M^l}} \sum_{\mathrm{k=1}}^{|\Omega|} \mathrm{r}_{\mathrm{k}}^{\mathrm{l,m,n}} \mathrm{A}_{\mathrm{i,k}}^{\mathrm{l,m}} + \mathrm{c}_{\mathrm{k}}^{\mathrm{l,m,n}} \mathrm{A}_{\mathrm{k,j}}^{\mathrm{l,m}} \tag{3}$$

Where *l* represents the *l*th layer of the CNN model, *m* represents the *m*th feature map of the brain, Ω is the nodes present in the brain region under consideration and *i, j* are used to represent vertex number. $A^{l,m}$ is the Adjacency matrix for the *m*th feature map of the *l*$^{\text{th}}$ layer. $[c^{l,m,n}, r^{l,m,n}] = w^{l,m,n} \in R^{2|\Omega|}$ such that $\left[w^{l,1,n}, \ldots\ldots, w^{l,M^l,n}\right] \in R^{2|\Omega| x M^l}$ are the weights learned at layer *l* for the *n*th filter. $A_{i,j}^{l+1,n}$ is the Adjacency matrix obtained after performing edge to edge convolution [16].



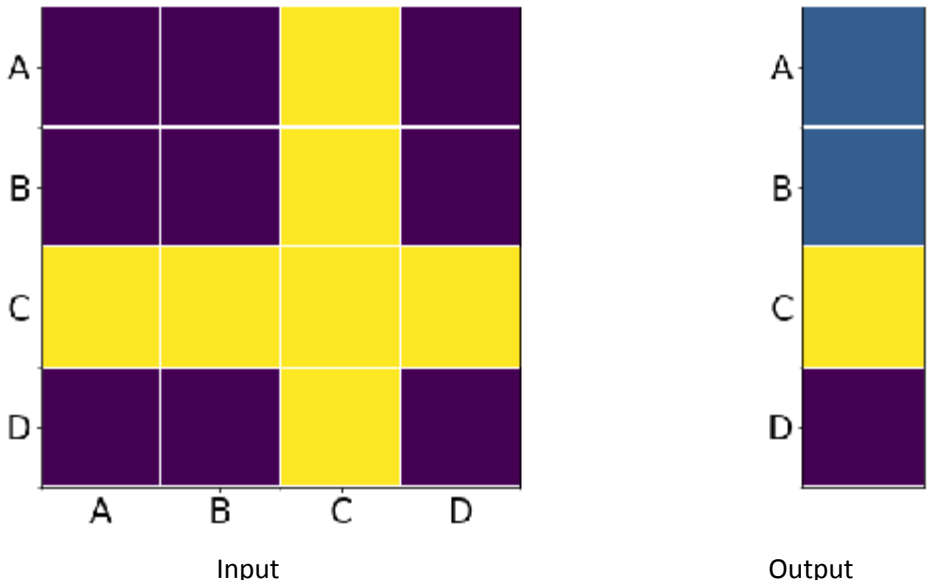
Input Output

(a)

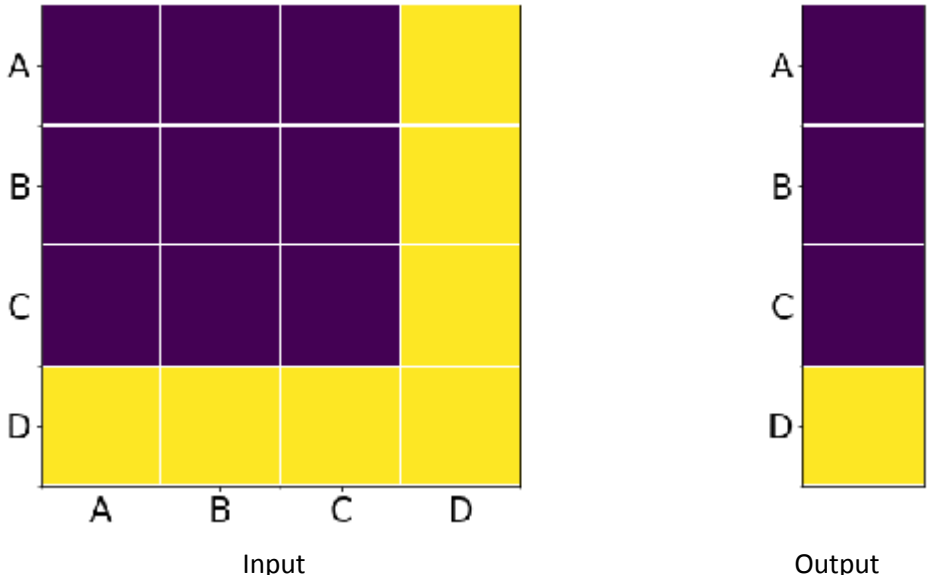


Input Output

(b)



(c)



(d)

Figure 5 (a), (b), (c) and (d). Show the two edge filters applied in yellow in the input matrix, the yellow cell in the output matrix shows the position of the corresponding output cell, the input adjacency matrix values in black and the output values in blue. The images depict how the filters move on the adjacency matrix in an edge to node convolutional layer to create the output matrix [17].

Adjacency matrices can also be filtered to create a node vector. The edge to node convolution operation is depicted in Figure 5 performs a filtering operation over all the neighboring edges of a single vertex and then calculates their weighted sum to generate the output nodes, one node per vertex. The horizontal filter moves one row down and the vertical filter moves one column to the right after every step. The input and the output dimensions are not the same. If the input was of dimension *N*×*N* then the output would have a dimension of *N*×1. This layer is used after the edge to edge convolution layer and before the node to graph layer. The filtering process in edge to node convolution layer is carried out by using (4).

$$a_{i,j}^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} r_k^{l,m,n} A_{i,k}^{l,m} + c_k^{l,m,n} A_{k,i}^{l,m} \tag{4}$$

Equation (4) is very similar to (3) for edge to edge convolution except, the feature maps is a vector of size *N* [16]. The edge to node graph convolution layer calculates a weighted sum of all the nodes in the graph and creates a single value for every feature map. Due to the nature of this convolution layer it is always used towards the end of the model. This layer is equivalent to converting from a convolution layer to a fully connected layer. The filtering process in node to node convolution layer is carried out by (5).

$$a^{l+1,n} = \sum_{m=1}^{M^l} \sum_{k=1}^{|\Omega|} w_i^{l,m,n} a_i^{l,m} \tag{5}$$

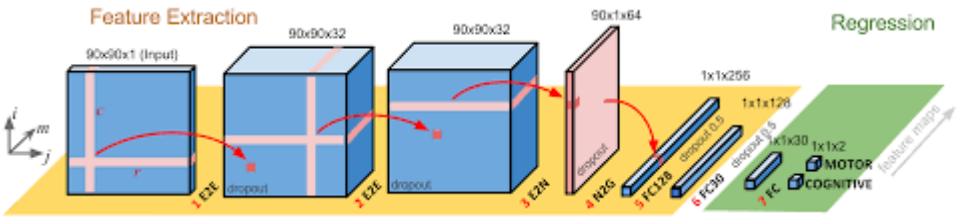


Figure 6. The architecture of the BrainNetCNN model. The depth of each block represents the number of feature maps [16].

The dataset used in this paper contained scans from 115 infants provided by the BC Children's Hospital in Vancouver, Canada. The dataset was augmented to produce a dataset which was almost 256 times the original dataset. The input to the model is the adjacency matrix containing the connectivity information about the different regions of interest present in the neonatal atlas. The model predicts the cognitive and motor development scores of the subjects.

3.2 3D CNN for classification of Functional Connectomes

Khosla et al. [18] preprocessed the rsfMRI data to extract the 3D spatial structure of rsfMRI data instead of only relying on the averaged information of each region. This paper used the connectivity information of each voxel with respect to each region of interest as the input to a 3D convolutional neural network model for detecting ASD.

Voxel-level maps are created by analyzing the connectivity information of each voxel with respect to the averaged value of each region of interest present in the selected atlas. The depth of each input image depends upon the number of regions defined in the atlas used to segment the brain. This depth of the images formed using the voxel-level maps range from 110 to 400. These images are called as connectivity fingerprints and are used as the input to a 3D CNN model.

