



FAKULTÄT FÜR
INFORMATIK

Digital Engineering

Source Level Brain Connectome Based Visual Field Prediction with 3D Convolutional Neural Networks

Master Thesis

Chao Yan

Supervisor: Prof. Dr.-Ing. Andreas Nürnberger

Advisors: M.Sc. Jiahua Xu

M.Sc. Zheng Wu

September 21, 2020

Abstract

Stroke is a frequently seen disease, and it brings various complications: speech problems, visual problems, paralysis, and more other memory problems. According to the lesion location in the brain, the visual problems also vary. After some time with the proper treatment, the visual loss problem could be restored to some extent.

Based on the brain connectome on the source level, this thesis mainly analyses the correlation between the brain connectome and stroke patients' vision performance using deep learning technology. Multiple deep learning models were proposed and applied to explore several key questions. Given the current brain connectome data, the following results are proved. Firstly, it is possible to discriminate between patients and healthy controls with an accuracy of 72%. Subsequently, the possibility possible to predict whether a stroke patient can have notorious recovery in vision is 95%. Next, the accuracy of predicting vision distribution is 71.9%. However, it is challenging to predict vision percentage conditions.

Acknowledgement

Firstly, I would sincerely like to thank Prof. Dr.-Ing. Andreas Nürnberger for giving me the opportunity and the support to write my master thesis in his group.

Subsequently, I would also like to express my thanks to my advisor M.Sc. Jiahua Xu for his help in many ways, his successive guidance, patience and considerate encouragement during my whole master thesis period. I would like to thank M.Sc. Zheng Wu, for her feedback on the thesis as the second advisor.

Finally, I would like to thank my friends and family for their support throughout my whole master study.

Contents

List of Acronyms	v
List of Figures	viii
List of Tables	xi
1 Introduction	1
1.1 Motivation	1
1.2 Research Questions	1
1.3 Structure of the work	2
2 Background Knowledge	3
2.1 Stroke	3
2.2 Visual System	3
2.3 Brain Connectome	5
2.4 Electroencephalogram (EEG)	7
2.5 Deep Learning	11
2.5.1 Three Typical Algorithms	11
2.5.2 Multilayer Perceptron Model (MLP)	12
2.5.3 Gradient Descent Optimization Algorithms	16
2.5.4 Loss Functions	17
2.6 Convolutional Neural Network (CNN)	19
2.6.1 The Convolutional Layer	20
2.6.2 The Pooling Layer	21
2.6.3 Classic CNN Model: VGG16	22
2.6.4 3D CNN	23
2.7 Graph Neural Network	24
2.7.1 Spectral Methods	25
2.7.2 Spatial Methods	26
3 Related Work	28

4	Methods	34
4.1	Experiment Setting and Input Data Collecting	34
4.2	Output Data Collecting	35
4.3	Data Processing	36
4.4	Research Questions	41
4.4.1	Question No. 1	41
4.4.2	Question No.2	42
4.4.3	Question No.3	43
4.4.4	Question No.4	54
5	Discussion	60
6	Conclusion	63

List of Acronyms

tACS	Transcranial Alternating Current Stimulation
tDCS	Transcranial Direct-Current Stimulation
LGN	Lateral Geniculate Nucleus
HH	Homonymous Hemianopia
DTI	Diffusion Tensor Magnetic Resonance Imaging
CNN	Convolutional Neural Network
BOLD-fMRI	Blood-Oxygen-Level-Dependent Imaging-Functional Magnetic Resonance Imaging
EEG	Electroencephalogram
MEG	Magnetoencephalography
DNN	Deep Neural Network
AI	Artificial Intelligence
ML	Machine Learning
DL	Deep Learning
SL	Supervised Learning
GAN	Generative Adversarial Network
GPU	Graphics Processing Unit
UL	Unsupervised Learning
LSTM	Long-short Term Memory
SSL	Semi-Supervised Learning
MLP	Multilayer Perceptron

SGD Stochastic Gradient Descent

Adam Adaptive Moment Estimation

MSE Mean Squared Errors

MAE Mean Absolute Error

VGG16 Visual Geometry Group 16

RPN Region Proposal Networ

GNN Graph Neural Network

RNN Recurrent Neural Network

ReLU Rectified Linear Unit

ResNet Residual Neural Network

AM Adjacency Matrix

AD Alzheimer’s Disease

DLB Dementia with Lewy Bodie

SVM Support Vector Machine

MDC-CNN Multi-Domain Connectom CNN

SZ Schizophrenia

ASD Autism Spectrum Disorder

TC Typical Controls

ROIs Regions Of Interests

AUC Area Under Receiver Operating Characteristic Curve

s-GCN Siamese Graph Convolutional Neural Network

dFC dynamic Functional Connectivity

ALFF Amplitude of Low-Frequency Fluctuation

ReHo Regional Homogeneity

VAE Variational Autoencoder

ALS Amyotrophic Lateral Sclerosis

AE Autoencoder

DAFA Deep Neural Network with Autoencoder Pre-trained Functional Connectivity Analysis

HMM Hidden Markov Model

ADHD Attention-Deficit / Hyperactivity Disorder

ICA Independent Component Analysis

AAL Automatic Anatomical Labeling

FCN Functional Connectivity Network

HRP High Resolution Perimetry

List of Figures

1	The visual system [1]	4
2	The normal vision [74]	4
3	Left HH vision [74]	4
4	Some common cases of hemianopia [49]	5
5	EEG/MEG, fMRI, DTI source of network [47]	6
6	As seen in the image, the EEG electrode needs to measure the signal through many tissue layers (such as the scalp, skull, dura, mater, arachnoid subarachnoid space). Synaptic currents in cells (golden pyramidal shape) can generate electrical fields (voltage) as signals. Only when signals are accumulated to be high enough, they can be detected [21][20].	8
7	An example of scalp EEG [16]	9
8	General structure of the brain [3]	9
9	General structure of the cerebrum [3]	9
10	An example of different types of normal EEG rhythms in a second [61].	10
11	Relationships among AI, ML and DL	12
12	General examples of SL model and UL model	12
13	An example of MLP structure	13
14	ReLU Activation Function	14
15	Sigmoid Activation Function	14
16	Tanh Activation Function	15
17	Tracking Comparison between with and without momentum	17
18	A simple CNN model. There are seven layers in this model which are three convolutional layers and three fully connected layers. The last activation function is a Softmax.	19
19	An example of how the convolution operation calculation works [69].	20

20	The convolution operation The operation starts with the filter in the top left corner of the input grid and calculates the dot product between this filter and the input grid; a single value, as a result, is added to the matching position in the feature map. Next, slide the filter one step (one grid) to the right (the same color indicating the matching filter and result) and repeat the same operation and so on until the entire area of the input layer is gone through from left to right and top to bottom [69].	21
21	An practical illustration about the feature maps [69].	22
22	Left one is the max pooling operation, right one is the average pooling operation.	22
23	VGG16 structure [69]	23
24	An example of 3D convolutional operation: The input data is with a shape of $[12, 12, n, 3]$, and one filter size is $[3, 3, 3]$ (always keep the number of color channels in filters the same with that in input data). The cube filter (the yellow one) will do a dot product with a matching cube in input data (the red one) and get a new value (green one) in the following feature map. Here it is done with the same-padding operation, which maintains the original size of the input data.	24
25	The architecture of the model proposed in [71].	28
26	The architecture of the model proposed in [48].	29
27	The architecture of the model proposed in [33].	29
28	The architecture of the model proposed in [36].	30
29	The architecture of the model proposed in [80].	31
30	The architecture of the model proposed in [35].	32
31	The architecture of the model proposed in [42].	33
32	An example of health control's coherence connectivity network	36
33	A woman is taking a vision test [12]	37
34	Patient ID = 1, pre-treatment stage, frequency = 1Hz	38
35	Patient ID = [2, 13, 17], pre-treatment stage	40
36	Patient ID = [2, 13, 17], post-treatment stage	40
37	Patient ID = [2, 13, 17], follow-up stage	40
38	Loss and Accuracy during training process of the proposed model for hypothesis 1	42

39	Loss and Accuracy during training process of the proposed model for hypothesis 2	43
40	Swap operation of the matrix data	44
41	Swap operation of the matrix data	45
42	The architecture of the 1D CNN model	46
43	Loss and MAE during training process for 1D CNN model.	46
44	Comparisons of three vision conditions between predictions and ground truth on testing data.	47
45	Comparisons of three vision conditions between predictions and ground truth for on training data.	48
46	Loss and MAE during training process. Each model has a unique color representation as it is shown in this figure. The solid line indicates the loss (MSE), and the dotted line indicates the MAE.	50
47	Comparisons of three vision conditions between predictions and ground truth for different models on testing data.	51
48	Comparisons of three vision conditions between predictions and ground truth for VGG16 model and Xception model on training data.	52
49	Performances of different cases of 2D CNN models.	53
50	Performances of different cases of 3D CNN models.	54
51	A visual comparison of the original and new matrices. The red line on the left segmented the field into 4 areas, which are corresponding 4 areas on the right.	55
52	Loss and accuracy during training process. Each model has a unique color representation as it is shown in this figure. The solid line indicates the loss (binary cross-entropy), and the dotted line indicates the accuracy.	56
53	An example of the best performance model.	57
54	Performances of different cases of 2D CNN models.	58
55	Performances of different cases of 3D CNN models.	59
56	Architecture of the proposed model.	59
57	Coherence of the same frequency band in two trials of the same patient in pre-treatment stage (First row: 1Hz; Second row: 10 Hz).	61
58	Visualization of feature maps of the proposed 3D CNN model.	62

List of Tables

1	A table showing the lobes and corresponding functions	7
2	Common features that each frequency band is related to	10
3	Vision loss situation of patient group	34
4	Format of each measure for an instance	39
5	Positive values mean increment, and negative values mean decrement.	39
6	Results after treatment: 0 means non-improved, 1 means improved .	41
7	Parameters and details of the proposed 3D CNN model for hypothesis 1	42
8	Parameters and details of the proposed 3D CNN model for hypothesis 2	44
9	Parameters of the applied models	49
10	How the original 21 * 21 matrix (M1) is converted to the new 2 * 2 matrix (M2)	55

1 Introduction

1.1 Motivation

Vision loss is a severe problem for human beings, making the life of an individual difficult in various ways, from daily activities like reading and watching television to going out for exercising and coming back home. Simple tasks for normal-vision would become nearly impossible tasks for vision lost patients without assistance. About two-thirds of people have vision problems after a stroke [70]. This is because certain parts of the brain that can affect the vision is damaged. Visual field loss is the most common one of the problems, which means that the patient cannot see a part of their field of vision. The brain cannot process or analyze the scenes from specific area (damaged) of vision, though the eyes are regularly working.

Transcranial direct-current stimulation (tDCS) and Transcranial alternating current stimulation (tACS) for stroke patients are two treatment methods. After different periods of treatment, some patients can restore some vision. By recording the Electroencephalogram (EEG) data and vision field condition of stages correspondingly in different treatment periods, it is observed that the EEG data changed according to the restoration of the vision. This master's thesis aims to explore the correlation between the brain connectome and the visual field.

Deep learning techniques have drawn considerable attention and outperformed the traditional methods in analyzing the brain connectome in recent years. In this thesis, several proposed deep learning architectures will be compared and evaluated to find a comparatively ideal method to understand the relationship between visual field and brain oscillation after stroke and to predict the visual distribution as the output result.

1.2 Research Questions

This thesis mainly explores solutions to the following research questions:

- Is it possible to predict if the patients could be benefited before they take the

treatment, given their coherence matrix data?

- Is it possible to predict if the subject is a health control or a stroke patient, given their coherence matrix data?
- Is it possible to predict a patient's percentages of vision condition, given their coherence matrix data?
- Is it possible to predict the vision distribution for the stroke patient, given their coherence matrix data?

1.3 Structure of the work

- Chapter 2 covers the necessary background knowledge about this thesis: stroke, visual system, brain connectome, EEG, and essential contents of machine learning, including some popular machine learning architectures.
- Chapter 3 introduces and analyses some papers on classification tasks based on brain connectome using machine learning models. The models include the Convolutional Neural Network (CNN) and Graph Neural Network (GCN).
- Chapter 4 illustrates the methods and experiments conducted on the proposed questions. Different 1D CNN, 2D CNN, 3D CNN and GCN models were designed, compared and tried out for results.
- Chapter 5 and Chapter 6 summarise and analyze the results and give a conclusion of this thesis.

2 Background Knowledge

2.1 Stroke

According to an authoritative disease study in 2015, stroke came as the second most frequent cause of death after coronary artery disease, up to 6.3 million deaths (11% of the total), and only less than half of people who have had a stroke survive more than one year [75]. A stroke is a medical emergency in which low blood flow to the brain causes cell death. There are two types, ischemic stroke, and hemorrhagic stroke. An ischemic stroke is caused by a blood clot blocking or plugging a blood vessel in the brain, preventing blood from flowing to the brain. In a few minutes, brain cells would die. A hemorrhagic stroke occurs when a weakened blood vessel blasts and bleeds into the surrounding brain. A stroke has various complications: such as speech problems, visual problems, weakness, paralysis, and problems with balance or coordination, swallowing problems, memory problems, and so forth.

2.2 Visual System

The visual system consists of the optic nerve, the eye, optic chiasm, optic tract, lateral geniculate nucleus (LGN), optic radiation, and visual cortex, as depicted in Figure 1. The figure demonstrates how visual information passes from the left and right visual fields as they travel back to the occipital cortex. The scene from the left visual field (temporal) passes through the lens and reaches the right half of both retinas. Equally, the right visual field (nasal) lands the left half of both retinas. Thus, the scene human beings see is projected onto the retinas inversely.

Four visual conditions occur following a stroke: loss of central vision, visual field loss, visual perceptual abnormalities, and eye movement abnormalities. They occur in combination more frequently than in isolation. Loss of central vision is common after a stroke. Patients may suffer from blurred or altered vision. Though the vision improves by the time, some vision loss is permanent. Visual perceptual deficits indicate that visual information can not be interpreted or processed by the brain of patients so that they have difficulties recognizing faces or objects, with color vision

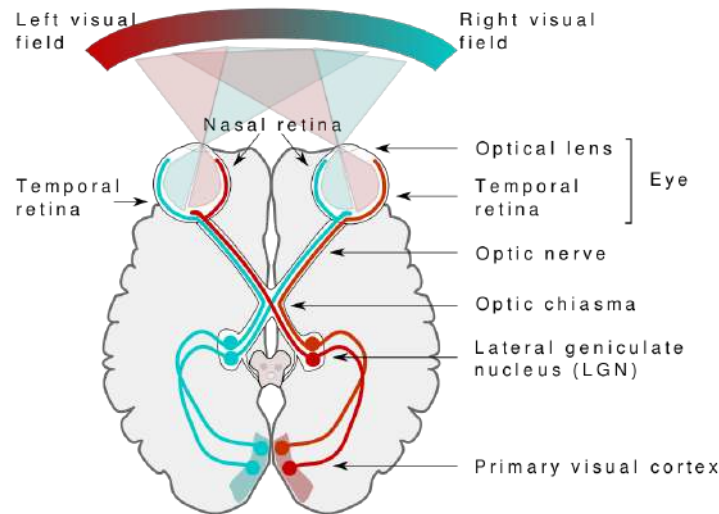


Figure 1: The visual system [1]

and with some other perceptual problems. Eye movement abnormalities contain misaligned eyes, having difficulties control the eyes to stare at near objects, or double vision.

Nearly half of people with a stroke have a visual field loss situation. Homonymous hemianopia (HH) is a highly seen type, which means vision loss is either in the right or the left visual field, as depicted in Figure 2 and Figure 3. These two figures show the normal vision (left one) and the left homonymous hemianopia patients' vision (right one).



Figure 2: The normal vision [74]

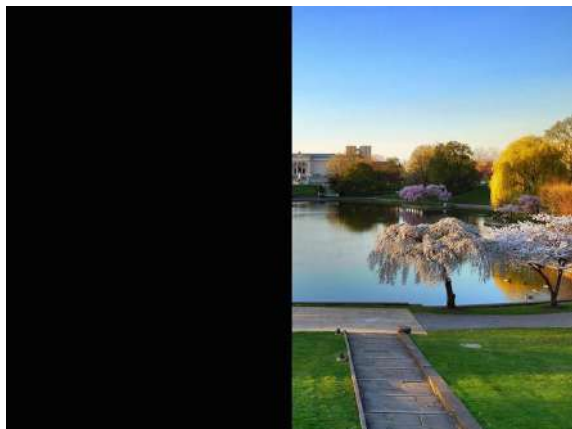


Figure 3: Left HH vision [74]

Depending on the defected areas, hemianopia has different visual field deficit cases. Figure 4 shows the common ones of hemianopia. Some cases, such as cases 4, 5, 6, and 7, have symmetric impaired areas on the visual fields. However, some do not, such as cases 1, 2, and 3. Different lesions may also result in the same impaired vision, such as case 4 and 7. Furthermore, the impaired area also varies from a quarter to a half or even more.

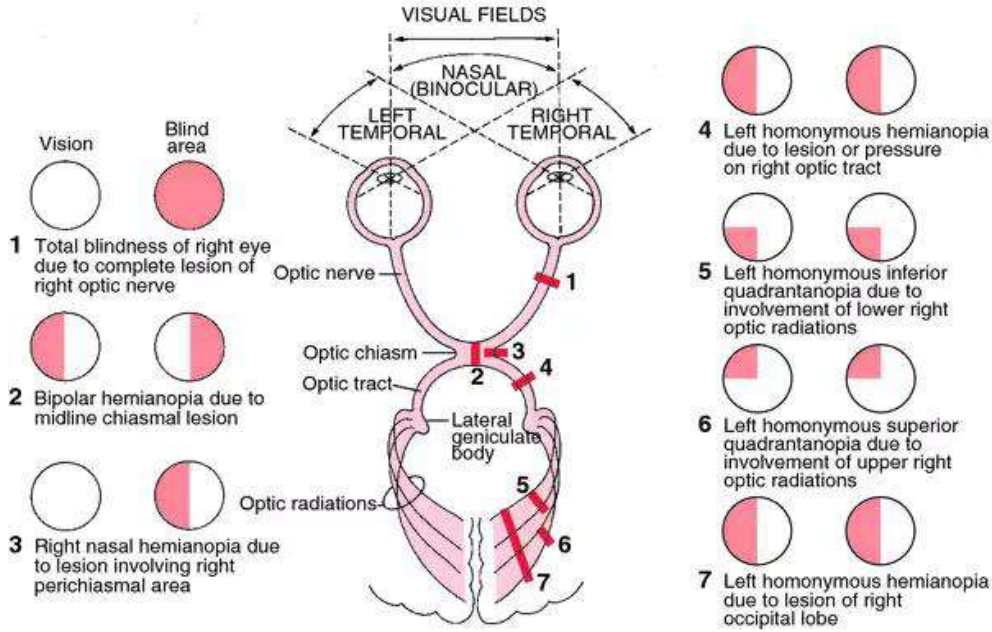


Figure 4: Some common cases of hemianopia [49]

2.3 Brain Connectome

The brain has large numbers of interconnected neurons that could be furtherly represented as the anatomical (structural) network, the functional network, and the effective network depending on the corresponding dataset.

The structural network means anatomical connections and is usually calculated with fiber tractography from diffusion tensor magnetic resonance imaging (DTI). Functional network and effective network are generally deduced from Blood-oxygen-level-dependent imaging-Functional magnetic resonance imaging (BOLD-fMRI) or EEG / Magnetoencephalography (MEG) signals. Figure 5 shows the respective source of three networks. The structural network connectivity is presented with broken lines,

functional connectivity with undirected solid lines and effective connectivity with directed solid lines.

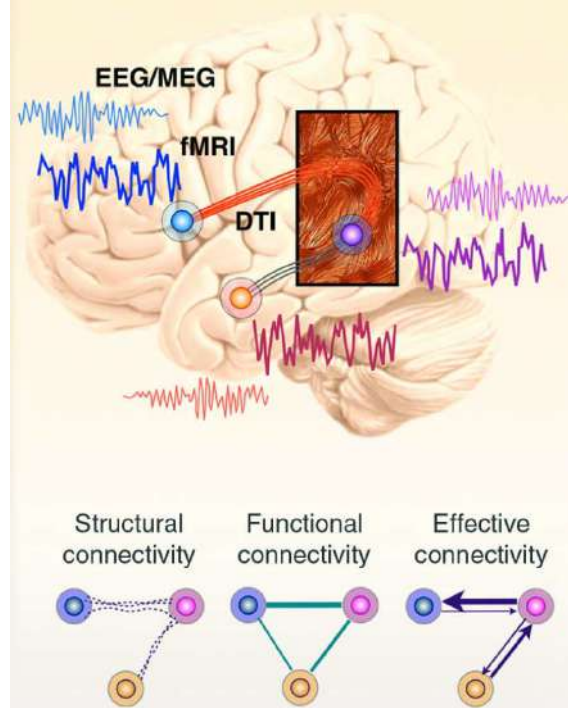


Figure 5: EEG/MEG, fMRI, DTI source of network [47]

Structural information of network elements (nodes) and connections (links) constitute the brain connectome [63]. Moreover, they are significant and helpful for human beings to figure out normal brain function and disease-related dysfunction.

The nodes should utmostly represent different brain regions with corresponding anatomical or functional connections. Parcellation schemes that include various connected brain regions and compress them into single nodes may be less useful. Furthermore, an ideal parcellation scheme should evenly cover the whole cortex's surface with individual nodes not overlapping each other spatially. If networks originate from the same parcellation scheme, they may be adequately compared. [25] and [28] also pointed out that the use of either MEG or EEG sensors may give rise to problems in this case, given that sensors can detect and collect overlapping signals out of the required boundaries. In chapter 4, the specific parcellation scheme used in this thesis would be covered.

The links are different based on their weight and directionality. Binary links indicate

the presence (value 1) or the absence (value 0) of the connections, and the weighted links contain extra information apart from directions and status. In anatomical networks, weights can represent density, the size, or the coherence of the nodes. In functional or effective networks, they can represent magnitudes of interactions [54]. Weak and trivial links may be spurious and noisy, thus applying a proportional threshold to the links would be needed. Directed links are seldomly used in neuroimaging. In this thesis, we only consider undirected links.

2.4 Electroencephalogram (EEG)

EEG is a monitoring method used for recording electrical activity in the brain. EEG was first discovered and obtained by Richard Caton (1842 - 1926) in the 19th century from some animals' brains, and later in 1924, Hans Berger (1873 - 1941) was the first person to record human EEG signals successfully. When neurons send a signal to other neurons, an electrical current is produced, which is extremely weak. If a large number of electrical currents are produced synchronously, and they are high enough, the electrode can detect them [8][58][50], and experts on the equipment could observe the signal. Figure 6 [61] indicates the details about how EEG record through tissues.

There are two types of EEGs, scalp (noninvasive) and intracranial (invasive), depending on where the signals are detected. As for the scalp EEG, electrodes are placed on the scalp with assured physical and electrical contact. In the intracranial EEG, electrodes are implanted in the brain during the surgery and collect signals. Figure 7 shows a subject taking the scalp EEG.

Lobe	Functions
Frontal lobe	Personality, behavior, emotions and so on.
Parietal lobe	Sensation, sensory comprehension, recognition and so on.
Occipital lobe	Visual processing (color, light and movement).
Temporal lobe	Recognition of auditory stimuli, speech, and so on.

Table 1: A table showing the lobes and corresponding functions

The brain consists of three parts: the cerebrum, cerebellum, and brainstem (Fig. 8). The cerebrum is the largest and most significant section of the human brain, and it is in charge of the advanced functions of human beings. Furthermore, the

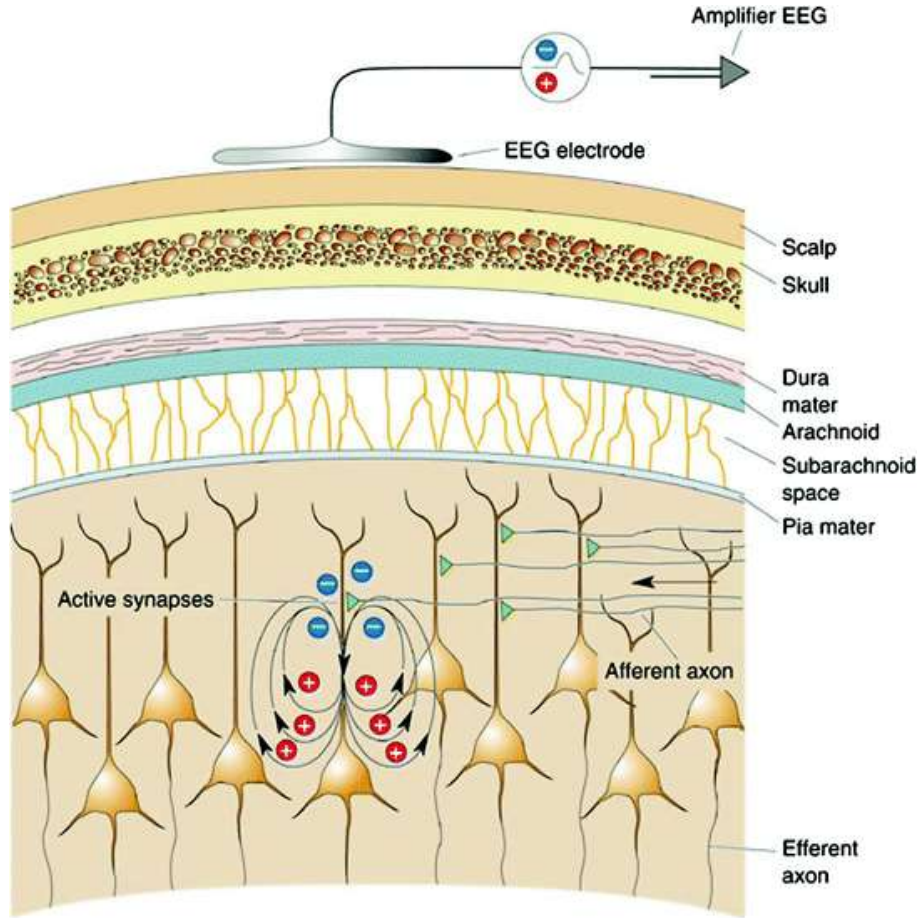


Figure 6: As seen in the image, the EEG electrode needs to measure the signal through many tissue layers (such as the scalp, skull, dura, mater, arachnoid sub-arachnoid space). Synaptic currents in cells (golden pyramidal shape) can generate electrical fields (voltage) as signals. Only when signals are accumulated to be high enough, they can be detected [21][20].

cerebrum can be divided into left and right hemispheres, each hemisphere with four lobes: frontal, parietal, occipital, and temporal [50], which are responsible for various bodily functions. Each lobe can further be divided into subregions (Figure 9).

Table 1 shows the functions each lobe is responsible for.

The cerebellum is situated under the cerebrum at the lower back of the head. It is smaller than the cerebrum yet still the second largest section of the brain and includes more than half of its neurons. The cerebellum has control of these functions: muscle movements, posture maintenance, body coordination, and so forth. The brainstem is situated at the base of the brain connecting the cerebrum to the spinal cord. It is related to the body's various functions, such as motor and sensory



Figure 7: An example of scalp EEG [16]

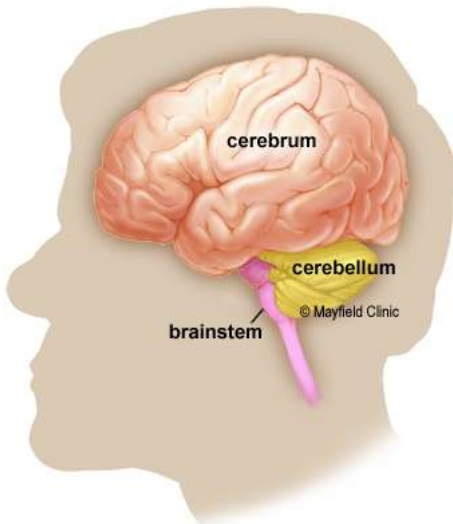


Figure 8: General structure of the brain [3]

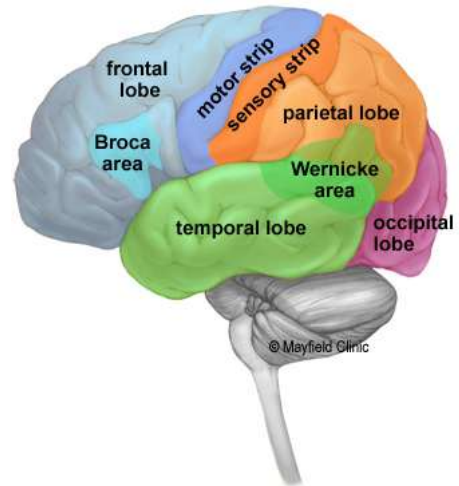


Figure 9: General structure of the cerebrum [3]

functions (heat, pain, etc.), eyes and mouth movements, even heartbeat, and so on.

When using EEG for measurement, the cerebral cortex's corresponding electrical activity has the most influence on the recordings as the cerebral cortex is on the surface position, and it is responsible for cognitive tasks. Clinically, EEG is used for diagnosis and investigation of neurological disorders and other abnormality diseases. What EEG measures are the oscillation of neurons, and the signal is presented as a combination of several base frequencies, which usually are Delta ($< 4\text{Hz}$), Theta ($4 - 7\text{Hz}$), Alpha ($7 - 12\text{Hz}$), Beta ($12 - 30\text{Hz}$) and Gamma ($> 30\text{Hz}$), as shown in Figure

10. Table 2 displays the common features that each frequency band is related to.

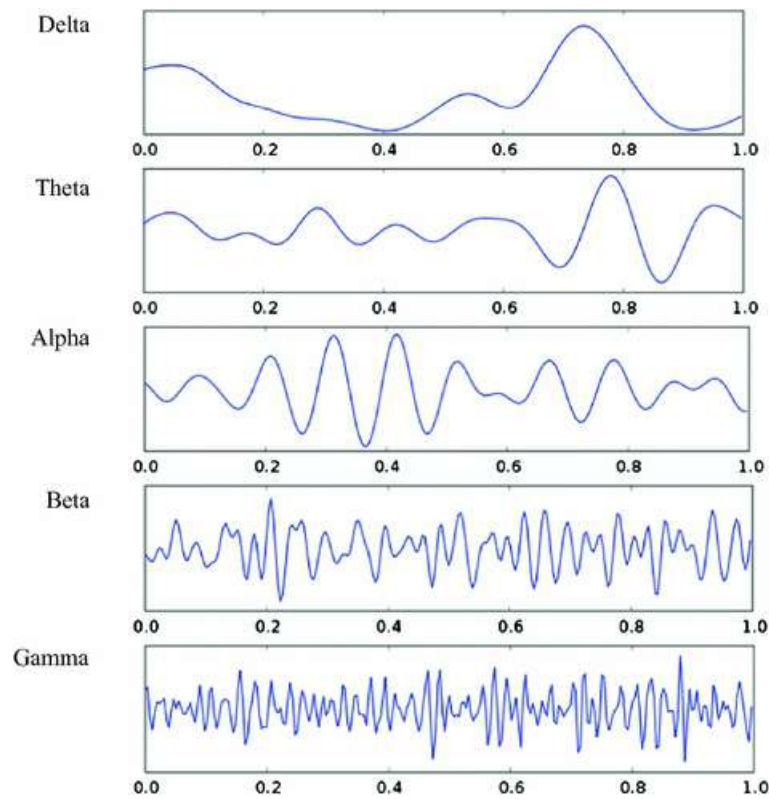


Figure 10: An example of different types of normal EEG rhythms in a second [61].

Band	Features
Delta	Assessing the depth of sleep, serious brain disorders and the waking state.
Theta	Correlating with mental operations, increasing with task difficulty.
Alpha	Correlating with reflecting sensory, motor and memory when eyes closed.
Beta	Correlating with active attentions or solving concrete problems
Gamma	Correlating with cognition, attention, memory, and so on [68].

Table 2: Common features that each frequency band is related to

EEG can be in cognitive research use or clinical abnormality disease diagnosis use, and it has been used to study stroke for decades [76][46][5]. As was mentioned in Chapter 2.1, most stroke patients will recover in a few months with rehabilitation.

Rehabilitation is a clinically therapeutic method aimed at recovering patients' physical, psychological, and social potential maximumly [29]. Some mutually independent experiments proved that patients recover most in the first three months [16], while some other studies found that recovery reached its most in the range of the second and fifth months after stroke [30][18]. There is consensus that in the following six months after the stroke, recovery is highly active and potential; after six months, the recovery is limited.

2.5 Deep Learning

Arthur Samuel first introduced the idea of machine learning in 1952. However, till the 1990s, machine learning started to develop fastly and shift from a knowledge-driven approach to a data-driven approach and later on till now, due to the speed of GPU and tremendous amounts of data, deep learning as a subset of machine learning, has boosted the development of modern society in various aspects. It is widely applied in daily life, from object detection [17] to face recognition [65], from mailbox filtering [77] to product recommendation [79], from fraud detection [53] to stock prediction [40], from speech recognition [37] to automatic music composition [14].

The terms *Artificial Intelligence* (AI), *Machine Learning* (ML) and *Deep learning* (DL) are commonly seen, and the connections and differences among them are also straightforward and simple (Figure 11). AI describes machines (or computers) that mimic human beings' behaviors. ML is the algorithm that fulfills the goal of AI. DL is a subset of ML.

2.5.1 Three Typical Algorithms

There are mainly three different categories in ML algorithms grouped by learning style: Supervised Learning (SL), Unsupervised Learning (UL), and Semi-Supervised Learning (SSL), and DL is also based on them. SL method means that the model is trained with both input data and corresponding output data (input data and output data are in pairs), and the example tasks are classification and regression (Figure 12). UL method indicates that input data is given to the model without corresponding output data, and the model organizes the data according to some predefined metrics like similarity. SSL method gets the input data containing both

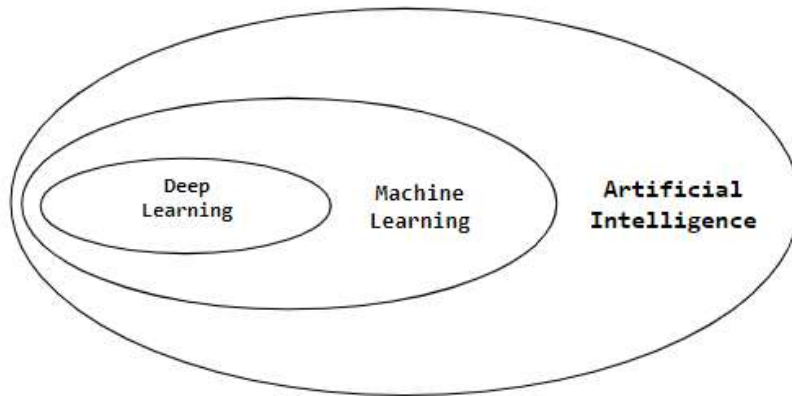


Figure 11: Relationships among AI, ML and DL

labeled and unlabeled instances, and it is also used for classification and regression problems.

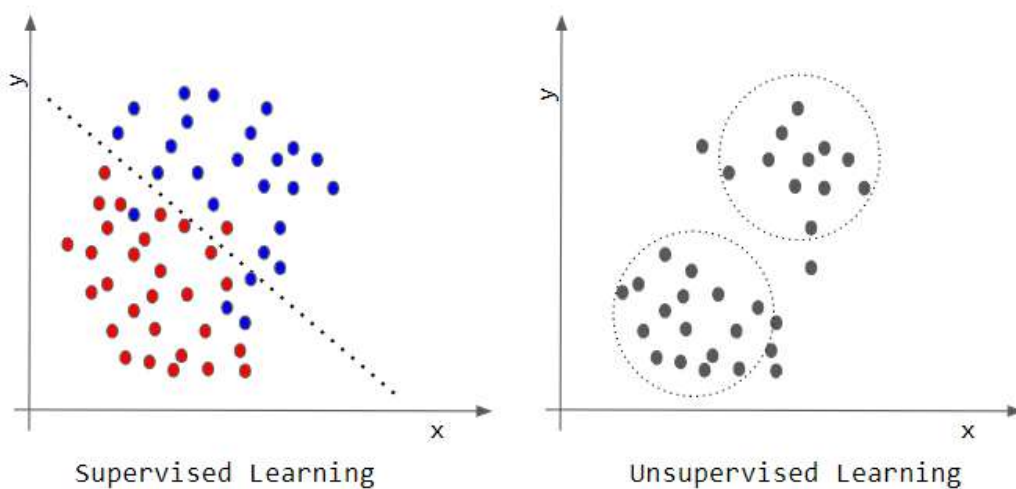


Figure 12: General examples of SL model and UL model

2.5.2 Multilayer Perceptron Model (MLP)

Figure 13 shows a basic deep learning structure called MLP, and it is a supervised learning method. There are three main parts in this structure: Input Layer, Hidden Layers and Output Layer. In the input layer, $X = A^{[0]}$ is the input data which

contains two units: $a^{[0]}_1$ and $a^{[0]}_2$. In the hidden layer, there are three temporary output units: $a^{[1]}_1$, $a^{[1]}_2$, $a^{[1]}_3$. In the output layer, there is one output $a^{[2]}$ and a final output \hat{Y} . When training starts, all the \mathbf{W} s multiply (dot product) with their corresponding \mathbf{a} . \mathbf{W} s indicate the weights, and they are trainable and usually randomly initialized. The computation function could be presented as Equation 1, where ϕ is the activation function, bias is a constant that helps the model flexible and fit for the given data.