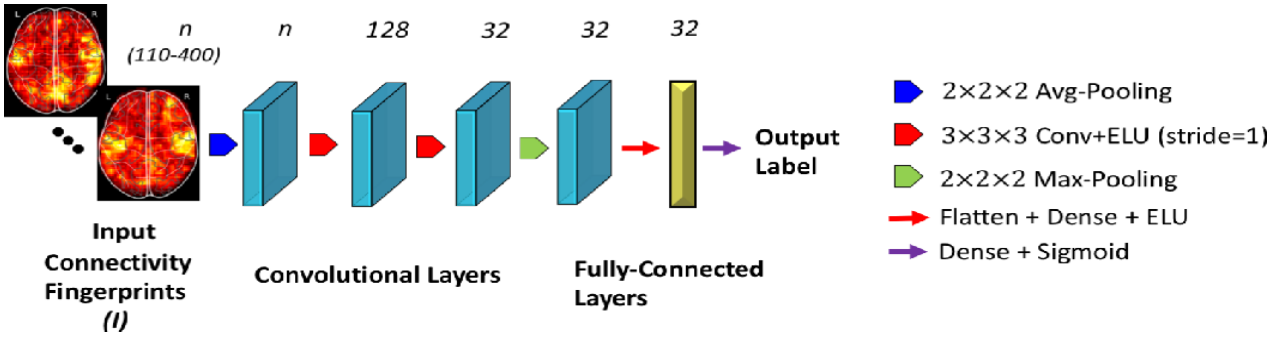


Figure 7. The connectivity fingerprints used as the input to the 3D CNN model used for detecting if a subject is neurotypical or ASD [18].

This paper has also experimented with the effects of using different atlases to perform segmentation of the brain. They concluded that the CC400 atlas performs the best with an

accuracy of 73.3% on the ABIDE I dataset [18].

3.3 Identification of autism spectrum disorder using deep learning and the ABIDE dataset

The main contribution of identification of autism spectrum disorder using deep learning and the ABIDE dataset written by Heinsfeld et al. [19] was the investigation of the connectivity in parts of the brain which showed a significant difference in their connectivity values for an ASD or neurotypical subject. This paper used autoencoders along with Multi-Layer Perceptron (MLP) to create a model and detect ASD.

The sMRI and rsfMRI data was obtained from the Preprocessed Connectomes Project (PCP) [20]. Here the raw data preprocessed by the C-PAC pipeline [21] was used. This pipeline performs skull stripping, motion correction, slice time correction, coregistration, smoothing and applies an atlas on the rsfMRI data to return a connectivity matrix. This connectivity matrix contains the connectivity information of all the region of interests present in the atlas. Therefore, if the CC200 atlas [22] was used then a connectivity matrix of shape $N \times N$ would be obtained, N representing the number of regions the atlas divides the brain into.

In addition to the connectivity information, the model also appends phenotypic information provided by the dataset for each subject. The data appended includes the age, sex of each subject and the Autism Diagnostic Observation Schedule (ASOD) score of the ASD subjects.

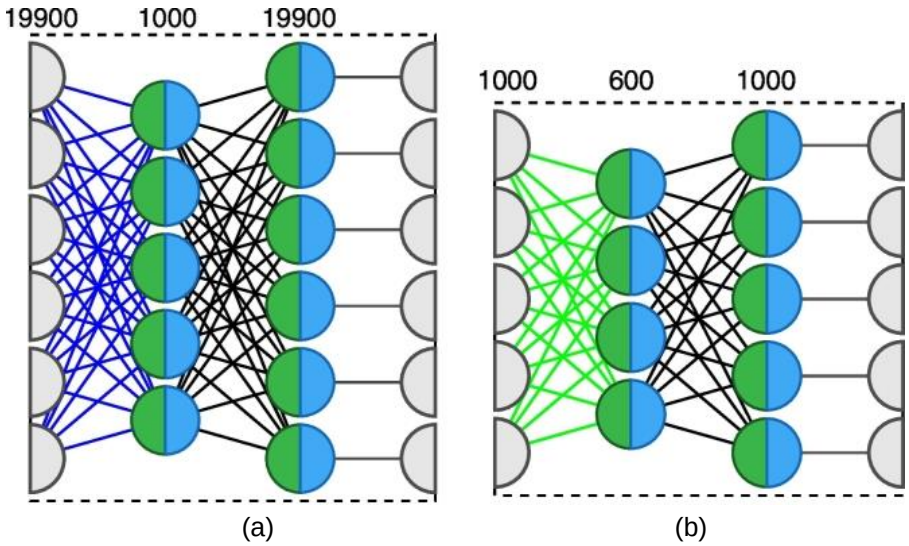


Figure 8 (a). The first autoencoder block used [19].
Figure 8 (b). The second autoencoder block used [19].

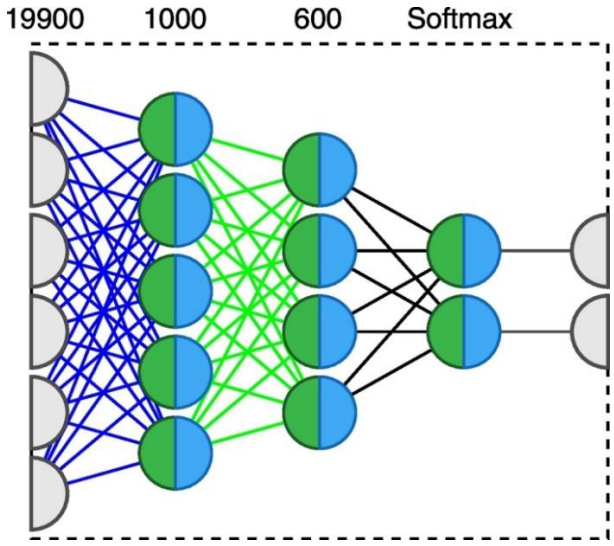


Figure 9. The MLP model created using the bottleneck layers of the two autoencoders and a softmax layer [19].

The connectivity information was passed through two denoising autoencoders [23] one with an input of 19900 nodes, a bottle neck of 1000 nodes and the probability of corruption of the data as 0.2. The second autoencoder has an input of 1000 nodes, a bottleneck of 600 nodes and the probability of corruption as 0.3. The bottlenecks of the two autoencoders are

then connected to a softmax layer to create an MLP model. The paper obtained an accuracy of 70 percent, however, the use of ASOD features made it easy for the model to detect ASD. This is because, the ASOD value was only present for ASD subjects and not for neurotypical subjects.

Chapter 4

Pre-processing

The MRI scans are extremely sensitive and even a slightest change in the parameters could greatly affect the captured data. The rsfMRI and the sMRI data is collected while the subject is awake and conscious inside the MRI machine. This increases the possibility of the subject moving their head inside the machine. This possibility along with multiple other factors can result in a variation between the data captured and the original information. The MRI data may get corrupted due to several causes like movement of the subject while scanning the brain, presence of noise in the scanner, variation in the shape and size of the brain of each subject, time required to capture the information and so on. In order to reduce the effect of noise or artifacts on the captured data some basic spatial and functional preprocessing methods are required to be executed before extracting any information from them.

The preprocessing methods that are essential to be applied to the sMRI data include: skull stripping, normalization, motion correction and noise reduction [26]. The preprocessing methods applied to rsfMRI data are: skull stripping, motion correction (realignment), normalization, time slice correction and coregistration. Once the data has been preprocessed to remove any artifacts or noise the data is then analyzed to extract the desired information.

The preprocessing was carried out by using Nilearn Neuro-Imaging library in python [10].

4.1 Spatial normalization

The size and shape of the human brain varies from subject to subject. While performing segmentation or extracting features each point in one brain should lie in the same location in another brain, to analyze or extract the same region for all the subjects. This also prevents the neural network to learn on their individual shape and sizes. Therefore, all the brains must be modified to a standard shape and size using a predefined standard template. This helps the model to select region of interests uniformly and reduce the effect of distortion [28].

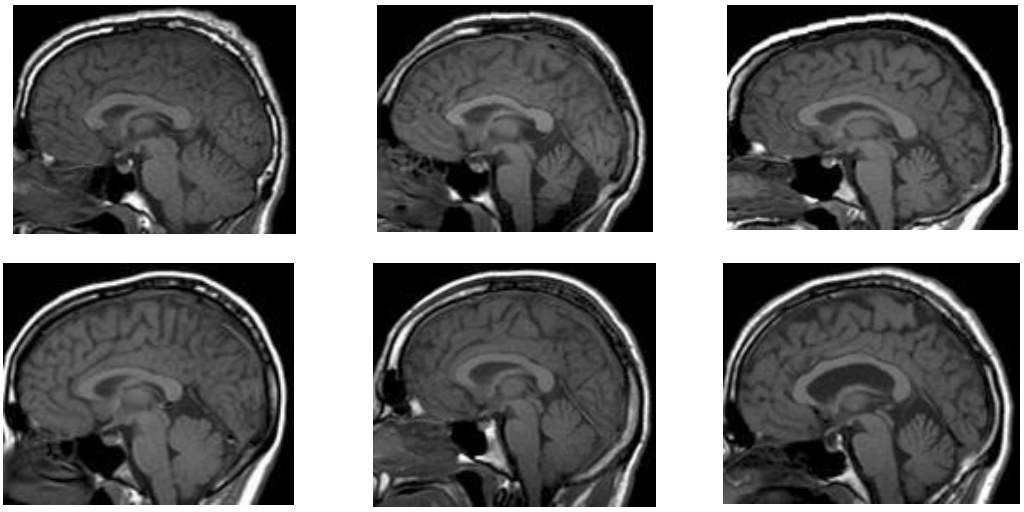


Figure 10. The figure shows the same slice sMRI scan of multiple subjects [2].