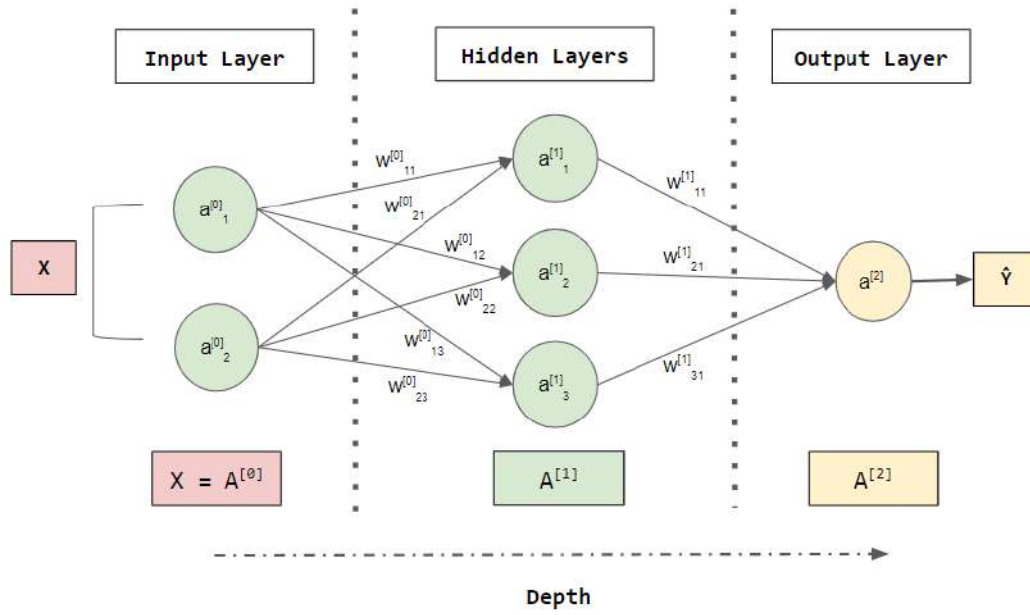


Figure 13: An example of MLP structure

$$\mathbf{A}^{[n+1]} = \phi(\mathbf{A}^{[n]} \mathbf{W}^{[n]} + \text{bias}) \quad (1)$$

The commonly used activation functions are rectified linear unit (ReLU) function [44], Sigmoid function, Tanh function and so on. Figure 14 is the ReLU function, and its matching mathematical equation is Equation 2. Figure 15 is the Sigmoid function, with Equation 3 as its matching mathematical equation. Figure 16 is the Tanh function, and Equation 4 is the matching mathematical equation.

$$\text{ReLU} : R(x) = \max(0, x) \quad (2)$$

$$\text{Sigmoid} : S(x) = \frac{1}{1 + e^{-x}} \quad (3)$$

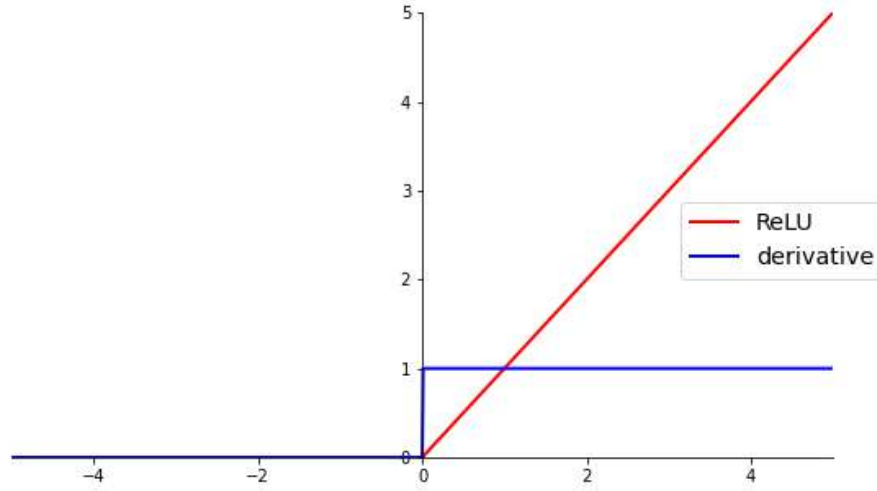


Figure 14: ReLU Activation Function

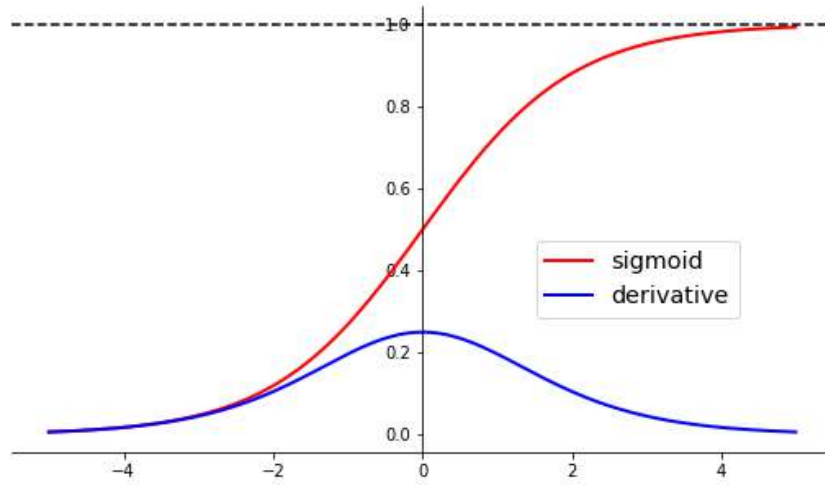


Figure 15: Sigmoid Activation Function

$$\text{Tanh} : T(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (4)$$

The information travels forward through the structure from left to right and does the necessary computation and activation functions until the final output. This process is called forward propagation. “During training, forward propagation can continue onward until it produces a scalar cost $J(\Theta)$.” [23]. The back-propagation

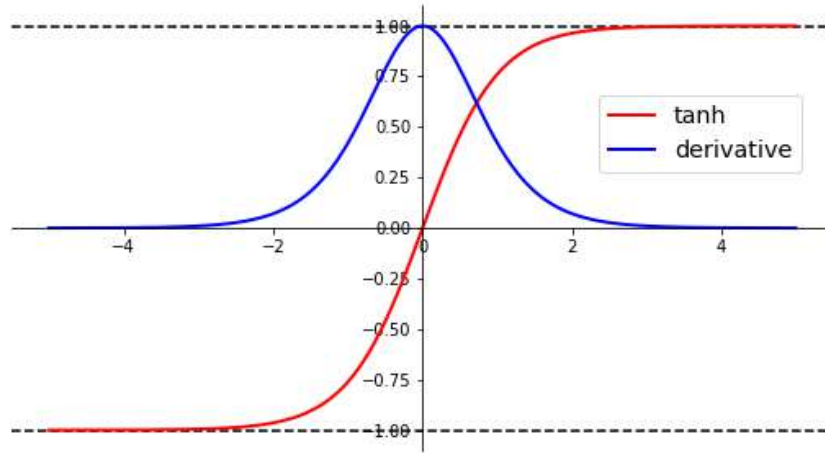


Figure 16: Tanh Activation Function

algorithm [56] is repeatedly applied in a reverse order starting from output towards input to compute the gradient. Back-propagation refers to the method of computing the gradient only, and there are other commonly used algorithms for optimizing the structure based on the gradient. These algorithms aim at converging to the minimum of the cost function as fast as possible. A cost function is a method of measuring the machine learning model's ability based on how bad the model is predicting output data given input data. The smaller the cost is, the better the model is. "Cost" usually is presented as a difference or distance between the predicted value and ground truth value.

Gradient descent [13] is a commonly used and efficient optimization algorithm attempting to find a minimum (local or global) of the cost function. The Equation 5 presents the gradient descent algorithm in a single line and a simple way. Θ^1 is the next position value, Θ^0 is the current position value, '-' means opposite gradient direction (reducing the current value), α is the learning rate deciding how big the moving step is, and $\nabla J(\Theta)$ indicates the partial derivative of cost function $J(\Theta)$ with respect to Θ . Iteratively, with an appropriate learning rate, the algorithm could find the minima. There are three variants of gradient descent, and they are different in how much data is used to compute the gradient of the cost function. Depending on the amount of data, a balance between the accuracy of the parameters' update and the time it takes to perform an update will be achieved [55].

$$\Theta^1 = \Theta^0 - \alpha \nabla J(\Theta) \quad (5)$$

Batch gradient descent computes the gradient of the cost function for the entire training dataset. The gradients for the whole dataset are calculated to perform one update, making the process slow and steep for datasets that are too large to fit in memory. This method is guaranteed to converge to the global minimum for convex surfaces and a local minimum for non-convex surfaces, respectively [55]. Stochastic gradient descent (SGD) [51] performs a parameter update for each training instance. Thus, it is usually much faster. However, SGD performs the update frequently and results in wild fluctuation in the cost function. This problem could be solved by slowly decreasing the learning rate, and SGD will give the same convergence behavior as batch gradient descent. Mini-batch gradient descent takes the best parts of both batch gradient descent and SGD and performs an update for every mini-batch, which contains n training instances. The whole original dataset will be divided into batches evenly, and the typical mini-batch size is from 50 - 256 instances, but it still varies depending on the real application. Mini-batch gradient descent, though it has advantages of the other two methods, cannot guarantee good convergence. There are still several problems remaining:

- How to determine a proper learning rate.
- How to adjust the learning rate during training.
- How to deal with non-convex cost functions that have numerous local minima.

2.5.3 Gradient Descent Optimization Algorithms

To address the aforementioned problems, some gradient descent optimization algorithms were introduced. Momentum is a method that can accelerate SGD in the relevant direction and weaken fluctuations, as shown in Figure 17. Comparison between with and without momentum. An additional fraction value is multiplied with the current position value Θ^0 in Equation 5 to prevent the value from decreasing too fast.

Adagrad [15] is an algorithm for gradient-based optimization that adapts the learning rate based on the situations. It performs more massive updates for infrequent parameters, and for frequent ones, it does smaller operations. This algorithm works

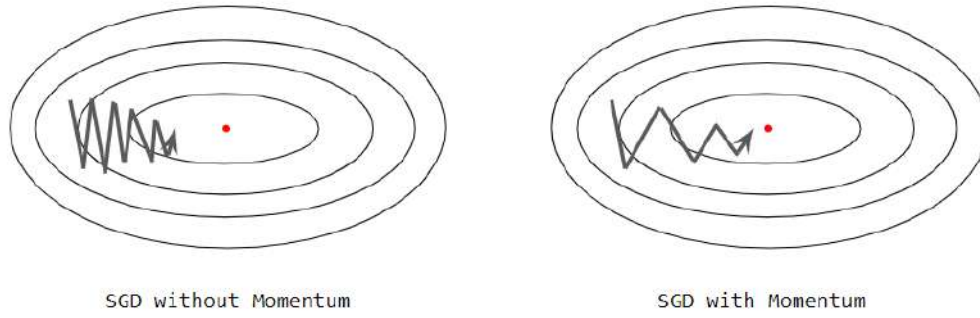


Figure 17: Tracking Comparison between with and without momentum

well for sparse data. One of Adagrad’s advantages is that there is no need for manual tuning the learning rate. In this way, the learning rate will become extremely small at a specific point, and from then on, it would be too small to give more information. Adadelata [78], RMSprop and Adaptive Moment Estimation (Adam) [31] were proposed subsequently to solve this weakness. All in all, Adam might be the best choice for sparser data [31].

2.5.4 Loss Functions

A loss function is a method of evaluating how well your models are performing. If the model predicts close to the actual values, the loss function will give a lower cost. If the model predicts far away from the actual values, the loss function will give a higher cost. During the training process, it leads the model to update its parameters in the right direction, and in the testing period, it validates whether the model is good or not.

Mean Squared Errors (MSE) indicates how close a regression line is to a series of given points. Its definition is the average of the sum of the square of the difference between all pairs of the predicted and ground truth values. It is frequently used as the loss function in regression tasks. A loss function is a method of evaluating how well your models are performing. If the model predicts close to the actual values, the loss function will give a lower cost. If the model predicts far away from the actual values, the loss function will give a higher cost. During the training process, it leads the model to update its parameters in the right direction, and in the testing period, it validates whether the model is good or not, as depicted in Equation 6, where $\hat{y}_{(i)}$ indicates the predicted value and $y_{(i)}$ indicates the true value.

$$MSE = \sum_{i=1}^n \frac{(\hat{y}_{(i)} - y_{(i)})^2}{n} \quad (6)$$

Mean Absolute Error (MAE) is another metric that can be used to check the performance of the model in regression tasks. What makes it different from MSE is that instead of calculating the square of the difference between all pairs of the predicted and ground truth values, it calculates the absolute value between those values, as depicted in Equation 7, where $\hat{y}_{(i)}$ indicates the predicted value and $y_{(i)}$ indicates the true value.

$$MAE = \sum_{i=1}^n \frac{|\hat{y}_{(i)} - y_{(i)}|}{n} \quad (7)$$

R-Squared (also known as the coefficient of determination) is also a metric measure of how close the data is to the matching regression line, as depicted in Equation 8, where $\hat{y}_{(i)}$ indicates the predicted value, $y_{(i)}$ indicates the true value and \bar{y} indicates the average of all the actual values. In this thesis, these three metrics will be applied accordingly. Importantly, R-Squared cannot tell if the coefficient estimates and predictions are biased, which means it does not always indicate whether the model is adequate. A good model might have a low R-squared score, and a poor model might have a high R-squared score.

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_{(i)} - \hat{y}_{(i)})^2}{\sum_{i=1}^n (y_{(i)} - \bar{y})^2} \quad (8)$$

Cross Entropy is another loss function commonly used in deep learning, generally in classification tasks. The cross entropy equation uses two distributions, namely the true distribution $p(x)$ and the estimated distribution $q(x)$, which are defined by the discrete variable i , as depicted in Equation 9, where n indicates the number of the classed in the classification task. When i equals 2, it is binary cross entropy, which is also the loss function later applied in this thesis.

$$CE(p, q) = - \sum_{i=1}^n p(i) \log q(i) \quad (9)$$

2.6 Convolutional Neural Network (CNN)

CNN [38] is a type of neural network for processing data that possesses a grid-like topology. The data could be time-series data, which can be regarded as a 1D grid, including some instances at regular time intervals. It could also be image data that can be regarded as a 2D grid of pixels. The CNN models started to draw wide attention in 2012 because of ‘ImageNet competition,’ and AlexNet [34] topped the chart with 16.4%, which was 7.8% smaller than that of second place’s. Since then, CNN has been broadly updated and developed to narrow down the error rate and achieve competitive results. Similar to MLP, CNNs consist of layers of neurons that have learnable weights and biases. Each neuron receives corresponding input data, operates a dot product, and usually is followed with a proper activation function. This computation process is repeated layer by layer until the output layer when the network’s result is predicted. However, MLP-like models use fully connected layers, which means that each neuron is connected to every neuron in the previous layer. Furthermore, it does not work well in terms of computational resources. Figure 18 is an example of the structure of CNN. This CNN is composed of different layers, such as convolutional layers, activation layers, pooling layers, fully connected layers, and a softmax layer. Softmax is another activation function widely used in classification tasks. These layers can be stacked in several ways.

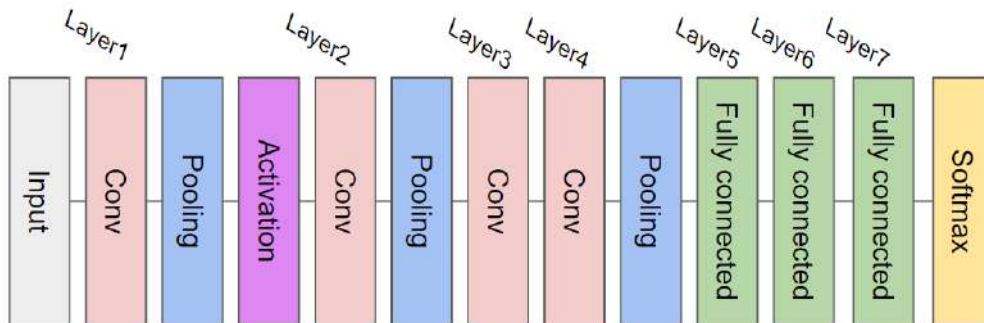


Figure 18: A simple CNN model. There are seven layers in this model which are three convolutional layers and three fully connected layers. The last activation function is a Softmax.

2.6.1 The Convolutional Layer

The convolutional layer includes a series of filters, and its task is to perform a convolution operation between these filters and the layer's input to create feature maps. A filter is a grid of discrete values, and usually, it is square-shaped. The values in it refer to a single template/pattern. Input data is a set of values in a $6 * 6$ grid (can be regarded as image data), and the filter is a $3 * 3$ grid. The convolution operation initially positions the filter over the left upper section of the image. It then performs an element-wise product between the filter's parameters and the matched $3 * 3$ grid in the input grid, followed by summing the result to a single value (Figure 19); in a nutshell, it performs dot product operation.

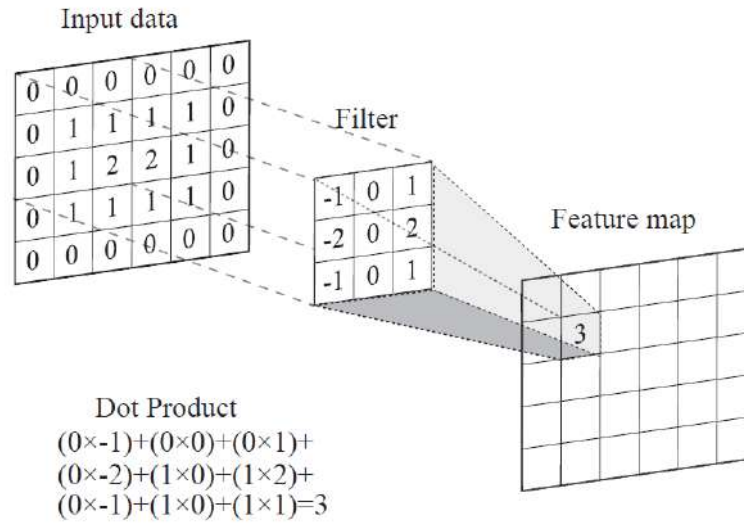


Figure 19: An example of how the convolution operation calculation works [69].