4.2 Skull stripping

Once all the brains have been converted to a predefined shape and size, the structure of the skull along with other parts of the body are removed as they does not provide any salient information. Hence all the information regarding the skull, eyes, spinal cord and the muscles in the face and neck are stripped from the dataset.

Figure 11. The axial view of a slice of a sMRI scans before and after the skull information was removed [3].

4.3 Realignment

The ABIDE dataset [1] contains scans from subjects with their age ranging from 5 to 64 years. It is difficult for all the subjects to lie perfectly still while the scans are being captured. This results in a change in alignment of the brain over the multiple scans of the same individual. To reduce this noise component, a process to correct this motion is performed. The process involves selecting one of the brain volumes of the subject as the reference volume and then aligning the data present in all the volumes of that subject to the reference volume. This reference volume selected was the center scan. The realignment of the brain takes place by using three dimensional parameters and 3 rotational parameters to move the data.

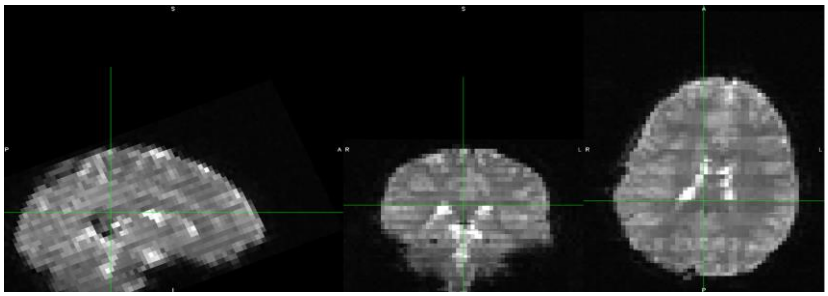


Figure 12. Slice of rsfMRI data before motion correction.

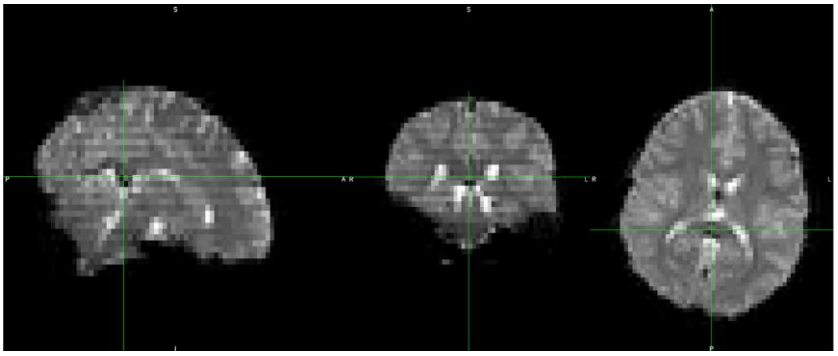


Figure 13. Slice of rsfMRI data after motion correction.

4.4 Smoothing (Noise reduction)

The spatial noise present in the rsfMRI scans are generally Gaussian noise. This noise is independent of each voxel and centered around zero. Therefore by averaging the intensity of the BOLD values over large number of voxels will reduce the noise towards zero and the signal towards a nonzero value. Hence performing smoothing on the images will improve the Signal to Noise Ratio (SNR) of the images by lowering the overall spatial frequency. It is equivalent to passing the image through a low pass liter. The drawback to this process is that the lower the SNR, the greater the amount of smoothing required of the image. Hence, a good tradeoff value should be decided for the dataset being used to avoid loss of valuable information. Smoothing will also help improve the spatial correspondence between the brains of the subjects at a group level.

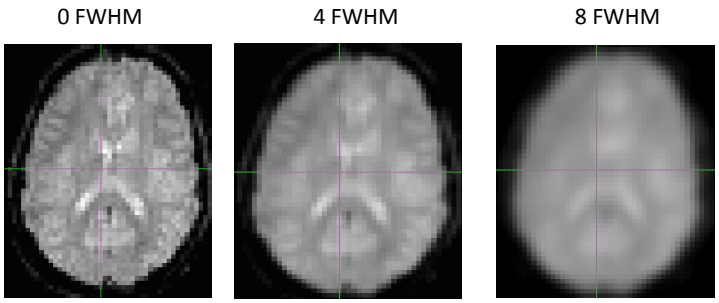


Figure 14. The figure shows a slice of the sMRI scan with varying amount of smoothing applied to it. The figure (a) has no smoothing applied to it. The figure (b) has a filter of 4 mm Full Width Half Max (FWHM) applied to it. The figure (c) is the result of the application of a filter of 8 mm FWHM [5].

4.5 Slice time correction

The resting state functional MRI data for each subject includes scans that capture the time series information. Each scan is to be recorded at a particular point in time to obtain the BOLD information for the same instant. However, it requires some time to obtain information for each slice in a volume. Therefore, the data captured at different layers of the brain are obtained at different time points. In order to resolve this problem slice time correction is performed. Slice time correction method calculates the Fourier transform of the signal at each voxel to convert the signal of that slice of the brain into a sum of scaled and phase shifted sine waves of each voxel. After which these converted signals are moved forward or backward depending on the interpolation. These values are interpolated between the values of the points that were actually sampled to obtain a value that would mimic the voxel value that would have been captured at that time instant.

4.6 Coregistration

The anatomical label maps of the brain are created to fit over the sMRI data. Therefore, the sMRI scan is used to coregister the rsfMRI data to the right dimensions and then the anatomical label map is used to segment the rsfMRI data into segments. The sMRI data

containing the anatomical data is present in the T2- weighted format while the rsfMRI data is present in the T1- weighted format [24]. The goal of coregistration is to align the sMRI and rsfMRI data in order to apply the anatomical masks created using the sMRI data. This is required so that we can have a one to one match of the ROIs defined in the atlas with the areas in the rsfMRI data.

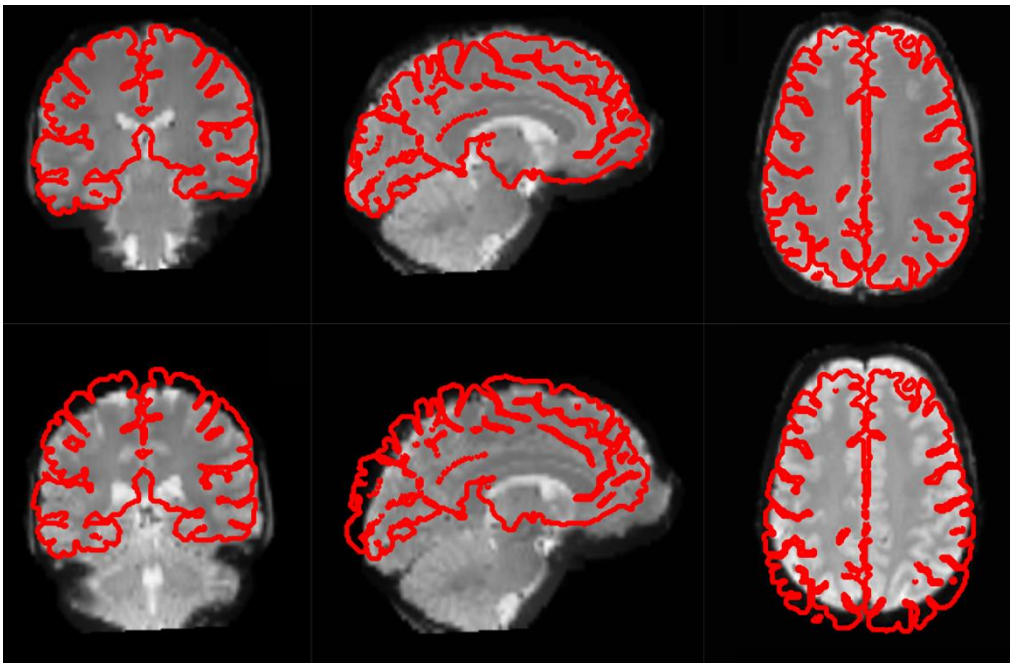


Figure 15. The red represents slice of the sMRI scan overlaid on the rsfMRI slice [24].

Chapter 5

Feature extraction

5.1 Atlas

The brain atlas in an anatomic label map which is used to divide and label region of interests (ROIs) in the brain [6]. The atlas chosen divides the brain into multiple segments. These segments are used to extract and analyze the time series and connectivity information from the rsfMRI data. There are multiple atlases designed by neurologists and they all divide the brain into different number of segments. Different atlases may focus on different parts of the brain as well. Some of the atlases are Harvard Oxford (HO) atlas [6], Automated Anatomical Labeling (AAL) atlas [7] [8], Craddock 200 (CC200), Craddock 400 (CC400) [22] and many more. This thesis primarily focused on using the HO atlas [6] for analysis.

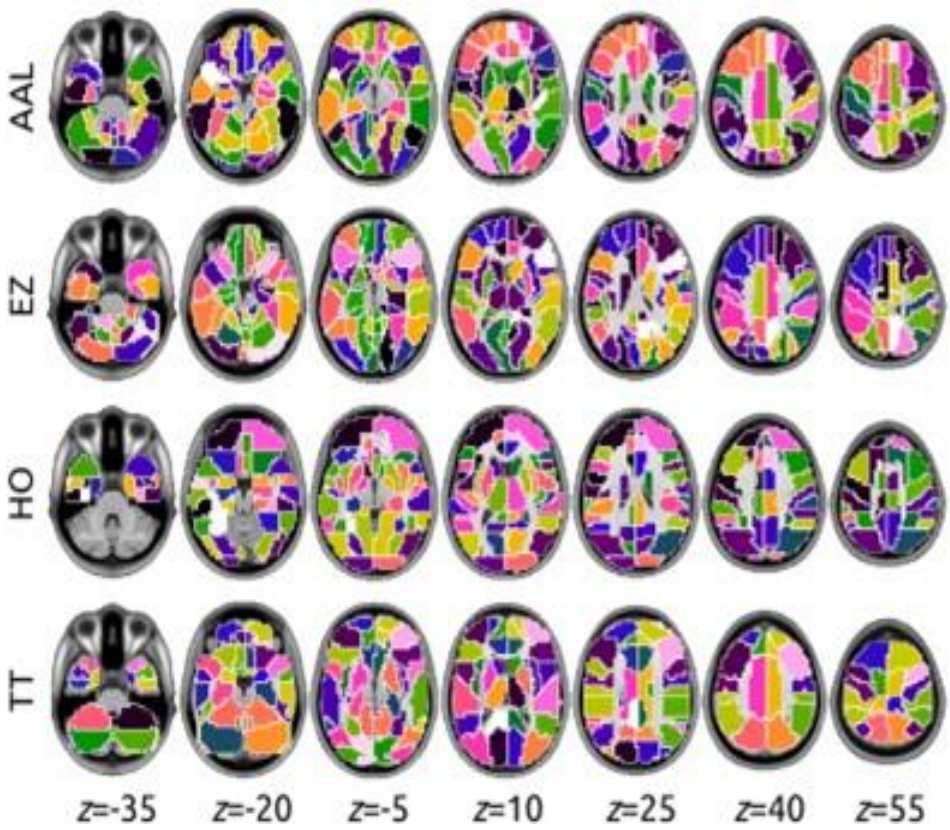


Figure 16. Different types of atlas and the regions they cover in a slice of non-pre-processed brain scan [35].

The HO atlas [6] contains 111 ROIs which cover regions from the cortical and subcortical regions of the brain.

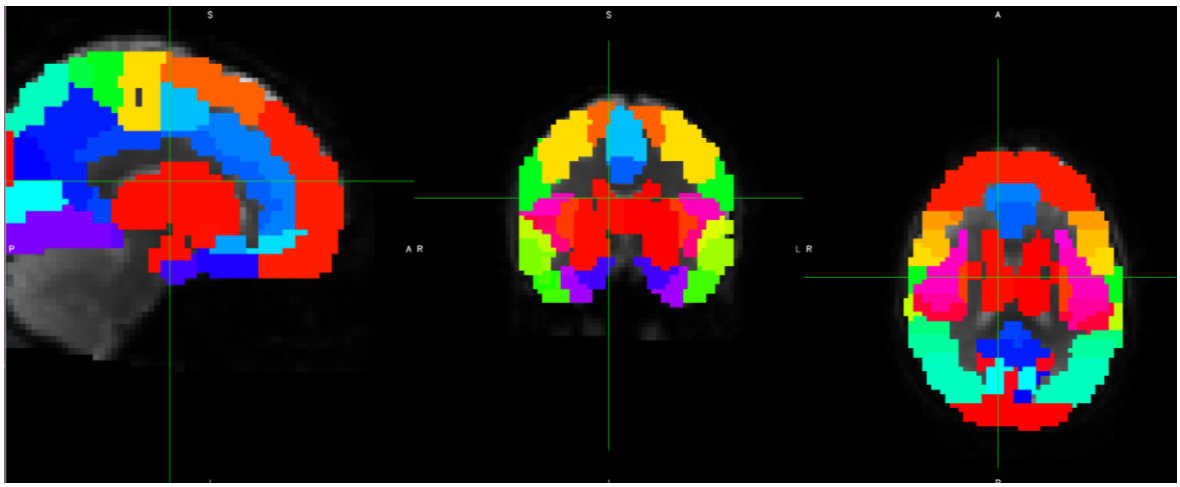


Figure 17. The Harvard Oxford atlas applied on a slice of pre-processed brain scan.

5.2 Time series

As the rsfMRI data contains multiple scans of the slices of the brain taken over time. The data collected is used to track any change in the neurological activity of a subject. Once the rsfMRI data is pre-processed to reduce noise and remove unnecessary artifacts, the desired atlas is superimposed on it. The atlas will segment the data into ROIs. Then the average intensity value of each region is calculated and used as the value representing that ROI.

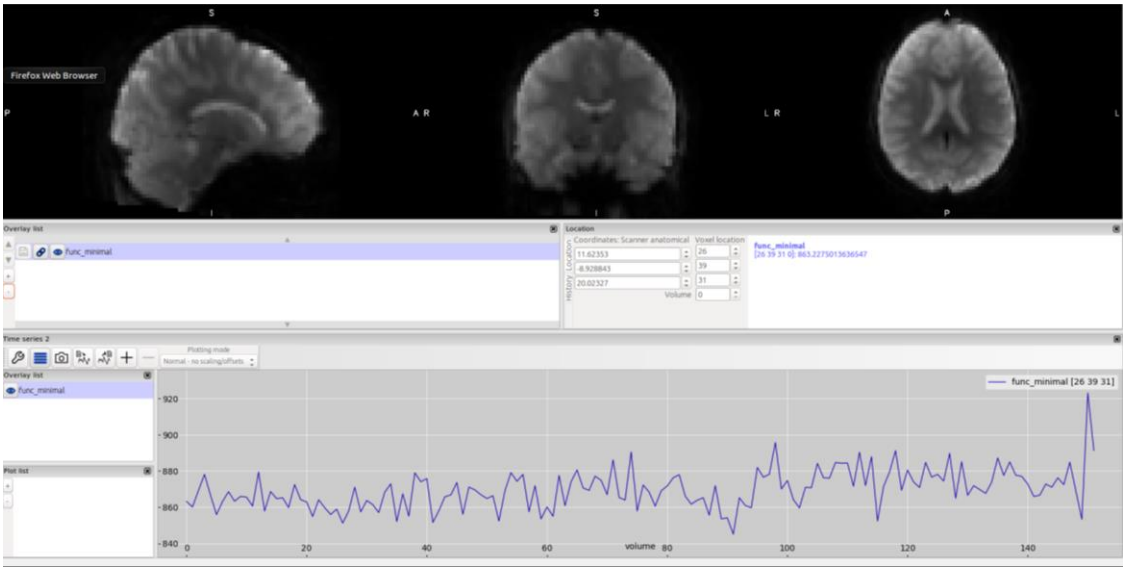


Figure 18. Time series information of a voxel present in a pre-processed rsfMRI data.

5.3 Connectivity

The anatomical label map divides the brain into different region of interests depending on the atlas used. Once the brain is divided, the connectivity information between these regions is to be calculated. This is done by calculating the average value of all of the voxel present in a ROI and then representing that ROI with the averaged value. This process is repeated for all the scans taken at different points in time. Once the average value representing all the ROIs is calculated then the correlation value between the ROIs is calculated to obtain the connectivity matrix. The correlation values range from -1 to 1, with -1 indicating that the two ROIs are

inversely correlated and 1 indicating that the ROIs are highly correlated. The model used in this thesis does not support negative edge values. Hence, the connectivity values are shifted by 1 and the range changes to 0 to 2 from -1 to 1.

The basic idea was to analyze the BOLD values of each voxel and if the voxels had intensities which were highly correlated and the voxels were in two different ROIs then the two ROIs were said to be connected.

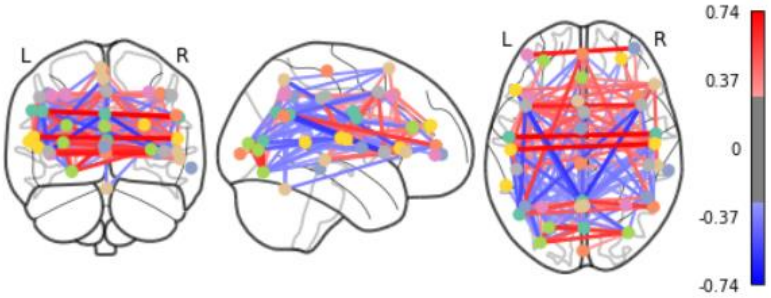


Figure 19. Connectivity between different parts of the brain when the HO atlas is applied to define ROIs.