Subsequently, the convolution operation slides the filter to the right and gets the dot product in this new position. The sliding regulation is from left to right and top to bottom across the input grid and produces the corresponding new values, as shown in Figure 20. When the convolution operation is finished, a new grid with new values is generated. It is called the feature map, which contains the spatial information of the original input grid.

In this way, each filter preserves only one pattern. Scanning the image grid by merely one pattern is likely to result in a very biased network. Thus, a convolutional layer usually has several filters, each one producing one 2D feature map. Then all of the feature maps are stacked together to form a 3D volume, as shown in Figure 21.

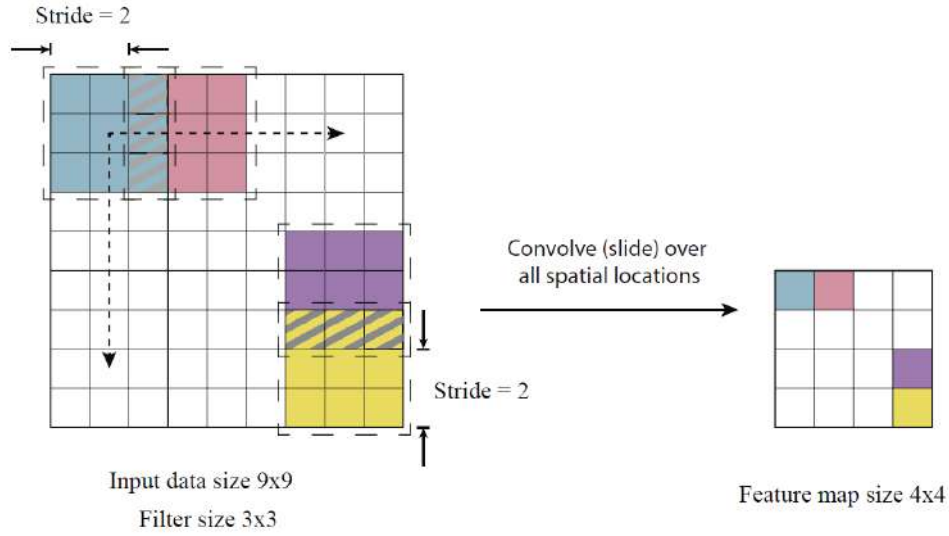


Figure 20: The convolution operation The operation starts with the filter in the top left corner of the input grid and calculates the dot product between this filter and the input grid; a single value, as a result, is added to the matching position in the feature map. Next, slide the filter one step (one grid) to the right (the same color indicating the matching filter and result) and repeat the same operation and so on until the entire area of the input layer is gone through from left to right and top to bottom [69].

Here, the convolution operation is applied to the input image several times using different filters (green and red, and other colors not shown). Each filter produces a unique feature map. All the feature maps are stacked together to generate the layer's output. The number of the filters usually is set to the power of 2, mainly between 32 to 512 [69]. The filter's size is usually 3×3 , but may also be customized according to the application.

2.6.2 The Pooling Layer

The pooling layer is designed to decrease the size of the feature maps. It can remain almost the same information but make the feature map smaller. The commonly used forms of pooling are the max pooling and average pooling (Figure 22). The max-pooling layer slides a size-fixed window over its input and picks the max value, and discards all other values. The average pooling layer slides a size-fixed window over its input and computes the average value of the input area that the window falls on.

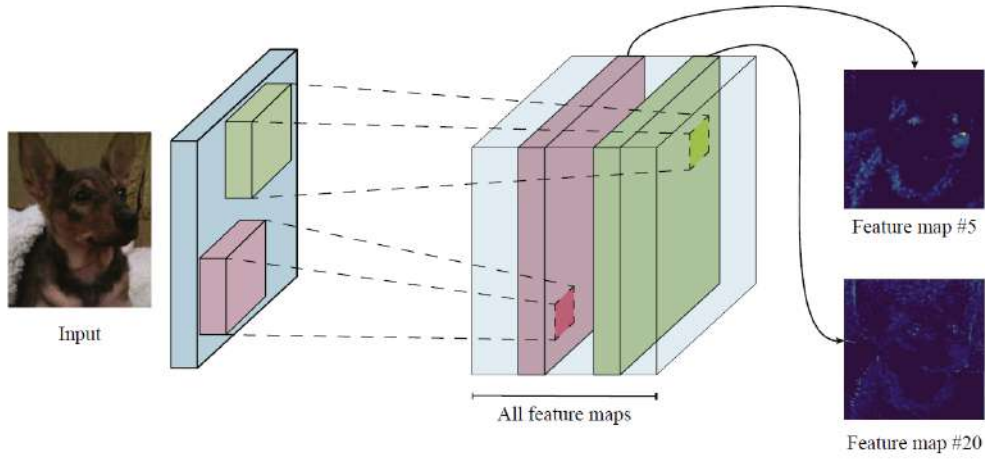


Figure 21: An practical illustration about the feature maps [69].

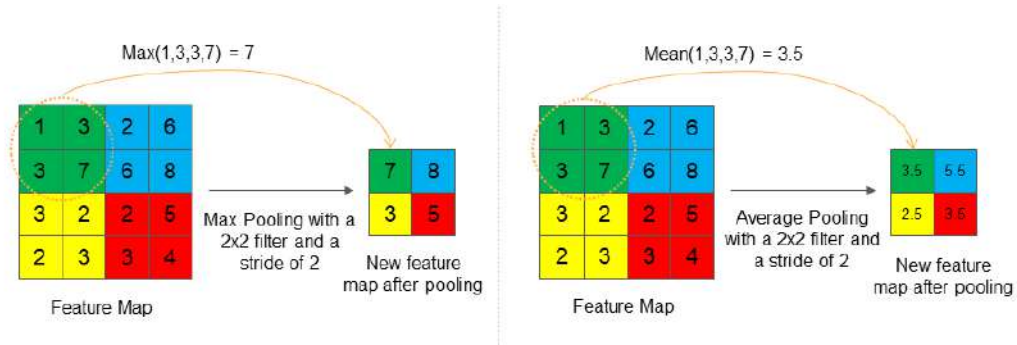


Figure 22: Left one is the max pooling operation, right one is the average pooling operation.

2.6.3 Classic CNN Model: VGG16

Visual Geometry Group 16 (VGG16) [60] is one of the most famous and widely used models based on CNN structure, which was proposed in 2014. Many following networks are further developed, taking VGG16 as the backbone. VGG16 is mainly applied in the image classification tasks. See Figure 23. The VGG16 structure can be regarded as two parts: the feature extraction part and the classification part. In the first part, each block contains certain pairs of convolutional layers and ReLU layers ending with a max-pooling layer. Moreover, the initial input size to the conv1 layer is $224 * 224 * 3$, which is an RGB image. This block pattern is repeated five times, and the feature map is reduced to the size of $7 * 7 * 512$. In the second part, feature maps are flattened to a 1D vector and then followed by two FC layers until the last softmax activation function, which gives the predefined 1000 classes' probabilities.

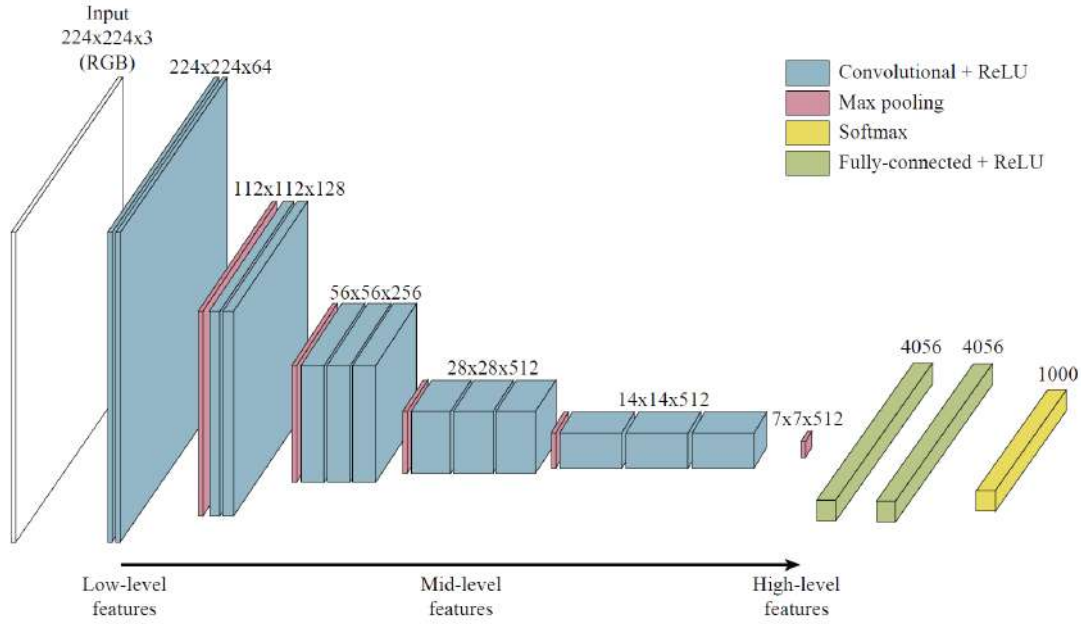


Figure 23: VGG16 structure [69]

2.6.4 3D CNN

Like typical 2D CNN, 3D CNN has the similar pipeline. 3D CNN-based architectures have been used to perform various 3D vision tasks, remarkably outperforming many previous 2D CNN approaches. For example, [43] proposed Voxnet architecture of real-time object recognition for 3D point cloud data, which includes LiDAR point clouds, RGBD point clouds, and CAD models. [62] proposed the first 3D Region Proposal Network (RPN) that takes a 3D volumetric scene from an RGB-D image as input and outputs 3D object bounding boxes, which is much faster than the original Sliding Shapes method. [22] proposed a 3D CNN approach that takes a 3D volumetric representation of the hand depth image as input and captures the 3D spatial structure of the data and accurately regresses full 3D hand pose in a single pass real-timely. 3D U-Net proposed in [11] extends the previous u-net [52] architecture by replacing all 2D operations with their 3D counterparts and achieves both semi-automatic and fully-automated medical image segmentation. Generally, in 2D CNN, the filter moves in 2 directions (from left to right, top to bottom), and input and output data are 3D. While in 3D CNN, the filter moves in three directions. Like a 2D image with three dimensions, the 3D image has four dimensions, with the last dimension indicating the color channels (Figure 24).

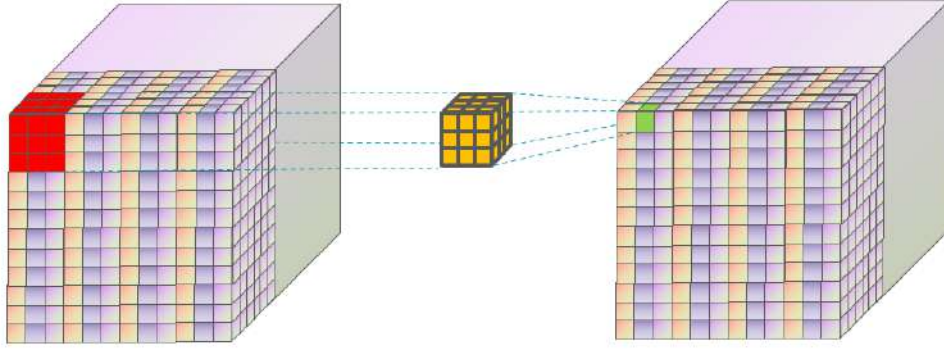


Figure 24: An example of 3D convolutional operation: The input data is with a shape of $[12, 12, n, 3]$, and one filter size is $[3, 3, 3]$ (always keep the number of color channels in filters the same with that in input data). The cube filter (the yellow one) will do a dot product with a matching cube in input data (the red one) and get a new value (green one) in the following feature map. Here it is done with the same-padding operation, which maintains the original size of the input data.

2.7 Graph Neural Network

CNN has outstanding performances on grid-like data. However, for non-grid-like data, for example, graph-structured data, the performance is barely acceptable. Many scientists started to work on graph neural networks (GNNs) to address this bottleneck, which was first proposed in [59]. Due to its competitive performance and high interpretability, GNN has been widely applied in graph analysis and had many variants for the past few years.

Graphs are the prerequisites in the study of GNNs. Therefore, to figure out how GNN works, the fundamental graph theory is required. A graph usually is denoted by $\mathbf{G} = (\mathbf{V}, \mathbf{E})$. \mathbf{V} means the set of vertices (nodes), and \mathbf{E} is the set of edges. An edge $\mathbf{e} = (\mathbf{u}, \mathbf{v})$ has two endpoints \mathbf{u} and \mathbf{v} , which is linked by \mathbf{e} . In this case, \mathbf{u} is called a neighbor of \mathbf{v} , meaning that \mathbf{u} and \mathbf{v} are adjacent. As it was mentioned earlier in chapter 2.3, an edge can either be directed or undirected, and we only focus on undirected graphs in this thesis. The degree of vertex \mathbf{v} , denoted by $d(\mathbf{v})$, is the number of neighbors that \mathbf{v} has, or the number of edges linked with \mathbf{v} .

Adjacency matrix (AM): $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is a graph with n vertices, and it can be presented by an AM $\mathbf{A} \in \mathbb{R}^{n \times n}$, and \mathbf{A} is a symmetric matrix.

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if } \{v_i, v_j\} \in \mathbf{E} \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

Degree matrix: $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is a graph with n vertices, and its degree matrix $\mathbf{D} \in \mathbb{R}^{n \times n}$ is a diagonal matrix, where $\mathbf{D}_{ii} = d(v_i)$.

Laplacian matrix: $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ is a graph with n vertices, and its Laplacian matrix $\mathbf{L} \in \mathbb{R}^{n \times n}$ can be defined as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where

$$\mathbf{L}_{ij} = \begin{cases} d(v_i), & \text{if } i = j, \\ -1, & \text{if } \{v_i, v_j\} \in \mathbf{E} \text{ and } i \neq j, \\ 0, & \text{otherwise.} \end{cases} \quad (11)$$

Symmetric normalized Laplacian: the symmetric normalized Laplacian then can be defined as:

$$\begin{aligned} \mathbf{L}_{sym} &= \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{D}^{-\frac{1}{2}} (\mathbf{D} - \mathbf{A}) \mathbf{D}^{-\frac{1}{2}} \\ &= \mathbf{I} - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} \end{aligned} \quad (12)$$

\mathbf{I} is the identity matrix.

2.7.1 Spectral Methods

In 2014, [6] proposed a spectral network, where the convolution operation is defined in the Fourier domain by computing the eigendecomposition of the graph Laplacian. The operation is defined as the multiplication of a signal $\mathbf{x} \in \mathbb{R}^N$ (a scalar for each node) with a filter $\mathbf{g}_\theta = \text{diag}(\theta)$ parameterized by $\theta \in \mathbb{R}^N$:

$$\mathbf{g}_\theta * \mathbf{x} = \mathbf{U}_\theta(\boldsymbol{\Lambda}) \mathbf{U}^T \mathbf{x} \quad (13)$$

In Equation 13 [6], \mathbf{U} is the matrix of eigenvectors of the normalized graph Laplacian $\mathbf{L} = \mathbf{I}_N - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} = \mathbf{U} \boldsymbol{\Lambda} \mathbf{U}^T$, where $\boldsymbol{\Lambda}$ is the diagonal matrix of its eigenvalues. The drawback of this operation is that it causes potentially intense computations and non-spatially localized filters.

In order to alleviate the problem of overfitting on local neighborhood structures for graphs with broad node degree distributions, [32] proposed a solution, which is to set the lay-wise convolution operation $\mathbf{K} = 1$ and generalize the definition to a signal $\mathbf{x} \in \mathbb{R}^N$ with \mathbf{C} as input channels and \mathbf{F} as filters for feature maps in equation 14, where $\Theta \in \mathbb{R}^{\mathbf{C} * \mathbf{F}}$ is a matrix of filter parameters and $\mathbf{Z} \in \mathbb{R}^{N * \mathbf{F}}$ is the convolved signal matrix.

$$\mathbf{Z} = \tilde{\mathbf{D}}^{-\frac{1}{2}} \tilde{\mathbf{A}} \tilde{\mathbf{D}}^{-\frac{1}{2}} \mathbf{x} \Theta \quad (14)$$

2.7.2 Spatial Methods

In spectral methods, the Laplacian eigenbasis determines the learned filters, and the graph structure determines the Laplacian eigenbasis. This means a specific model could only be applied to the particular graph with the same structure as the one used for training. On the contrary, spatial methods self-define convolutions directly on the graph, operating on its spatially close neighbor nodes. However, this also brings other problems, such as the difficulties with defining the convolution operation with differently sized neighborhoods and maintaining the local invariance of CNNs [41].

[45] proposed The PATCHY-SAN model, which first selects and normalizes exactly k neighbors for each node. These normalized neighbors work as the receptive field on which the convolutional operation is applied. It has four steps to conclude the whole process:

- It starts by getting the order of the nodes and the sequence of the nodes. Then it selects the nodes with a predefined stride from the sequence until the required number of nodes are selected.
- The model uses a simple breadth-first search to collect k neighbors for each node. It collects the 1-hop neighbors of the node at first. When 1-hop neighbors are all collected, it collects high-order neighbors until k neighbors are included.
- In this step, nodes in the receptive field are given orders mapping from an unordered graph to a vector space. With this feature, nodes from two different graphs could be assigned to similar relative positions when they have similar structural patterns.

- When the receptive fields' normalization operation is done, the regular CNN architectures can be applied.

This method somehow switches the graph problem into the typical euclidean problem.

3 Related Work

Owing to the rise in the popularity of deep learning, ML engineers have been trying to use deep learning models to deal with brain connectome analysis. [71] proposed a six-layer CNN for recognition and classification of Alzheimer’s disease (AD) and dementia with Lewy bodies (DLB) at the individual subject level with the accuracy, average precision, and recall being 0.73, 0.78, and 0.73. It proves that CNNs could further do structural brain network analysis to differentiate dementia disorders in individual subjects (Figure 25).