5.4 3D coordinates of the ROIs

The rsfMRI data was converted to the Montreal Neurological Institute 152 (MNI152) space from the voxel space to perform the various pre-processing steps. The rsfMRI data was converted back into voxel space to extract the x, y and z coordinates of the points representing each ROI segmented using an atlas. Once these coordinates are obtained they are used to restrict the number of outgoing edges that a vertex has.

The coordinate plane is divided into 8 parts and the x, y and z coordinates are used to check which of the 8 sections does the ROI points resides in. After which the distance of all the points from a vertex is calculated and then depending on the number of edges per vertex declared the nearest *k* neighbors are then considered connected to the vertex in question. The value of the edge that connects two vertices represents the distance between them in the baseline model.

Chapter 6

Model

6.1 Baseline model

6.1.1 Graph Convolution

The Graph CNN model used in this thesis was proposed by Dominguez et al [29]. It is a variation of the Petroski Such et al. [30] model. This model is a spatial graph convolution model and it uses the *x, y* and *z* coordinates of the vertices to extract spatial features. The input graph is a function of vertices and edges: $G = (V, A)$, where $A \in R^{N \times N \times m}$ is the adjacency matrix with N number of vertices in the graph and m are the number of slices of adjacency matrix used to represent direction. $V \in R^{N \times F}$ is the vertex matrix with F vertex features present in each vertex of the graph. The model proposed by Petroski Such et al. [30] used shift invariant convolutional filters to represent a polynomial of the adjacency matrix where an increase in the degree of the adjacency matrix represented the neighbors' k hop away from the corresponding vertex. The equation (4) is used to represent this.

$$H = h_0 I + h_1 A + h_2 A^2 + \cdots + h_k A^k \qquad H \in R^{\mathrm{N x N}} \tag{6}$$

The value of A represents the immediate neighbors, A^2 represents the two-hop neighbors and A^k represents the k hop neighbors. The *hi* filter is an isotropic filter used to extract features from all the vertices at the same distance from their corresponding vertex. Dominguez et al. [29] only used the information regarding self-connected vertices and one-hop neighbors. These connections were used as the model should learn higher order polynomials through the multiple convolutions performed on it.

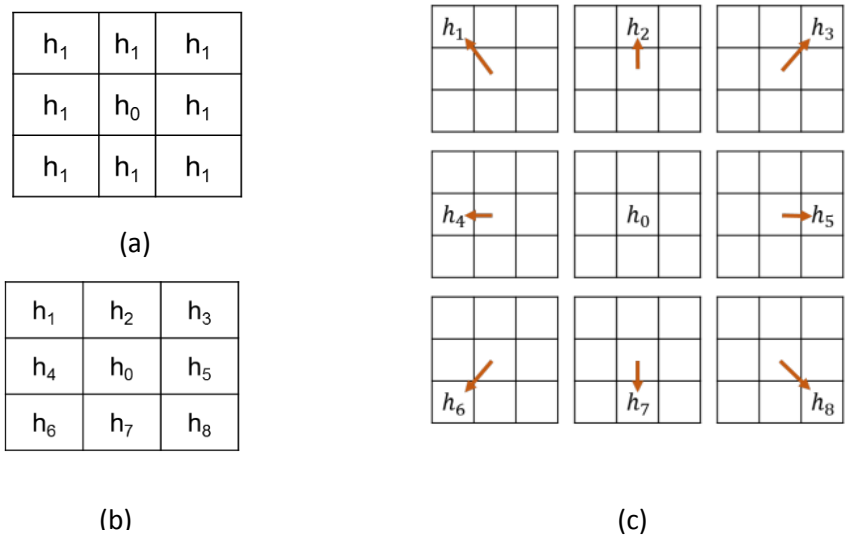


Figure 20 (a). 1-hop graph convolution filter [30].

Figure 20 (b). Standard 3×3 convolution filter [30].

Figure 20 (c). The nine different edge connections combined to form a 3 × 3 filter to analyze direction [30].

6.1.2 Graph pooling

This model also performs graph pooling to reduce the graph structure by reducing the size of the adjacency matrix and reduce the graph signal by reducing the vertex matrix, as in equations (5) and (6).

$$A_m^{reduced} = P^T A_m P \; for \; each \; m \; \in \{1, 2, \dots, M\} \tag{7}$$

$$V^{reduced} = P^T V \tag{8}$$

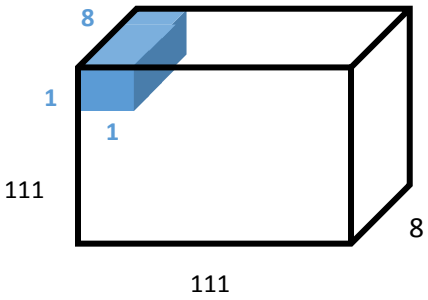
The non-zero entries in the column of matrix P indicates the clusters to which the corresponding vertices of the original graph belong. Then each cluster is represented by a single vertex value. This clustering algorithm analyzes the connection between vertices and not the values of the vertices.

6.2 Edge feature model

6.2.1 Adjacency convolution

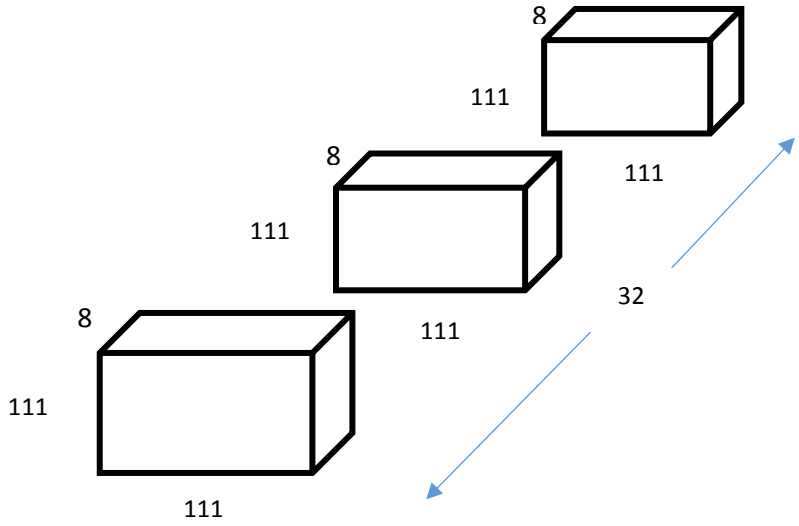
The adjacency matrix contains the information about the connectivity between different ROIs in the scan. This connectivity information is very important as the difference in the connectivity between various regions helps us to distinguish between subject with and without ASD.

A filter of shape $1\times1\times m$ is applied to the adjacency matrix $A \in R^{N\times N\times m}$ to extract spatial information. This filter is used to perform 3D convolution over the adjacency matrix to calculate the cross-correlation between connectivity values of the vertices. The output is of the shape $N\times N\times m\times D$, where D is the number of filters instead of $N\times N\times 1\times D$ as the padding variable was set to 'same' instead of 'valid'. This output is then collapsed to create a new adjacency matrix with the same shape as the input adjacency matrix or it is collapsed to create new features for the vertices in the graph. If the output of the adjacency convolution layer is used to create new features, then the output is summed along the non-required dimensions to produce a matrix of shape $N\times D$. The result of the summation is then concatenated with the vertex matric $V \in R^{N\times F}$ to produce a vertex of shape $N\times(F+D)$. If the output of the adjacency convolution layer is summed along the depth, then it will create a new adjacency matrix with the same dimension as the old one.



(a)

Figure 21 (a). Adjacency convolution layer with filter of dimensions (1×1×m), where m = number of slices and a new adjacency matrix as the output.



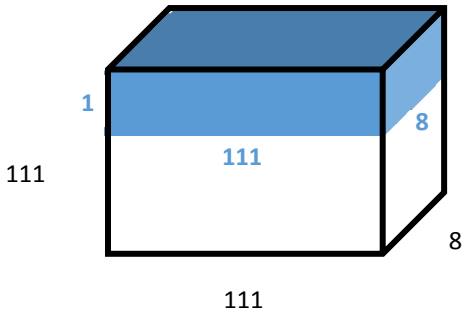
(b)

Figure 21 (b). The output of the adjacency convolution layer with the filter shown in Figure 21 (a).

6.2.2 Variation of adjacency convolution

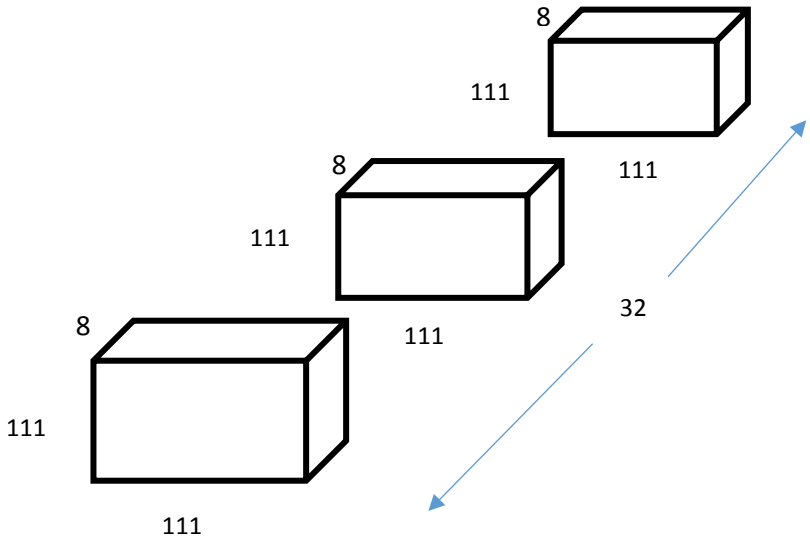
Features are extracted from the adjacency matrix in a manner similar to the one used in the adjacency convolution layer. However, here a filter of shape *1×N×m* is used to perform 3D

convolution over the adjacency matrix. The idea here is to analyze the connectivity information of one vertex with respect to all its connections vertices.



(a)

Figure 22 (a). Adjacency convolution layer with filter of dimensions (1×N×m), where N = number of vertices and m = number of slices and a new adjacency matrix as the output.



(b)

Figure 22 (b). The output of the adjacency convolution layer with the filter shows in Figure 22 (a)

6.3 Temporal model

The Graph CNNs used earlier would reduce the vertex feature information to the number of filters applied. This could cause the loss of valuable information. Therefore, in order to retain the vertex feature information, a new Graph CNN model was created called temporal graph convolutional neural network model. The temporal model has a block containing two layers defining one graph convolution layer. The block contains one convolution layer and one graph convolution layer. The standard convolution layer first expands the dimension of the input vertex matrix to create a vertex matrix of dimensions $N \times F \times 1$. Then standard convolution is performed to create a vertex matrix of the shape $N \times F \times F_t$ where, F_t is the number of temporal features. This vertex matrix is then passed through a modified graph convolution layer. This layer uses the *x, y* and *z* coordinates of the vertices to select which vertices to connect and the connectivity values are calculated using the atlas as the edge values. Then the modified graph convolution layer performs graph convolution on the new third dimension, F_t of the vertex matrix. Therefore, we do not lose any vertex features but only extract information.

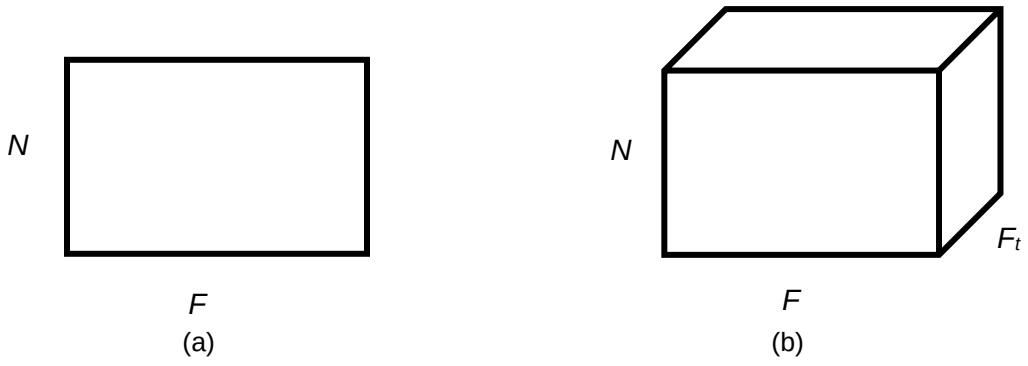


Figure 23 (a). The input vertex matrix passed into the standard convolution layer.
Figure 23 (b). The output vertex matrix obtained from the standard convolution layer. It is used as the input for the temporal graph convolution layer.

The adjacency convolution layer was combined with this new temporal graph convolutional neural network model. Adding this layer to the temporal model allowed additional extraction of information from the connectivity matrix obtained from analyzing the time series information of the scans. The adjacency convolution layer was used with a filter of shape $1 \times N \times I$ and the output matrix was summed along its dimensions such that we would obtain new features. These new features were concatenated to the three dimensional vertex matrix of shape $N \times F \times F_t$ to create a new vertex matrix of shape $N \times (F+D) \times F_t$, where D is the number of filters applied in the adjacency convolution layer .

Chapter 7

Dataset

The dataset contains information about 1035 subjects collected from 17 different international sites out of which 539 subjects have ASD and 573 subjects are Neurotypical. The subjects include 948 men and 164 women between the ages of 6 to 64 at the time their scans were captured. The dataset provides the sMRI and the rsfMRI data of each subject as well as several other features regarding the subjects. Some of these features are: the site where the scans were taken, the subject identification number, the sex of the subject, the age of the subject at the time the data was captured, the dominant hand of the subject, scores of different tests given to the subjects and so on. All of the information which provides additional knowledge on the subjects are provided in a csv file.

The sMRI and the rsfMRI data for each subject is provided in the format of ‘rest.nii.gz’ as well as ‘anatomical.nii.gz’. The time series information captured in each rsfMRI data is different as the length of time series depends on the site where the data was captured. Along with a variation in the length of the time series information there are other variations observed in the MRI scans captured at different sites.

The dataset was randomly split in the ratio 9:1 to create train and test splits respectively. A single random split was used to run all the experiments. In order to obtain the margin of error for the experiments two additional unique random splits of the ratio 9:1 were created. These splits were then used to rerun the experiments and calculate the margin of error.

Chapter 8

Results

The Dominguez et al. [29] model was run to set the baseline for this experiment. The model was run with the different variations of the adjacency convolution layer and it was modified to implement the time domain version as well. In addition to this, an extensive ablation analysis was also carried out to find the right architecture and phenotypic features. The results for all of these experiments along with their architecture and various parameters used are given below:

8.1 Baseline results

Architecture of the model: two graph convolution layers of depth four, one graph convolution layer of depth eight, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). The train batch size was 200, test batch size 105 (entire test dataset) with the learning rate starting at 0.01 with a decay rate of 0.25, the learning rate was updated after every 200 iterations and the number of edges per vertex was 25. The model was trained for 2500 iterations.

The model was tested to evaluate if the sparse version of the Dominguez et al. [29] model performed better or a non-sparse version:

Figure 24. The model used to obtain the baseline results.

Table 1: Baseline results with and without sparsity.

Model	Features	Accuracy
Sparse baseline	Age, sex	47.115 ± 0.965
Non-sparse baseline	Age, sex	48.0769 ± 0.619

The model was run with the adjacency values changed from using the distance information to the connectivity information. The non-sparse version of the model was used.

Table 2: The baseline model is with adjacency matrix elements as the distance between the vertices. The baseline connectivity model is the baseline model with the adjacency matrix elements as connectivity values instead of distance between vertices. Here 25 is the number of vertices connected to each vertex.

Model	Features	Accuracy
Non-sparse baseline	Age, sex	48.0769 ± 0.619
Baseline connectivity-25	Age, sex	51.08 ± 0.307

8.2 Number of neighbors per vertex

Architecture of the model: two graph convolution layers of depth four, one graph convolution layer of depth eight, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). The train batch size was 200, test batch size 105 (entire test dataset) with the learning rate starting at 0.01 with a decay rate of 0.25, the learning rate

was updated after every 200 iterations and the number of edges per vertex varied. The model was run with the adjacency values changed from using the distance information to the connectivity information. The non-sparse version of the model was used and it was trained for 2500 iterations.

The experiment was to test to check the number of vertices each vertex should be connected to when the connectivity information was used as the values for the adjacency matrix and sparsity was set to false.

Table 3: The baseline connectivity model to explore the effects of increasing the number of vertices connected to each vertex. Here the number of vertices connected to a vertex is changed to 55 and 111.

Model	Features	Accuracy
Baseline connectivity-55	Age, sex	52.58
Baseline connectivity-111	Age, sex	54.47 ± 0.803

8.3 Phenotypic features results

Architecture of the model: two graph convolution layers of depth four, one graph convolution layer of depth eight, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). The train batch size was 200, test batch size 105 (entire test dataset) with the learning rate starting at 0.01 with a decay rate of 0.25, the learning rate get updated after every 200 iterations and the number of edges per vertex was 111. The model was run with the adjacency values changed from using the distance information to the connectivity information. The non-sparse version of the model was used and it was trained for 2500 iterations.

This experiment evaluates the effect of the fiq, piq and viq phenotypic features on the model.

Table 4: The baseline connectivity model to explore the effects of different phenotypic features. Here the number of vertices connected to a vertex is 111 and the features experimented with are full scale IQ (fiq), verbal IQ (viq) and performance IQ (piq).