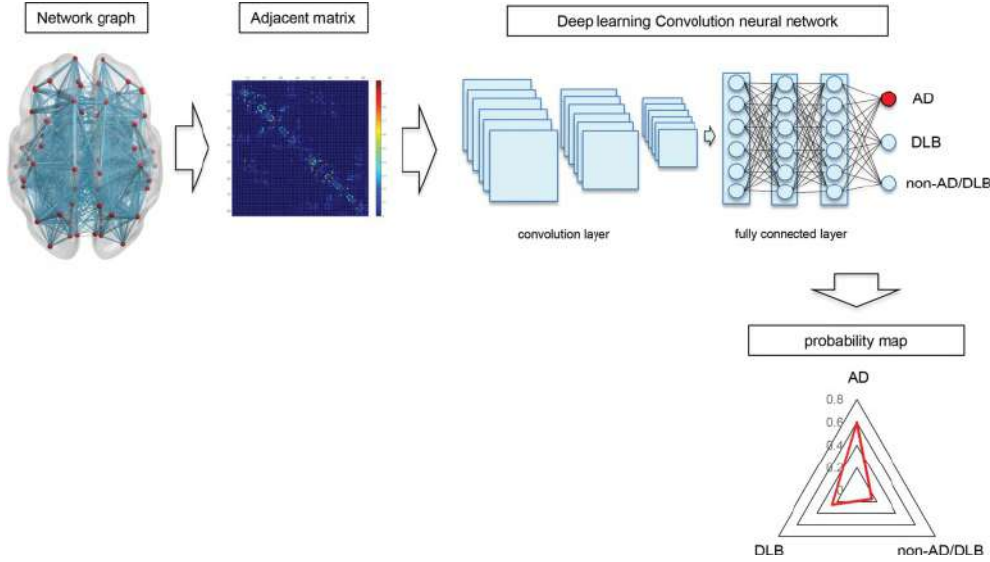


Figure 25: The architecture of the model proposed in [71].

[48] designed a multi-domain connectome CNN (MDC-CNN) for classification of EEG-derived brain connectome in schizophrenia (SZ), and this model contains a 1D CNN and a 2D CNN to extract features from various domains and dimensions respectively. It shows that proposed CNNs perform better than traditional support vector machine classifiers, and MDC-CNN improves performance over single-domain CNNs using individual features. Furthermore, MDC-CNN can accurately discriminate SZ from HC (Figure 26).

In [33], an MLP was proposed to perform Autism spectrum disorder (ASD)/ typical controls (TC) classification given individuals’ brain network as the feature representation. The authors first extracted connectivity features between each pair of

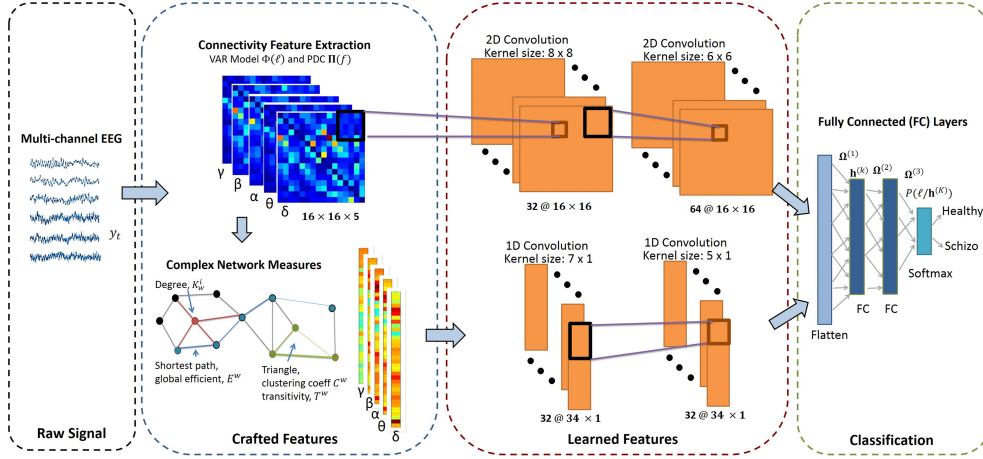


Figure 26: The architecture of the model proposed in [48].

different regions of interest (ROIs), which are seldomly considered in conventional methods. Subsequently, they selected the top 3000 features based on F-score, and finally, these features were fed into the model to predict. The results show that their method achieved an accuracy of 90.39% and the area under receiver operating characteristic curve (AUC) of 0.9738, which is better than some state-of-the-art methods in ASD/TC classification (Figure 27).

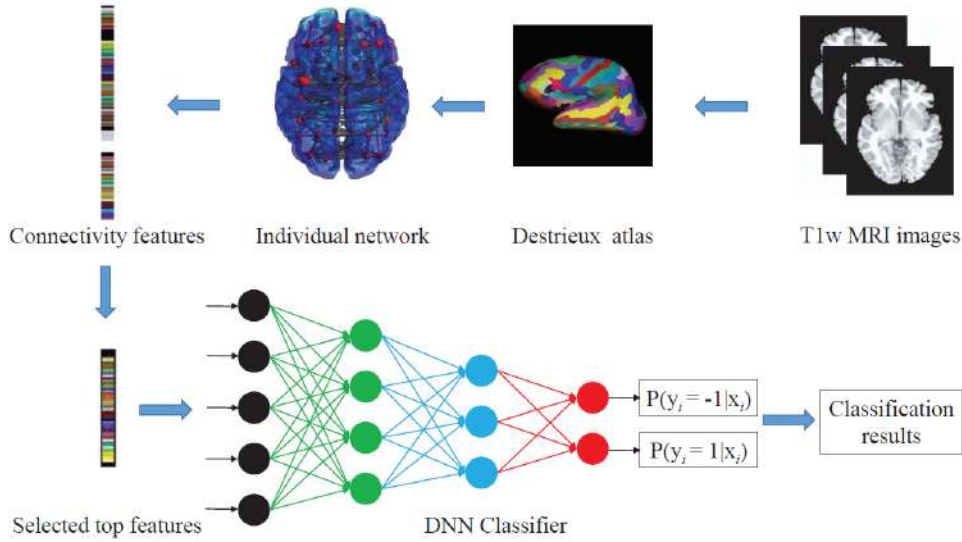


Figure 27: The architecture of the model proposed in [33].

In [36], the authors first converted the connectivity matrix to the labeled graph. Then they proposed a siamese graph convolutional neural network (s-GCN) to learn a graph similarity metric and apply the model on several datasets. The results prove that the method's performance for the classification tasks between matching and

non-matching graphs, and individual subject classification and manifold learning, is better compared to conventional methods (Figure 28).

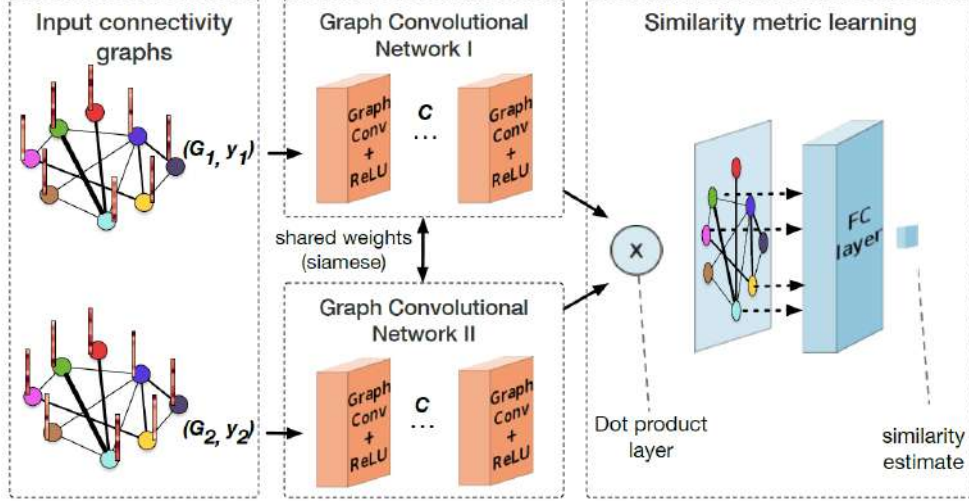


Figure 28: The architecture of the model proposed in [36].

[2] introduced a method that can seize changes in the brain based on dynamic functional connectivity (dFC). Four unique features are compared and combined that are: amplitude of low-frequency fluctuation (ALFF), regional homogeneity (ReHo), dFC, and the adjacency matrix of different brain structures between subjects, respectively. GCN is applied to classify non-Euclidean domains. The proposed method's accuracy was 91.3%, and the area under the receiver operating characteristic curve was 98.4%. This result demonstrated that the proposed method could be used for detecting AD.

In [39], the authors proposed a graph representation learning method to model the connection between brain functional connectivity and structural connectivity using a graph encoder-decoder system. The GCN model in the encoder part extracts interactions between pairwise brain regions in the node embeddings. Plus, the model learns node representations for functional connectivity network reconstruction and graph representations for subject classification by leveraging a particular loss function. The subjects of this experiment are a large number of non-drinkers and heavy drinkers. The results show that the model can provide a characteristic description of the population pattern in the functional connectivity and structural connectivity pairwise relationship. Furthermore, it proves GCN-based graph representation learning based on brain networks can be used to model human brain activity and

function.

In [9], the authors proposed a deep learning method to identify multivariate and nonlinear functional connectivity patterns of ASD. The dataset's size is 972 subjects (465 ASD and 507 healthy controls), which was first preprocessed into functional connectivity matrices with 90 predefined brain regions. Subsequently, variational autoencoder (VAE) concluded the functional connectivity matrices into two features, and patient group was then compared to control group. The results indicate that the feature had noticeable differences between ASD and healthy controls. It shows that feature extraction methods based on deep learning could be applied to recognize complicated modes of functional connectivity of ASD.

Moreover, [80] pointed out that to seize the complex relationships and exploit the graph structure of brain networks is still a challenge. Furthermore, it proposed a method to extract the graph embeddings based on GCN and used an MLP for the final regression output. The results prove that this way gives better performances on clinical depression scores than related baselines (Figure 29).

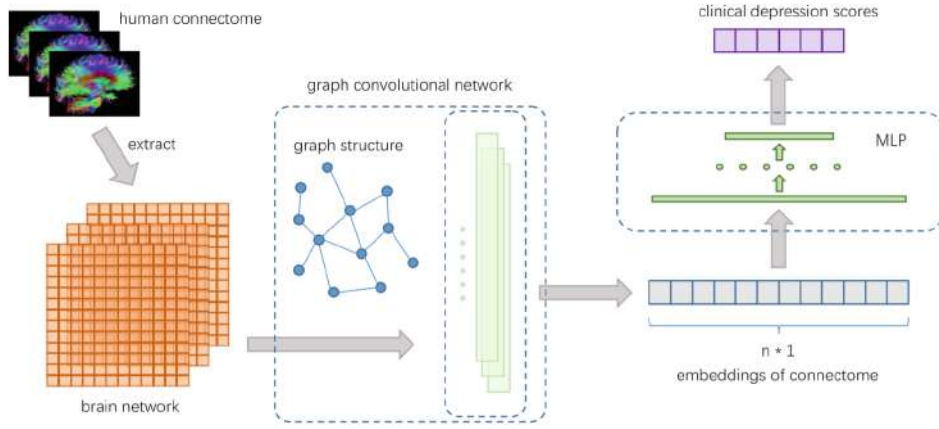


Figure 29: The architecture of the model proposed in [80].

[4] proposed a GCN based visual attribution method to perform a sex classification task given brain connectivity. It demonstrated that the method could be used to identify brain regions without prior node labels, and it is robust in highlighting reproducible regions across individuals.

[7] proposed a deep learning model to predict the survival of Amyotrophic lateral sclerosis (ALS) patients given clinical characteristics combined with MRI data. With

135 ALS patients (83 for training, 20 for validation, and 32 for testing), the model given clinical characteristics as input data performed with an accuracy of 68.8%, 62.5% given MRI structural connectivity, and 62.5% given brain morphology data. Remarkably, the accuracy rose to 84.4% when three sources were mixed as input. The results show that MRI based brain connectivity helps with survival prediction in ALS, and deep learning has a promising future in disease prognostication.

In [73], the authors proposed a method, named Deep neural network (DNN) with Autoencoder (AE) pre-trained Functional connectivity Analysis (DAFA), to track changes in functional connectivity in fMRI during brain aging. The results prove that the proposed DAFA method was better than conventional FC-determining methods in detecting changes in brain functional connectivity during aging.

[35] proposed a metric learning method to measure the distance between graphs that uses CNNs, meanwhile taking knowledge from spectral graph theory to operate on some irregular graphs. The results show that the method performs 11.9% better in a k-nn classifier than a traditional distance metric (Figure 30).

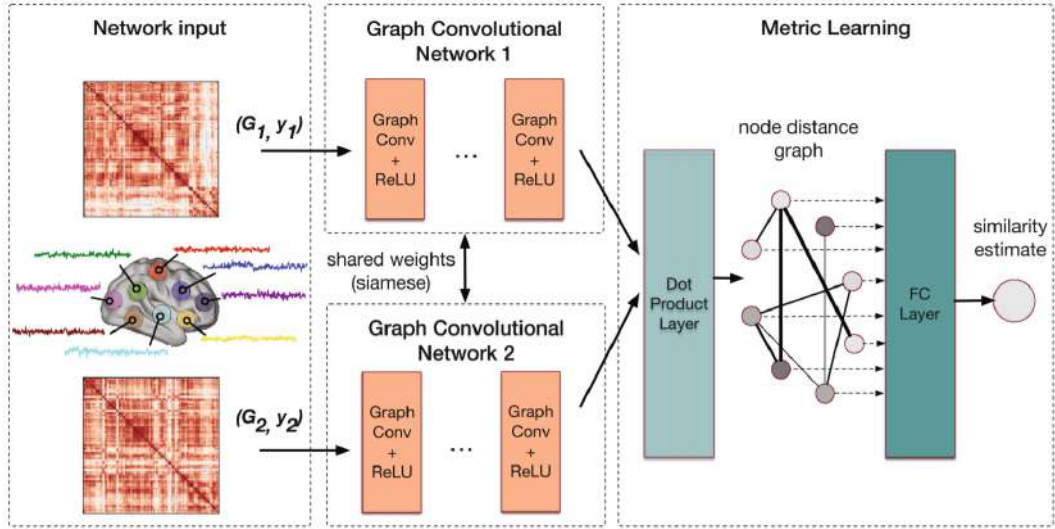


Figure 30: The architecture of the model proposed in [35].

[64] proposed an architecture combining deep learning and state-space modeling and used it on rest state-fMRI (rs-fMRI) based Mild Cognitive Impairment (MCI) diagnosis. Firstly the authors applied a Deep Auto-Encoder (DAE) to convert the brain regional features into an embedding space. Then they used a Hidden Markov Model (HMM) to estimate dynamic characteristics of functional networks inherent

in rs-fMRI via internal states with the embedding space. They experimented on two unique datasets and achieved accuracies of 72.58% and 81.08%, respectively, which outperformed other methods.

[42] proposed a CNN-based framework with adaptive graph transforms to predict diagnostic labels and identify disease-based brain connectivity. The authors experimented on two datasets. The results demonstrate that their framework outperforms the traditional graph methods and gains better accuracy in predicting diagnostic labels and identifying brain connectivities of diseases like Attention-Deficit / Hyperactivity Disorder (ADHD) and AD (Figure 31).

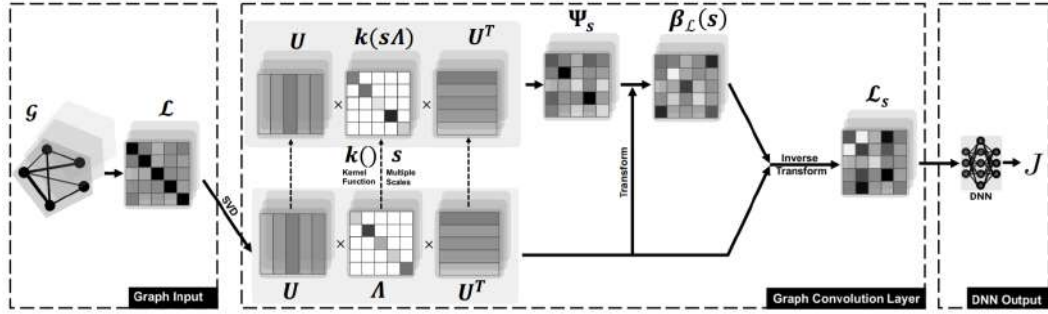


Figure 31: The architecture of the model proposed in [42].

4 Methods

This chapter mainly covers the process of the experiment. The first part is about the experiment setup and the collection of data. Subsequently, it demonstrates the details of processing data, including what different connectivity network measures are applied as input data and visualization of the output data.

4.1 Experiment Setting and Input Data Collecting

The original data was collected and provided by the Institute of Medical Psychology, Otto-v.-Guericke University of Magdeburg Medical Faculty. The experimental setting was as follows: 24 hemianopic patients with occipital ischemic stroke (age: 58.4 ± 10.9 yrs., lesion age > 6 months) and 24 age-matched health controls ($n = 24$, 57.4 ± 10.5 yrs.) were recruited in the experiment. They were given ID from 1 to 24 for both patients and controls group (for example, patient 1, control 4). The lesion situations and visual problems for the patient group are given in Table 3. Ten patients lost their right vision, and fourteen lost their left vision.

ID	Side of vision loss	ID	Side of vision loss	ID	Side of vision loss
1	Right	2	Right	3	Right
4	Left	5	Right	6	Right
7	Left	8	Left	9	Right
10	Right	11	Left	12	Left
13	Left	14	Left	15	Left
16	Left	17	Right	18	Left
19	Right	20	Right	21	Left
22	Left	23	Left	24	Left

Table 3: Vision loss situation of patient group

High dense array EEG was recorded using a HydroCell GSN 128-channel net and a Net Amps 300 amplifier (EGI Inc., Eugene, Oregon, USA) (sample frequency: 500 Hz; impedance < 50 k Ω). Five-minute long resting-state EEG per subject was recorded under eye-closed and no-task condition. Later, the EEG recordings were segmented into 2 seconds, epochs overlapping 0.5s, with comparable clean trials count for controls (123.6 ± 20.1) and patients (120.1 ± 21.2). Components of eye-

blinks or cardiac activity of health controls (3.7 ± 1.4 , mean \pm SD) and patients (5.9 ± 3.1 , mean \pm SD) were removed by independent component analysis (ICA); The signal was then decomposed to Delta (1 - 3Hz), Theta (4 - 7Hz), Alpha1 (8 - 10Hz), Alpha2 (11 - 13Hz), Beta1 (13 - 21Hz), Beta2 (22 - 30Hz) and the total alpha (8 - 13Hz) frequency bands. The automatic anatomical labeling (AAL)-VOIs atlas [67] was applied to every subject.

Functional connectivity network (FCN), or functional network can quantify the interaction between different pairs of brain regions with the imaginary part of coherence, a method proposed in [66], which is insensitive to false connectivities arising from volume conduction. The mathematical definition of coherence is:

$$icoh_{(f,t)} = |im(\frac{\sum_{n=1}^N S_1^n(f,t)S_2^{n*}(f,t)}{\sqrt{\sum_{n=1}^N |S_1^n(f,t)|^2 \sum_{n=1}^N |S_2^n(f,t)|^2}})| \quad (15)$$

where $S_1^n(f,t)$ and $S_2^{n*}(f,t)$ are frequency-decomposed EEG data from two specific regions for every subject and condition. Followingly, a parcellation scheme with the AAL atlas [54] was adopted to reduce the dimension in the connectivity matrix, and the new connectivity matrix (116 * 116) network measures were produced. The coherence matrices represent "ROIs * ROIs" for all pairs at the source level.