Model	Features	Accuracy
Baseline connectivity-111	Age, sex, fiq	49.04
Baseline connectivity-111	Age, sex, piq	52.96
Baseline connectivity-111	Age, sex, viq	49.04
Baseline connectivity-111	Age, sex, fiq, piq, viq	56.87 ± 0.979

8.4 Adjacency convolution results

Architecture of the model: adjacency convolution layer with four filters, graph convolution layers with four filters, adjacency convolution layer with eight filters, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). The train batch size was 200, test batch size 105 (entire test dataset) with the learning rate starting at 0.01 with a decay rate of 0.25, the learning rate get updated after every 200 iterations and the number of edges per vertex was 111. The model was run with the adjacency values changed from using the distance information to the connectivity information. The non-sparse version of the model was used and it was trained for 2500 iterations.

The experiment explored the two different types of filters as well as the effect of changing the adjacency matrix and concatenating features to the vertex matrix.

Figure 25. The model used to analyze the different variations of the adjacency convolution layer.

8.4.1 Filter of dimension 1×N×m

Table 5: The adjacency convolution layer along with the baseline connectivity model to explore the effects of new adjacency matrix and new features concatenated to the vertex matrix. *F1* is filter of shape (1×*N*×*m).*

Model	Features	Accuracy
Adjacency convolution f1 new matrix	Age, sex, fiq, piq, viq	60.41
Adjacency convolution f1 new features	Age, sex, fiq, piq, viq	65.18 ± 1.079

8.4.2 Filter of dimension 1×1×m

Table 6: The adjacency convolution layer along with the baseline connectivity model to explore the effects of new adjacency matrix and new features concatenated to the vertex matrix. *F2* is filter of shape (1×*1*×*m).*

Model	Features	Accuracy
Adjacency convolution f2 new matrix	Age, sex, fiq, piq, viq	55.86
Adjacency convolution f2 new features	Age, sex, fiq, piq, viq	56.94

8.5 Temporal model

Architecture of the model: convolution layer with four filters to perform convolution on the vertex matrix, temporal graph convolution layers with four filters, convolution layer with eight filters to perform convolution on the vertex matrix, temporal graph convolution layers with eight filters, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). The train batch size was 200, test batch size 105 (entire test dataset) with the learning rate starting at 0.01 with a decay rate of 0.25, the learning rate get updated after every

200 iterations and the number of edges per vertex was 111. The model was run with the adjacency values changed from using the distance information to the connectivity information. The non-sparse version of the model was used and it was trained for 2500 iterations. This architecture along with the described parameters provided an accuracy of **63.76 ± 1.29 %**.

TGC = Temporal Graph Convolution layer

FC = fully connected layer

Figure 26. The temporal model used on the ABIDE-I dataset.

The adjacency convolution layer is added to the temporal Graph CNN model and the following architecture is used: convolution layer with four filters to perform convolution on the vertex matrix, temporal graph convolution layers with four filters, adjacency convolution layer with four filters and the new features are concatenated to the vertex matrix, convolution layer with eight filters to perform convolution on the vertex matrix, temporal graph convolution layers with eight filters, adjacency convolution layer with eight filters and the new features are concatenated to the vertex matrix, one fully connected layer of depth 32 and last fully connected layer of depth two (number of classes). All the other parameters were kept constant. This architecture provided an accuracy of **66.37 ± 0.82 %**.

AC = Adjacency Convolution layer

TGC = Temporal Graph Convolution layer

FC = fully connected layer

Figure 27. The combination of temporal graph convolution and adjacency convolution layer used on the ABIDE-1 dataset.

In all the experiments executed the data was preprocessed using the HO atlas. In order to evaluate how well the model would perform on another atlas, the data is preprocessed using the CC400 atlas. This preprocessed data is then used as an input for our best model: the temporal model combined with the adjacency convolution layer. We obtained an accuracy of **69.15%.** The output from the model shown in Figure 25 for the data pre-processed with the HO and the CC400 atlas was used to perform voting. Combing the results from these two different atlases increased the accuracy to **70.23%**.

8.6 Comparisons with other models

Khosla et al. [18] preprocessed the raw rsfMRI data to extract voxel maps and create connectivity fingerprints for each atlas. They then used the connectivity fingerprint as an input to a 3D CNN model and trained the model to predict ASD. Khosla et al. [18] obtained the state of the art result of 73.30% by using an ensemble of all the atlases.

Dvornek et al. [36] used the raw time series information from preprocessed rsfMRI data and a Long Short Term Memory (LSTM) model. The LSTM model was able to predict ASD with an accuracy of 70.1%. Aghdam et al. [38] combined the time series information obtained from the preprocessed rsfMRI with the gray and white matter information obtained from the sMRI data.

They obtained an accuracy of 64.13% when this data was passed through a linear SVM. Nielsen et al. [37] calculated the connectivity information from preprocessed rsfMRI data. The connectivity matrix along with a few phenotypic features like age, gender, handedness and the site where the data was collected were used as inputs. Using the site where the data was collected does not let the model generalize well, so it is preferred to learn features independent of the site. This is important as the data captured from different sites has multiple differences. Nielsen et al. [37] were able to predict ASD with only 59.6 accuracy.

Table 7: Comparing the best result of this research with other works

Model	Features	Accuracy
3D CNN [18]	rsfMRI	73.30
Long Short Term Memory (LSTM) [36]	rsfMRI	70.1
Support Vector Machine (SVM) [38]	rsfMRI, gray matter and white matter	64.13
Radial basis function kernel (RBF) SVM [37]	rsfMRI, age, sex, handedness, fiq, piq, viq and site	59.6
Temporal + Adjacency convolution f1 new features +voting (ho and cc400 atlas)	rsfMRI, age, sex, fiq, piq and viq	70.23

Chapter 9

Conclusion

ASD, being a neurological disorder can be predicted by analyzing the connectivity information between different parts of the brain. A Graph CNN model analyzed the graph created by using the ABIDE-I dataset to detect ASD in subjects. The model utilized the spatial information of the ROIs as well as the edge value connecting them. In addition to these features, the time series information and the phenotypic features also helped in improving the ability of the Graph CNN model to detect ASD in subjects.

When the Dominguez et al. [29] model was modified to use the connectivity information in place of the distance information in the adjacency matrix the accuracy improved. When the number of edges per vertex was defined to be equal to the number of ROIs present in the atlas, we further saw an improvement in the results.

Adding multiple phenotypic features helped to provide more data regarding each subject. When added individually to the model, these features did not show any substantial improvements. However, when they were all added to the vertex matrix together they showed an improvement in the results. The result obtained was 56.87 ± 0.979%.

The adjacency convolution layer had four different possible variations. It was seen that when the filter of shape *1xNxl* was used to perform the convolution operation and the output of the convolution operation was summed to create new features, the result obtained was 65.18 ± 1.079.

The temporal model performed graph convolutions on the time series information and obtained an accuracy lower than the Dominguez et al. [29] model with the best configuration of the adjacency convolution layer. However, when the temporal model was combined with the

adjacency convolution layer, an accuracy of 66.37 ± 0.82% was achieved for data preprocessed using the HO atlas.

The results obtained by passing the data preprocessed using the HO atlas and the CC400 atlas through the temporal graph convolution model increased the accuracy to 70.23%.

Chapter 10

Future Work

There are many steps that can be taken to improve the current results using the Graph CNN. One major problem that was brought to light with this research was that the size of the dataset was too small to train even a slightly deep neural network. Performing data augmentation will increase the dataset size which in turn will help improve the results.

As each atlas may segment different parts of the brain in different sizes, combining the information from all the different atlases will help increase the information present for each subject. It will also help analyze the connectivity between various regions without missing any data.

Analyzing the connectivity between each voxel in the brain without using an atlas will help create a graph where no information is lost due to averaging.

References

[1] Craddock, Cameron and Benhajali, Yassine and Chu, Carlton and Chouinard, Francois and Evans, Alan and Jakab, Andrs and Khundrakpam, Budhachandra Singh and Lewis, John David and Li, Qingyang and Milham, Michael and Yan, Chaogan and Bellec, Pierre. The Neuro Bureau Preprocessing Initiative: open sharing of preprocessed neuroimaging data and derivatives, Frontiers in Neuroinformatics, ISSN: 1662-5196.

[2] Smith, Jason F and Hur, Juyoen and Kaplan, Claire M and Shackman, Alexander J. The Impact of Spatial Normalization for Functional Magnetic Resonance Imaging Data Analyses Revisited, bioRxiv 2018, doi: 10.1101/27230.

[3] E. Roura, A. Oliver, M. Cabezas, J.C. Vilanova, A. Rovira and Ll. Ramió-Torrentà, X. Lladó. MARGA: Multispectral adapting region growing algorithm for brain extraction on axial MRI. Computer Methods and Programs in Biomedicine, 115(3), pp. 147-161. 2014.

[4] Beeck, Hans P. Op de. “Against hyperacuity in brain reading: Spatial smoothing does not hurt multivariate fMRI analyses?” NeuroImage 49 3 (2010): 1943-8.

[5] Makris N, Goldstein JM, Kennedy D, Hodge SM, Caviness VS, Faraone SV, Tsuang MT, Seidman LJ. Decreased volume of left and total anterior insular lobule in schizophrenia. Schizophr Res. 2006 April 83(2-3):155-71.

[6] Frazier JA, Chiu S, Breeze JL, Makris N, Lange N, Kennedy DN, Herbert MR, Bent EK, Koneru VK, Dieterich ME, Hodge SM, Rauch SL, Grant PE, Cohen BM, Seidman LJ, Caviness VS, Biederman J. Structural brain magnetic resonance imaging of limbic and thalamic volumes in pediatric bipolar disorder. Am J Psychiatry. 2005 Jul;162(7):1256-65.

[7] N. Tzourio-Mazoyer, B. Landeau, D. Papathanassiou, F. Crivello, O. Étard, N. Delcroix, B. Mazoyer, and M. Joliot. Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain. NeuroImage 2002. 15: 273-289.

[8] Rolls ET, Joliot M & Tzourio-Mazoyer N. Implementation of a new parcellation of the orbitofrontal cortex in the automated anatomical labeling atlas. (2015). NeuroImage.

[9] Petroski Such, Felipe & Sah, Shagan & Dominguez, Miguel & Pillai, Suhas & Zhang, Chao & Michael, Andrew & Cahill, Nathan & Ptucha, Raymond. (2017). Robust Spatial Filtering

with Graph Convolutional Neural Networks. IEEE Journal of Selected Topics in Signal Processing. PP. 10.1109/JSTSP.2017.2726981.

[10] Abraham, Alexandre and Pedregosa, Fabian and Eickenberg, Michael and Gervais, Philippe and Mueller, Andreas and Kossaifi, Jean and Gramfort, Alexandre and Thirion, Bertrand and Varoquaux, Gael. Machine learning for neuroimaging with scikit-learn. (2014). Frontiers in Neuroinformatics.

[11] Baio J, Wiggins L, Christensen DL, et al. Prevalence of Autism Spectrum Disorder Among Children Aged 8 Years — Autism and Developmental Disabilities Monitoring Network, 11 Sites, United States, 2014. MMWR Surveill Summ 2018; 67 (No. SS-6):1–23.

[12] Autism Society. (2018). *Facts and Statistics - Autism Society*. [online] Available at: http://www.autism-society.org/what-is/facts-and-statistics/.

[13] P. Such, S. Sah, M. A. Dominguez, S. Pillai, C. Zhang,A. Michael, N. D. Cahill, and R. Ptucha. Robust spatial fil-tering with graph convolutional neural networks. IEEE Journal of Selected Topics in Signal Processing, 11(6):884–896, Sept 2017.

[14] J. Bruna et al. Spectral networks and locally connected networks on graph, arXiv: 13126203, 2013.

[15] N. Kipf, Max Welling. Semi-Supervised Classification with Graph Convolutional Networks. ICLR 2017.

[16] Kawahara, J., Brown, C. J., Miller, S. P., Booth, B. G., Chau, V., Grunau, R. E., Zwicker, J. G., and Hamarneh, G. (2017). BrainNetCNN: Convolutional neural networks for brain networks; towards predicting neurodevelopment. NeuroImage, 146(July), 1038–1049.

[17] Kawahara, J. (2018). *jeremykawahara/ann4brains*. [online] GitHub. Available at: https://github.com/jeremykawahara/ann4brains.

[18] M. Khosla, K. Jamison, A. Kuceyeski, M. Sabuncu. 3D Convolutional Neural Network for Classification of Functional Connectomes. arXiv: 1806.04209, June 2018.

[19] Heinsfeld, A. S., Franco, A. R., Craddock, R. C., Buchweitz, A., & Meneguzzi, F. (2017). Identification of autism spectrum disorder using deep learning and the ABIDE dataset. *NeuroImage. Clinical*, *17*, 16-23. doi:10.1016/j.nicl.2017.08.017.

[20] A. Di Martino, C. G. Yan, Q. Li, E. Denio, F. X. Castellanos, K. Alaerts, J. S. Anderson, M. Assaf, S. Y. Bookheimer, M. Dapretto et al., “The autism brain imaging data exchange: towards a large-scale evaluation of the intrinsic brain architecture in autism,” Mol Psychiatry, vol. 19, no. 6, pp. 659-67, 2014.

[21] Craddock C, Sikka S, Cheung B, Khanuja R, Ghosh SS, Yan C, Li Q, Lurie D, Vogelstein J, Burns R, Colcombe S, Mennes M, Kelly C, Di Martino A, Castellanos FX and Milham M (2013). Towards Automated Analysis of Connectomes: The Configurable Pipeline for the Analysis of Connectomes (C-PAC). Front. Neuroinform. Conference Abstract: Neuroinformatics 2013.

[22] C. Craddock, Y. Benhajali, C. Chu, F. Chouinard, A. Evans, A. s. Jakab, B. S. Khundrakpam, J. D. Lewis, Q. Li, M. Milham et al., "The Neuro Bureau Preprocessing Initiative: open sharing of preprocessed neuroimaging data and derivatives," Frontiers in Neuroinformatics, 2013.

[23] Harvard-Oxford cortical and subcortical structural atlases, http://fsl.fmrib.ox.ac.uk/fsl/fslwiki/Atlases.

[24] Miykael.github.io. (2018). *Nipype Beginner's Guide — All you need to know to become an expert in Nipype*. [online] Available at: http://miykael.github.io/nipype-beginner-s-guide/faq.html#coregistration.

[25] Ogawa S., Lee T.M., Kay A.K. and Tank D.W., (1990) Brain Magnetic Resonance Imaging with Contrast Dependent on Blood Oxygenation Proc. Natl. Acad. Sci. (USA), 87, 9868-9872.

[26] Nakamura, K. (2013). *Structural Image Preprocessing*. [ebook] Available at: http://www.bic.mni.mcgill.ca/uploads/Seminars/2013-09-23_-_Structural_Image_Preprocessing_-_Kunio_Nakamura.pdf.

[27] Abraham Alexandre, Pedregosa Fabian, Eickenberg Michael, Gervais Philippe, Mueller Andreas, Kossaifi Jean, Gramfort Alexandre, Thirion Bertrand, Varoquaux Gael. Machine learning for neuroimaging with scikit-learn, Frontiers in Neuroinformatics, 2014.

[28] En.wikipedia.org. (2018). *Spatial normalization*. [online] Available at: https://en.wikipedia.org/wiki/Spatial_normalization.

[29] M. Dominguez, R. Dhamdhere, A. Petkar, S. Jain, S. Sah, R. Ptucha. General purpose deep point cloud feature extractor. Winter Conference on Applications of Computer Vision, 2018.

[30] F. P. Such, S. Sah, M. A. Dominguez, S. Pillai, C. Zhang,A. Michael, N. D. Cahill, and R. Ptucha. Robust spatial fil-tering with graph convolutional neural networks. IEEE Jour-nal of Selected Topics in Signal Processing, 11(6):884–896, Sept 2017.

[31] Gorgolewski K, Burns CD, Madison C, Clark D, Halchenko YO, Waskom ML, Ghosh SS. (2011). Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in Python. Front. Neuroimform. 5:13.

[32] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, “Spectral Networks and Locally Connected Networks on Graphs,” *arXiv:1312.6203 [cs]*, Dec. 2013J.

[33] M. Niepert, M. Ahmed, and K. Kutzkov, “Learning convolutional neural networks for graphs,” in *International conference on machine learning*, 2016, pp. 2014–2023.

[34] T. N. Kipf and M. Welling, “Semi-Supervised Classification with Graph Convolutional Networks,” in *International Conference on Learning Representations (ICLR)*, 2017.

[35] Z. Yao, B. Hu, Y. Xie, P. Moore, J. Zheng “A review of structural and functional brain networks: small world and atlas”, Brain Informatics, March 2015J.

[36] Nicha C. Dvornek, Pamela Ventola, James S. Duncan. Combining phenotypic and resting-state fMRI data for autism classification with recurrent neural networks. 2018, IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018).

[37] Nielsen, J. A., Zielinski, B. A., Fletcher, P. T., Alexander, A. L., Lange, N., Bigler, E. D., Lainhart, J. E., Anderson, J. S. (2013). Multisite functional connectivity MRI classification of autism: ABIDE results. Frontiers in human neuroscience, 7, 599. doi:10.3389/fnhum.2013.00599.

[38] Akhavan Maryam, Sharifi Arash, Pedram Mohsen. Combination of rs-fMRI and sMRI Data to Discriminate Autism Spectrum Disorders in Young Children Using Deep Belief Network. 2018, Journal of Digital Imaging.