Figure 32 (vertical view) shows a coherence connectivity network of a control subject: the nodes indicate 116 defined regions and the links between two nodes indicate the coherence values. The higher the values are, the more red the color is.

4.2 Output Data Collecting

Binocular visual fields of the patients were tested by a special high-resolution computer-based campimetric test battery (High Resolution Perimetry (HRP)), which was developed by the Sabel laboratory [57]. The patients sat in front of an experimentally used monitor in a dim room. The head position was fixed firmly by a head-chin rest. The distance between the eyes and the monitor was adjusted to be about 40 cm. See figure 33. White-light stimuli were displayed in a grid of 21 * 21 stimulus locations with each location indicating 1.2 ° of the central visual field. The stimulus appeared in random order. The central reference point was positioned at the center

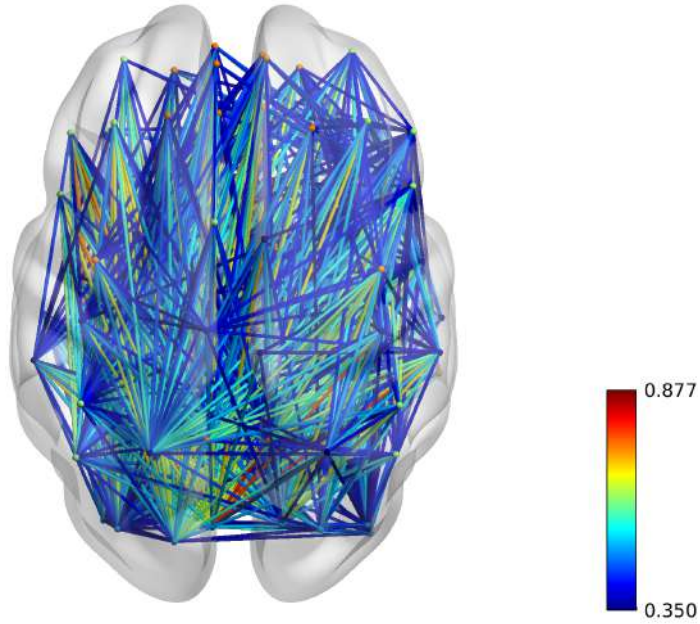


Figure 32: An example of health control’s coherence connectivity network

of the screen, used to adjust the screen to eye level. During the experiment, patients should keep looking at the central reference point and press the ‘space’ button whenever a target stimulus is seen or the color of the reference point changes.

There are three experiment stages for the patients which are: the day before treatment (pre), the day after 10-successive-day treatment (post), and two months after the completion of the whole treatment (follow-up). For each stage, the patients were required to be recorded the same set of data: EEG data and matching HRP vision distribution.

4.3 Data Processing

As aforementioned, nodes could represent brain regions, and edges represent the synchronization between two regions at the anatomical level. In this thesis, the brain regions are defined by the AAL atlas [54]. We calculated the modularity per density (starting from 10%, increasing by 10% each time, until 100%: 10 densities in total) to keep the network patterns diverse. In this case, the non-arbitrary and better network grouping were retained, and also the weak and noisy connectivities were removed. An example visualization of the coherence matrix is figure 34, and the features of all the matrices are: left symmetric, and the left diagonal values are zeros. In this figure, the higher the weight is, the darker the pixel shows.



Figure 33: A woman is taking a vision test [12]

Apart from the coherence, some other network measures are also applied in this thesis. The first one is **degree**, which is basic yet important. Its definition is: the degree of an individual node is equal to the number of links connected to that node [54]. The degree of a node i is shown in Equation 16. The degrees of all nodes in the connectivity network can be presented as a one dimension vector with the size of [1, 116].

$$k_i = \sum_{j \in N} a_{ij} \quad (16)$$

The second measure is **strength**, which means the sum of weights of links connected to the node [54]. The strengths of all nodes in the connectivity network can be presented as a one dimension vector with the size of [1, 116].

The third measure used in this thesis is **clustering coefficient**, which is the fraction of triangles around a node and is equivalent to the fraction of node's neighbors that are neighbors of each other [72]. Its mathematical presentation is:

$$C = \frac{1}{n} \sum_{i \in N} C_i = \frac{1}{n} \sum_{i \in N} \frac{2t_i}{k_i(k_i - 1)} \quad (17)$$

where C_i denotes the clustering coefficient of node i ($C_i = 0$ for $k_i < 2$). Clus-

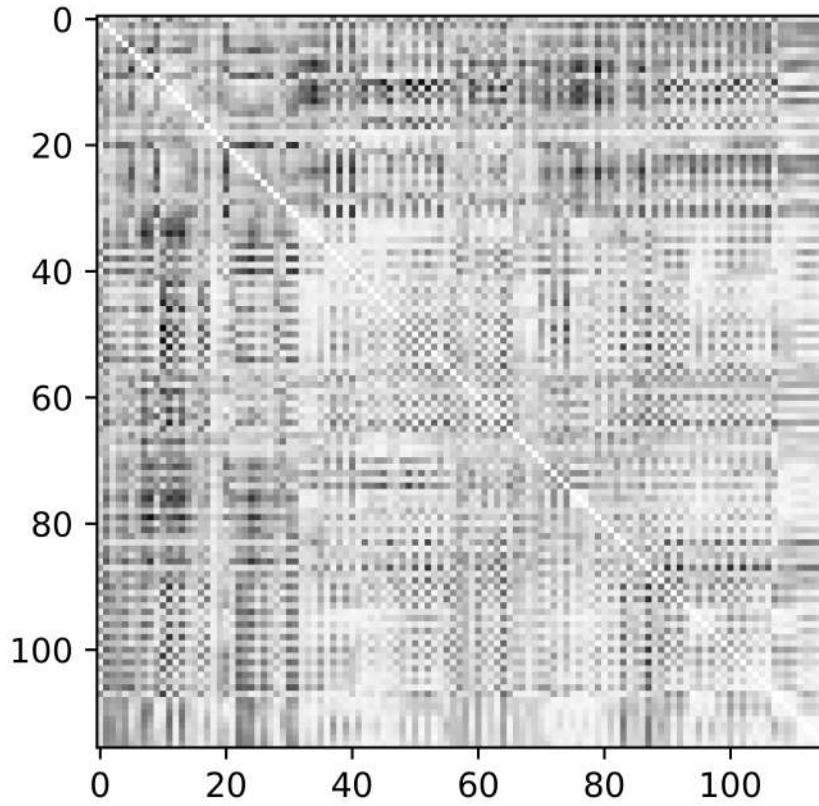


Figure 34: Patient ID = 1, pre-treatment stage, frequency = 1Hz

tering coefficient is a measure of functional segregation, and measures of functional segregation have direct explanations in both anatomical and functional networks. The clustering coefficient reflects the popularity of clustered connectivity (triangle) around single nodes.

The last measure is one of the measures of centrality, and it is called **betweenness centrality**. Its definition is the fraction of all shortest ways in the whole network, including the known node. From this measure, it could be inferred that a high value of betweenness centrality of a node indicates that it is involved in extensive shortest paths. As it is seen in Equation 18 [19], where ρ_{hj} denotes the number of shortest paths between h and j , and $\rho_{hj}(i)$ is the number of the shortest paths between h and j passing through i . In conclusion, the data shapes of these features are shown in Table 4.

$$b_i = \frac{1}{(n-1)(n-2)} \sum_{\substack{h,j \in N \\ h \neq j, h \neq i, j \neq i}} \frac{\rho_{hj}(i)}{\rho_{hj}} \quad (18)$$

Measure	Format
Coherence	[116, 116]
Degree	[1, 116]
Strength	[1, 116]
Clustering Coefficient	[1, 116]
Betweenness Centrality	[1, 116]

Table 4: Format of each measure for an instance

The output data is produced by HRP, and it's a matrix with the size of 21 * 21. There are three vision conditions in the grid:

- White: intact vision
- Grey: partial functional vision
- Black: impaired vision

After the stroke, the black areas are usually permanent, which means that those areas' vision cannot be restored or recovered. The grey areas are likely to be restored with some treatment (Figure 35, Figure 36, and Figure 37). The changes in the vision of these three patients could be seen in Table 5. All three patients had improvements in their vision recovery.

Patient ID	No. of Black	No. of Grey	No. of White
2	8	-18	10
13	10	-23	13
17	-1	-44	45

Table 5: Positive values mean increment, and negative values mean decrement.

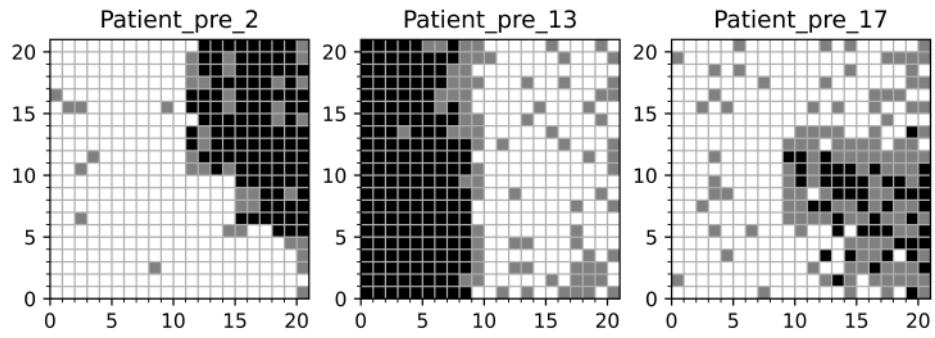


Figure 35: Patient ID = [2, 13, 17], pre-treatment stage

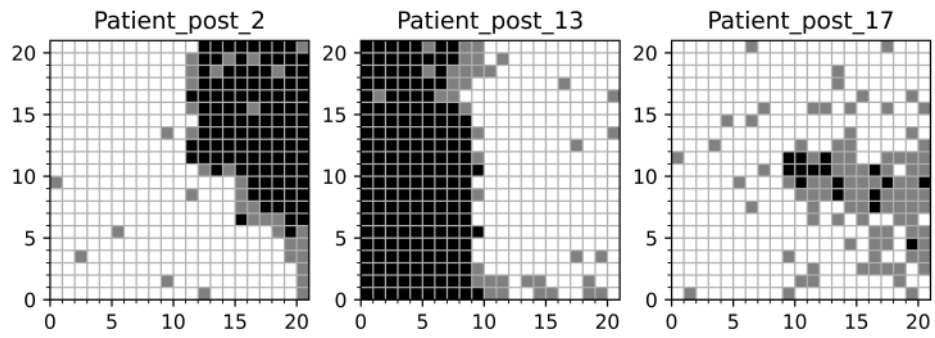


Figure 36: Patient ID = [2, 13, 17], post-treatment stage

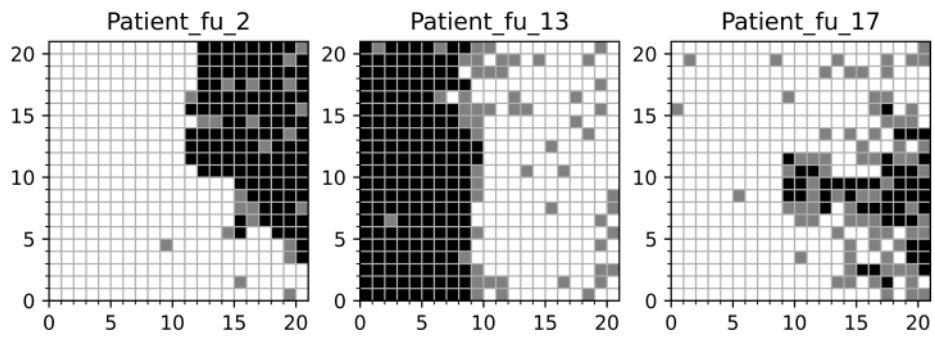


Figure 37: Patient ID = [2, 13, 17], follow-up stage

ID	Label	ID	Label	ID	Label
1	0	2	1	3	0
4	1	5	1	6	1
7	0	8	0	9	0
10	1	11	1	12	1
13	1	14	1	15	1
16	1	17	1	18	1
19	1	20	0	21	1
22	0	23	0	24	0

Table 6: Results after treatment: 0 means non-improved, 1 means improved

4.4 Research Questions

4.4.1 Question No. 1

Is it possible to predict if the patients will have improvements in vision before they take the treatment, given their coherence matrix data?

It is a binary classification task with two labels: improved or non-improved. Table 6 shows the patients' results, whether they had improvements or not after treatment. 15 out of 24 patients had positive results, and the rest nine patients had no improvements, and this data was the ground truth.

Each patient had 50 trials of input data with the format of $[30, 116, 116]$ and output labels '1' for improved or '0' for non-improved. Due to this input data format, I designed, compared, and updated several 3D CNN models to validate my hypothesis. The details of the best performance model are given in Table 7. Before feeding the data to the model, I shuffled data based on their ID to ensure that every patient's data could only appear in training data or testing data alone.

The input data was fed into the first Conv3D layer with a kernel size of $[2, 3, 3] * 32$ and a max-pooling operation with pooling size of $[2, 2, 2]$, which shrank the feature map to the half of the original ones. After another two Conv3D layers, the feature map size was $[2, 12, 12, 32]$. Followingly, the feature map was flattened into a 1D vector with 9216 values. The vector was then fed into two fully connected layers with 1024 units and 512 units successively. Finally, in the output layer, one output label was predicted through a sigmoid activation function. For all the other layers above, ReLU was applied as the activation function.

A k fold validation method ($k = 4$) was applied to examine the model's average performance. Figure 38 displays the training accuracy and loss. The training epoch was 30 and the accuracy of the test data is 95%, which proves my hypothesis that it is possible to predict if the patients will have improvements in vision before they take the treatment, given their coherence matrix data.

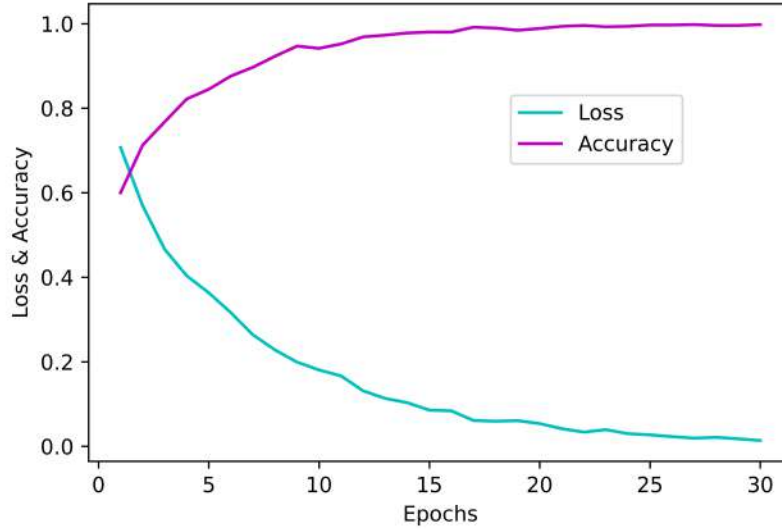


Figure 38: Loss and Accuracy during training process of the proposed model for hypothesis 1

Input Layer	Input_shape = [30,116,116,1]				
Layer	Filter	Dropout	AF	Pooling	Norm?
Conv3D 1	[2,3,3] * 32	0.3	ReLU	Max	Yes
Conv3D 2	[2,3,3] * 64	0.5	ReLU	Max	Yes
Conv3D 3	[2,3,3] * 32	0.5	ReLU	Max	Yes
Flatten Layer					
Dense 1	Unit = 1024		ReLU		
Dense 2	Unit = 512		ReLU		
Output Layer	Unit = 1		Sigmoid	10,038,209 Paras	

Table 7: Parameters and details of the proposed 3D CNN model for hypothesis 1

4.4.2 Question No.2

Is it possible to predict if the subject is a health control or a stroke patient given their coherence matrix data?

The output label for this hypothesis is 1 for controls and 0 for patients. The three stages of patients: pre-treatment, post-treatment and follow up period were regarded as different subjects. Thus there were 4800 instances in total: 1200 controls (label: 1), 1200 pre-treatment pieces (label: 0), 1200 post-treatment pieces (label: 0) and 1200 follow-up pieces (label: 0). All the data was fully shuffled, and three quarters (3600) were used for training and one quarter (1200) for testing. For the 3600 training instances, a five fold cross-validation was operated when trying out models. In the end, a three-layer model (Table 8) providing the best average performance was selected: the accuracy of training is 85%, and that of validation is 76%. Furthermore, for the test accuracy, it is 72%. The training loss and accuracy is shown in Figure 39.

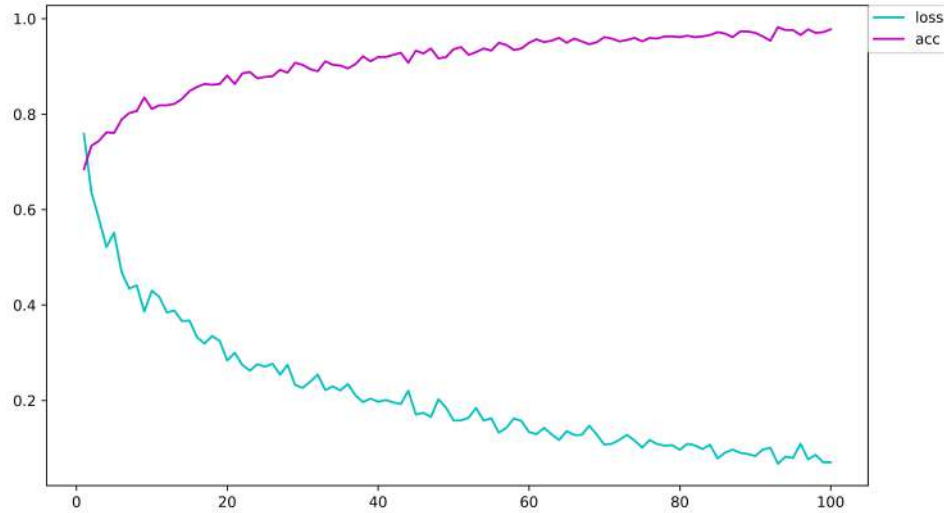


Figure 39: Loss and Accuracy during training process of the proposed model for hypothesis 2

4.4.3 Question No.3

Is it possible to predict a patient's percentages of vision condition given their coherence matrix data?

There are three vision conditions for stroke patients: intact vision ($C1$: using 2 to represent), partially damaged vision ($C2$: using 1 to represent) and impaired vision ($C3$: using 0 to represent) as shown in Figure 35. It is clear to see and calculate the

Input Layer	Input_shape = [30,116,116,1]				
Layer	Filter	Dropout	AF	Pooling	Norm?
Conv3D	[3,3,3] * 16	0.5	ReLU	Max	Yes
Conv3D	[3,3,3] * 32	0.5	ReLU	Max	Yes
Conv3D	[3,3,3] * 32	0.5	ReLU	Max	Yes
Flatten Layer					
Dense 1	Unit = 1024		ReLU		
Output Layer	Unit = 1		Sigmoid	9,481,537 Paras	

Table 8: Parameters and details of the proposed 3D CNN model for hypothesis 2

percentage of each vision condition (Equation 19). The vision distribution is a 21 * 21 matrix, thus there are 441 values in total, and the sum of the three percentages is 100%. Due to the very few numbers of $C2$, $C2$ and $C3$ were merged as single one condition for better experiment performance. Thus, only two conditions were considered in this thesis.

$$Ci_percentage = \frac{Ci}{441} \quad (19)$$

Fourteen patients had their stroke occur in their right brains, and ten patients had their stroke in their left brains. To keep the consistency of the brain connectivity and augment the attributes of connectivity, right stroke patients' data was swapped in a pair-wise way (Figure 40). Every two rows and every two columns swap their elements to form a new matrix, this operation was applied to all the right stroke patients and in this thesis, the term 'changed' indicates the matrix data after swapping.

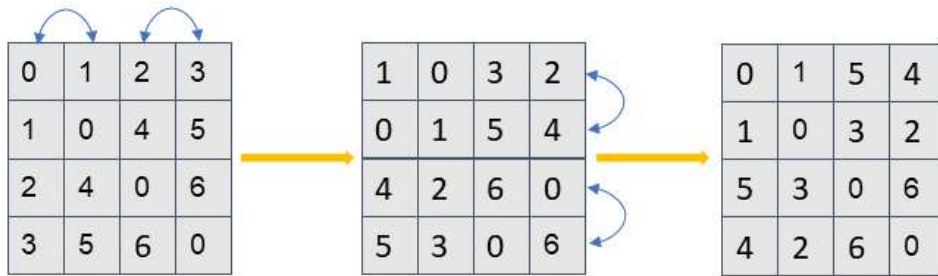


Figure 40: Swap operation of the matrix data

Due to the format features of the input data, I designed and conducted three groups of experiments: 2D CNN based methods, 1D CNN based models and 3D CNN base

models.

- Experiment 1: 1D CNN based methods.

The Input shape $[30, 116, 116]$ is an adjacent matrix that is symmetric along the diagonal. If considering only pure information, only half of the matrix is useful. Thus, I kept only half of the matrix and reshaped it to $[30, 6786]$ (removing the diagonal and the upper half of the matrix (Figure 41). In this way, the input data could be flattened and regarded as 30 1D vectors (with a length of 6786) for each instance.

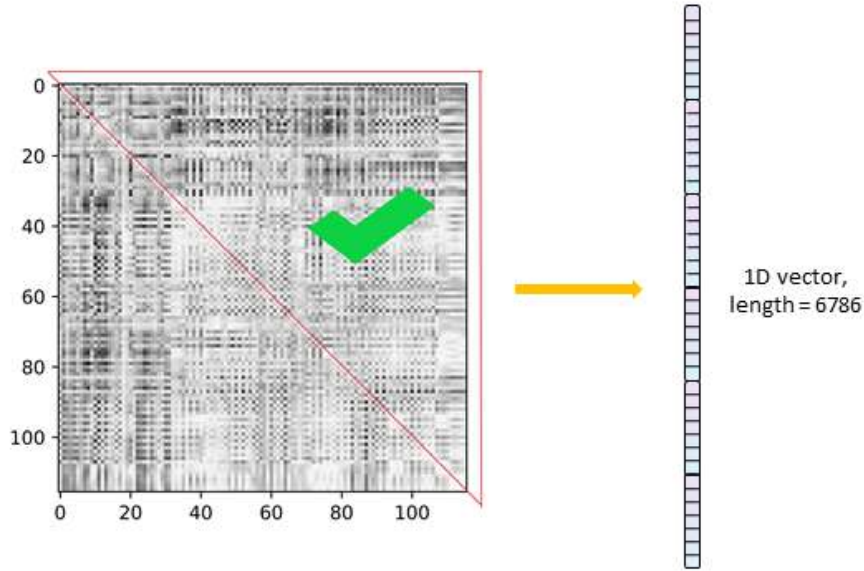


Figure 41: Swap operation of the matrix data

The 30 vectors were fed into the model and convolved separately. After several convolutional layers, the shortened vectors were flattened and fed to the several fully connected layers until the final output. Figure 42 is the general structure of the model.

The performance of the 1D CNN based model is not ideal. The MSE and MAE were adopted as the training loss and metric, and several gradient descent optimization algorithms from Adam, SGD to RMSprop were applied in turns. See Figure 43 for the training loss and epochs, Figure 45 shows comparisons between predictions and actual values for training dataset and Figure 44 shows comparisons between predictions and actual values for testing dataset. After many trials, I could not

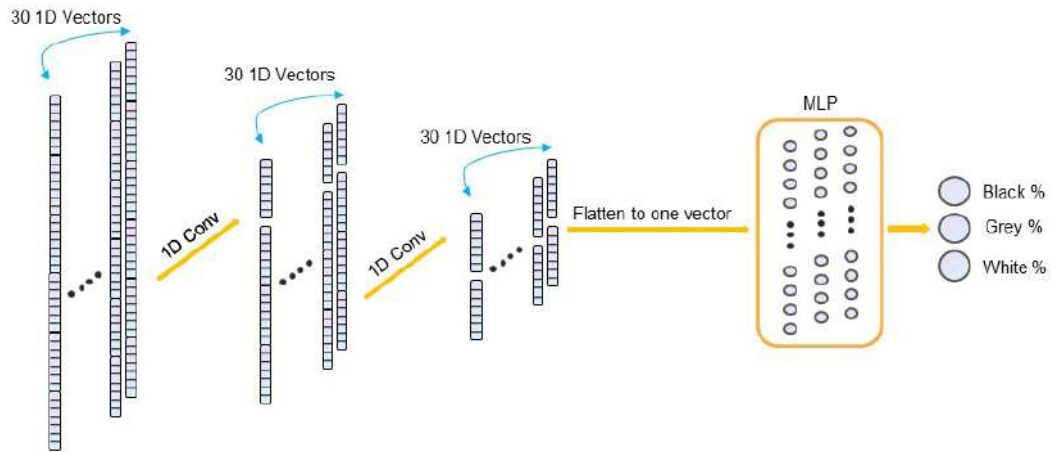


Figure 42: The architecture of the 1D CNN model

build a model that gives a good performance. The model was easily underfitting or overfitting.

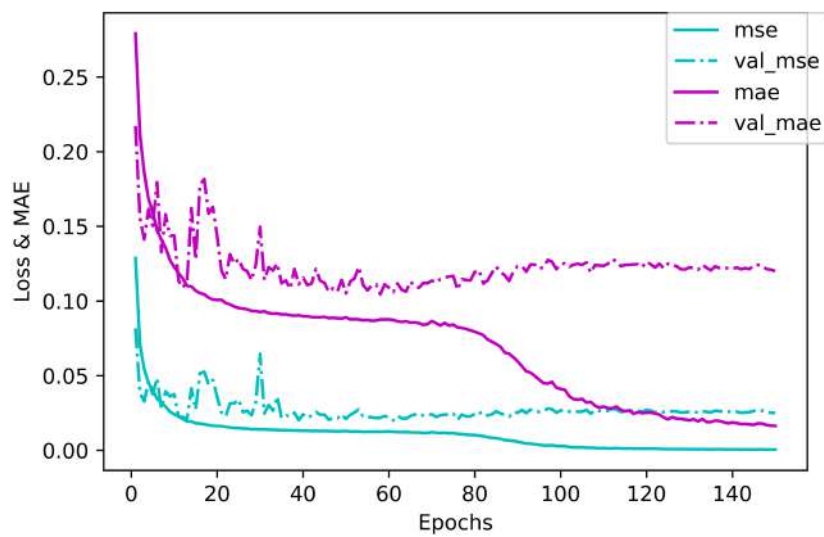


Figure 43: Loss and MAE during training process for 1D CNN model.

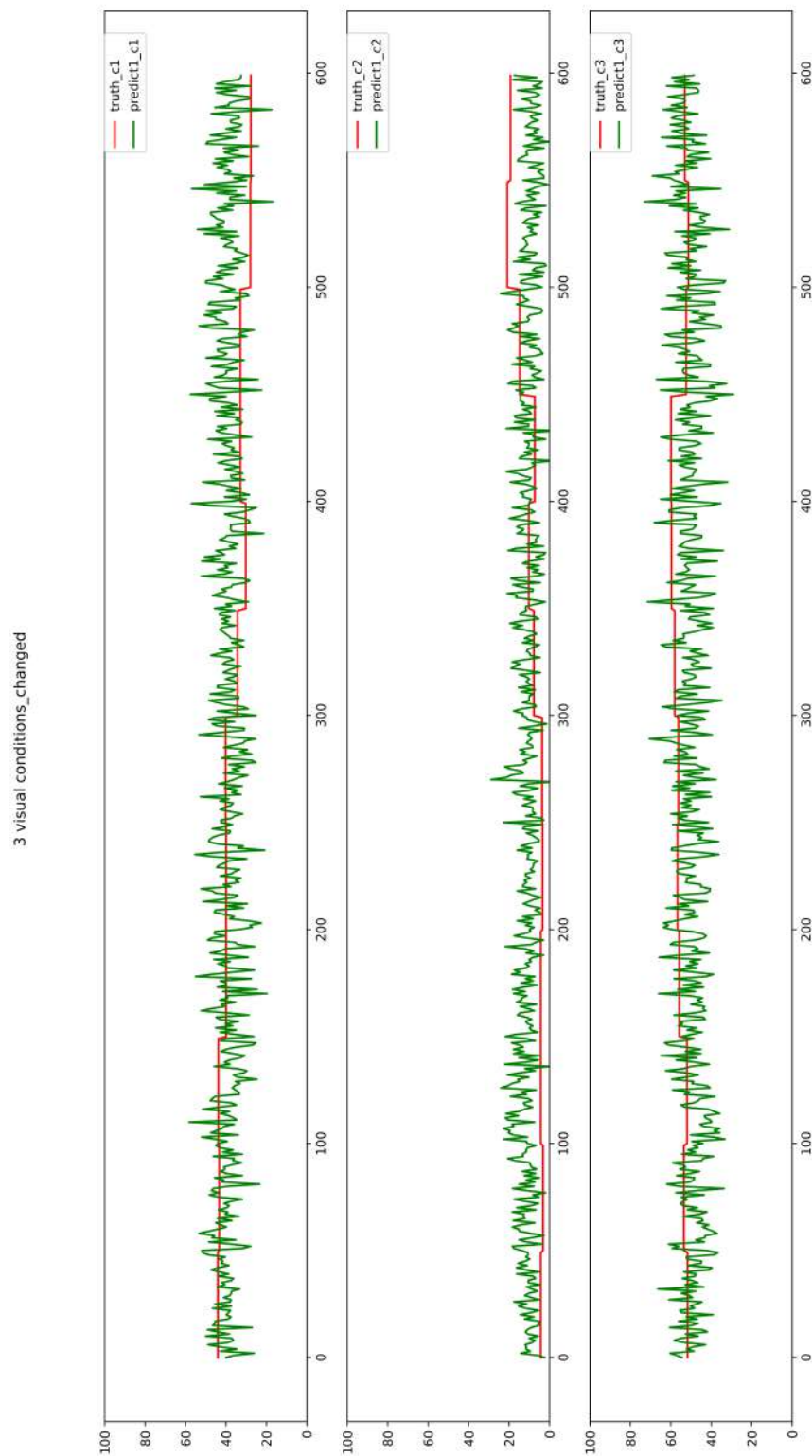


Figure 44: Comparisons of three vision conditions between predictions and ground truth on testing data.

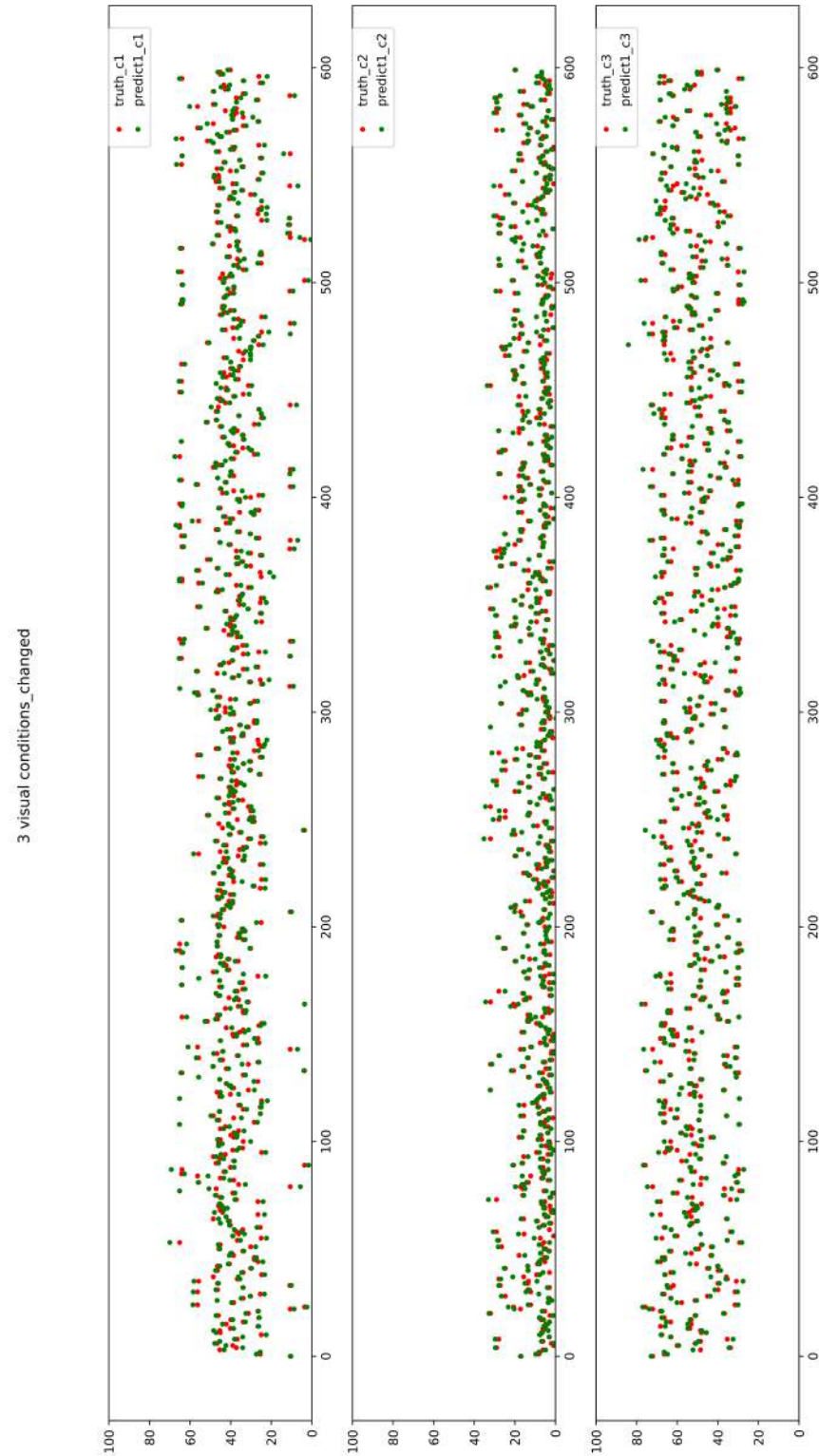


Figure 45: Comparisons of three vision conditions between predictions and ground truth for on training data.

- Experiment 2: 2D CNN based methods.

The input shape [30, 116, 116] could be further compressed to [3, 116, 116] based on the frequency band division: Delta from 0 to 4 Herz, Alpha from 4 to 12 Herz and Beta from 12 to 30 Herz. I did addition and average operations for these three frequency bands. In this way, the input data could be regarded as the standard image format with RGB color channels.

Since the compressed format is the standard image format, there are many mature deep learning models for image analysis that can be applied to validate my hypothesis.

Model	No. of Parameters
VGG16	19,960,130
ResNet50	57,668,994
Xception	54,942,762
DenseNet121	17,001,538
MobileNet	13,192,898

Table 9: Parameters of the applied models

VGG16, Xception [24], DenseNet121 [10], ResNet50 [27] and MobileNet [26] are the classic models that I deployed in this thesis. I kept the most part of these models, only freezing the top layers. I also added fully connected layers to make it fit for my own purpose. The parameters of the models is shown in table 9. There were in total 3600 instances, and 3000 (20 patients) were training data, and 600 (4 patients) were testing data. The training epoch was set to 75, and the batch size was 128. Before feeding the data into all the models, a shuffle operation was executed. MSE was the loss function, and MAE was the evaluation metric. During training process, the loss decreased fast, so did the MAE (Figure 46).

From the figure, we could notice that the MSE and MAE of VGG16 model are less competitive than other models' counterparts. Figure 47 shows the predictions of these five models on testing data, which was obviously inadequate. Figure 48 shows the VGG16 model's performance and Xception model's performance on training data (the rest three models have almost the same performance as Xception). VGG16 model is underfitting, and the rest four models are overfitting. Considering the number of parameters, MobileNet was chosen as the backbone to be further fine-tuned.

Based on MobileNet, many combinations were tried out, from different optimizers, learning rate, the number of layers, the number and size of filters, dropout rate, batch size to the number of epochs. After many trials, the model still could not provide an acceptable result. Several representative cases are posted in Figure 49. The model was either overfitting for training data or underfitting for training data, and more importantly, the model was always underfitting for testing data and validation data. The experiment shows that when the MSE was smaller than 0.0001, it could guarantee an R-squared of 0.9 (resulted from training session). In comparison, the MSEs for validation data and testing data were always greater than 0.001, which is ten times larger than the ideal MSE for training data, leading to negative scores for R-squared.

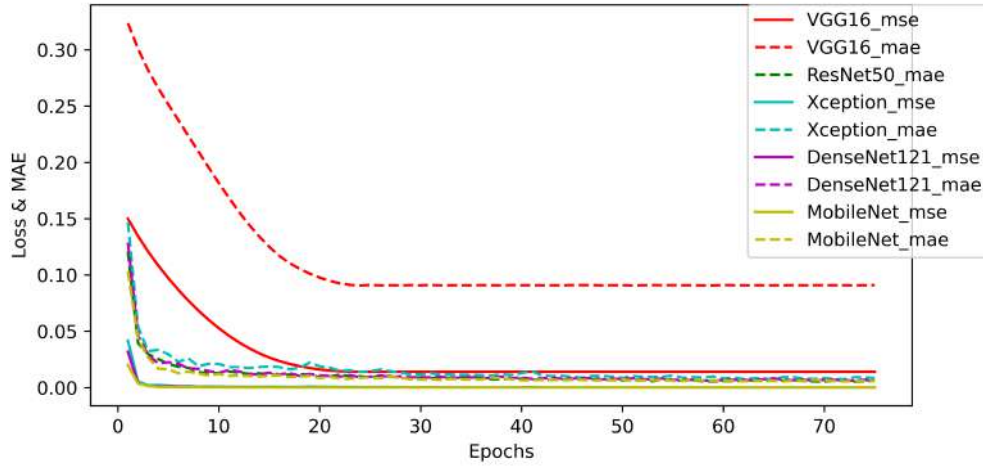


Figure 46: Loss and MAE during training process. Each model has a unique color representation as it is shown in this figure. The solid line indicates the loss (MSE), and the dotted line indicates the MAE.

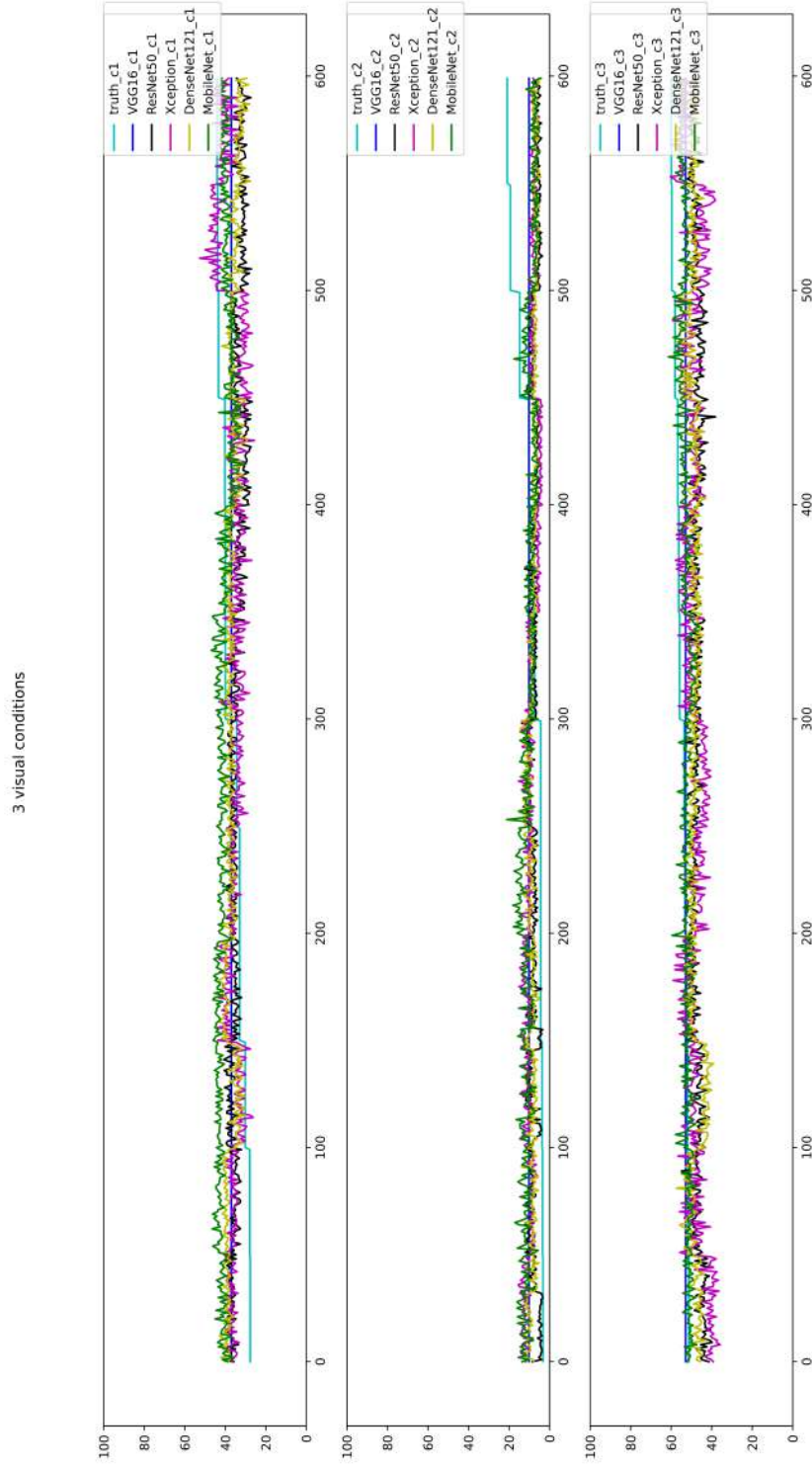


Figure 47: Comparisons of three vision conditions between predictions and ground truth for different models on testing data.

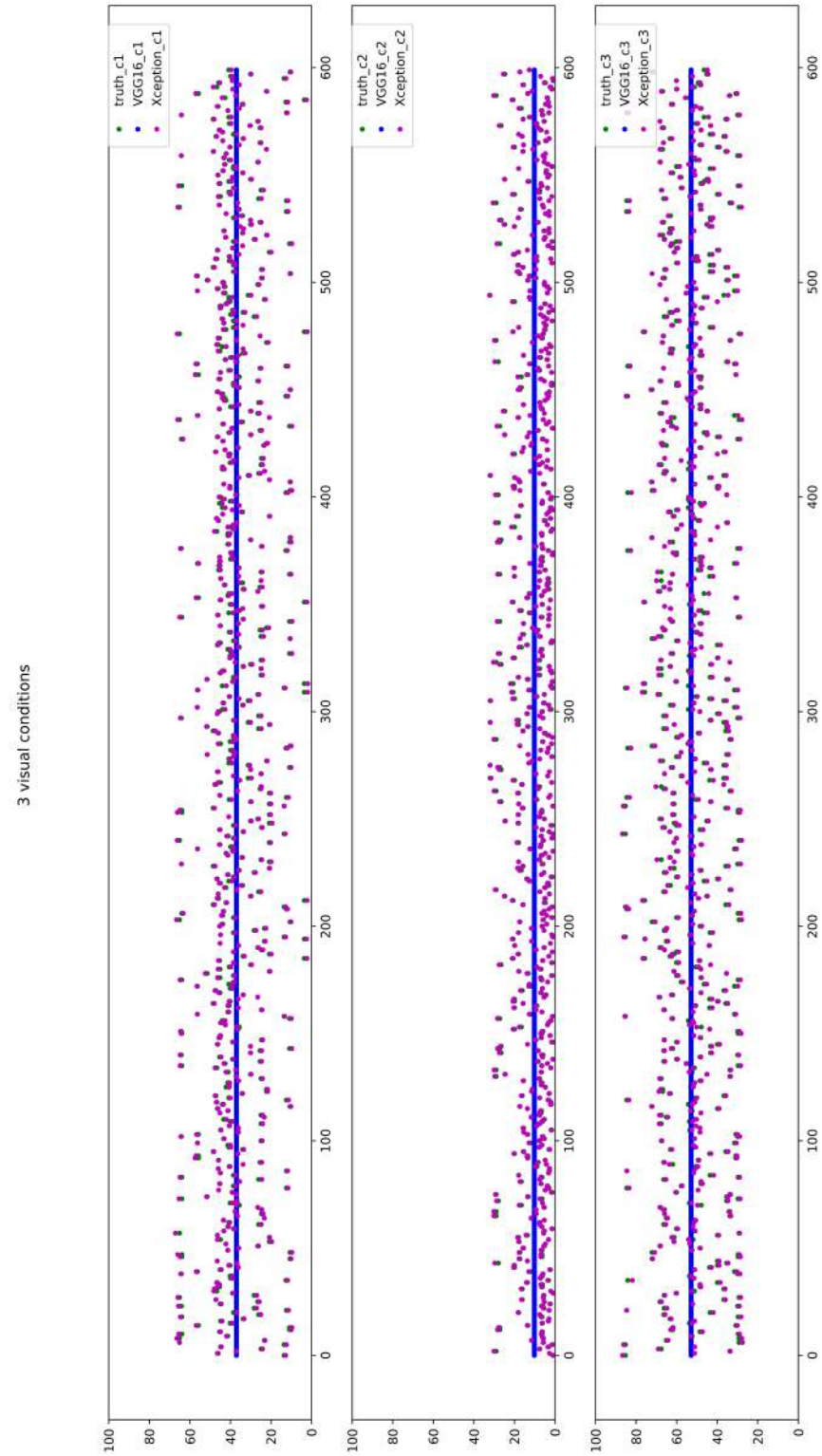


Figure 48: Comparisons of three vision conditions between predictions and ground truth for VGG16 model and Xception model on training data.

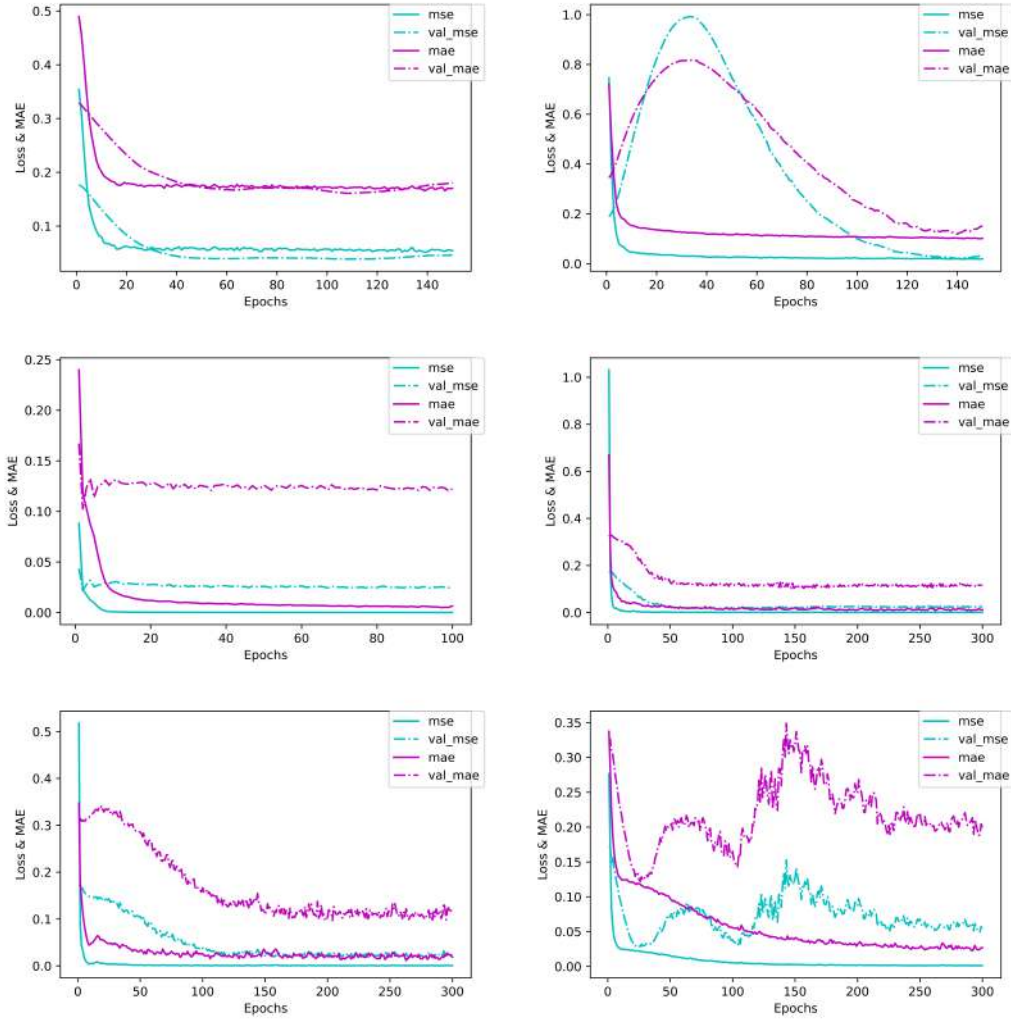


Figure 49: Performances of different cases of 2D CNN models.

- Experiment 3: 3D CNN based methods.

Due to the poor performance of the 2D CNN model, one possible cause of it might be that compressing 30 frequency bands to 3 bands would lose too much information. Thus, I kept the original the bands. In this way, the input data format is still [30, 116, 116], and 3D CNN models were designed and applied for this volumetric data. In experiments, 3600 (24 patients) instances were divided into two datasets: 3000 training data (20 patients) and 600 testing data (4 patients). 5 fold cross validation method was then applied on training dataset. Base on the average performance of different models, the best model was eventually selected for running the testing data. Figure 50 displays losses and MSEs of the training and validation data of some models.

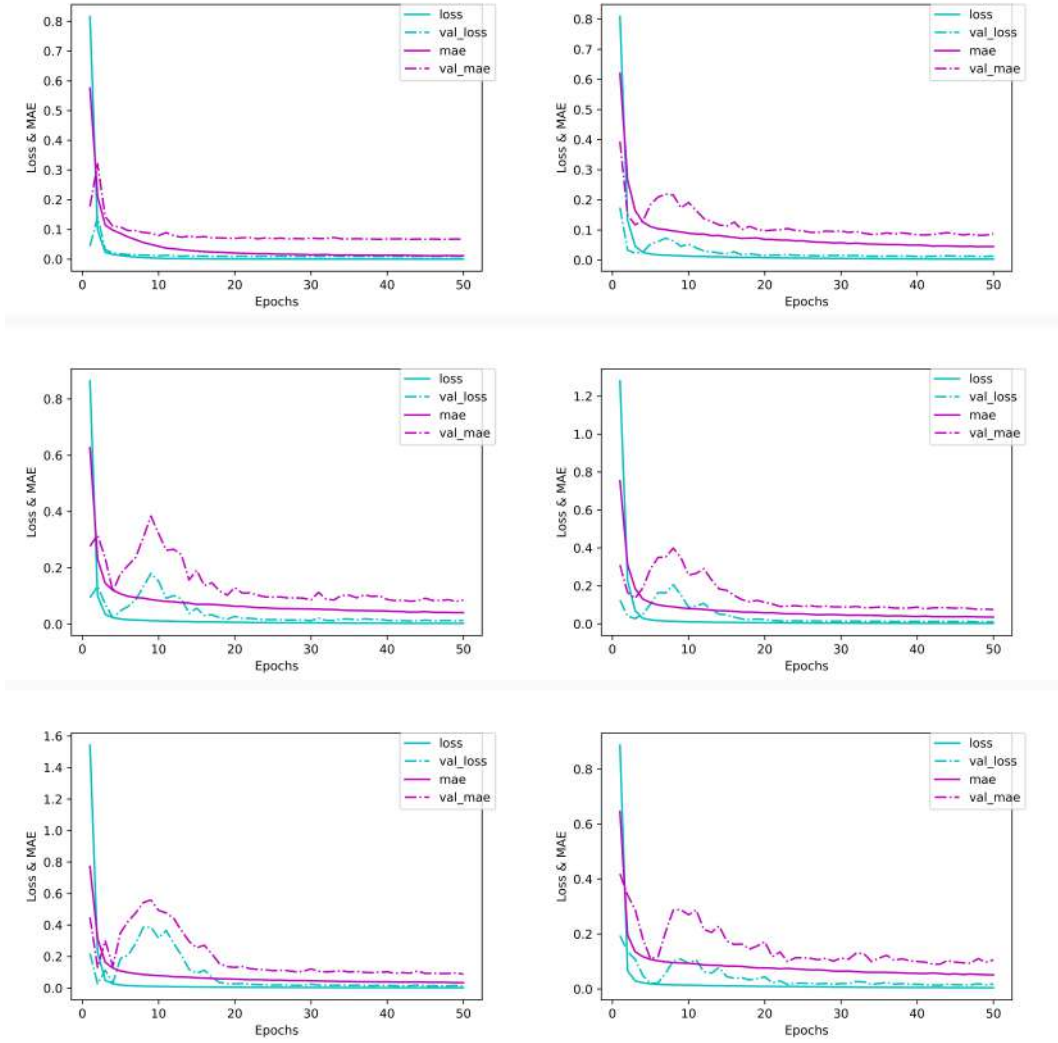


Figure 50: Performances of different cases of 3D CNN models.

4.4.4 Question No.4

Is it possible to predict the vision distribution for the stroke patient given their coherence matrix data?

The original vision distribution matrix is 21×21 , as shown in Figure 35, and to reconstruct the vision distribution based on the coherence matrix is a challenging task. Thus, The task was converted to the simple version: to make a less complicated vision distribution first. The 21×21 matrix was converted to a smaller size matrix: 2×2 , and each grid in the new matrix represented multiple grids in the original matrix (Table 10). For example, $M2[0, 0]$ represented the final result of the 100

Matrix: M1	[0:10, 0:10]	[10:21, 0:10]	[0:10, 10:21]	[10:21, 10:21]
Location	Left bottom	Right bottom	Left top	Right top
Matrix: M2	[0, 0]	[1, 0]	[0, 1]	[1, 1]

Table 10: How the original 21 * 21 matrix (M1) is converted to the new 2 * 2 matrix (M2)

grids in range $M1[0:10, 0:10]$, and the visual condition that had the highest number dominated that area. The rest three grids followed the same pattern. A direct display is given in Figure 51. The grey areas were still considered as black ones.

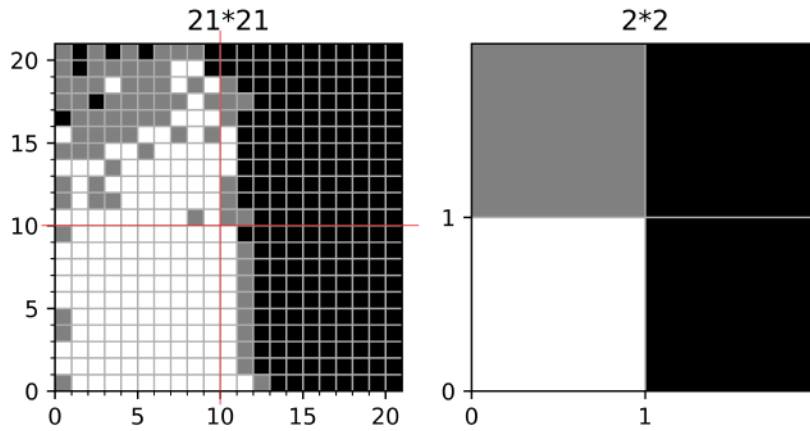


Figure 51: A visual comparison of the original and new matrices. The red line on the left segmented the field into 4 areas, which are corresponding 4 areas on the right.

- Experiment 1: 2D CNN based models

As described in Experiment 2 of Question 3, the same five classic 2D CNN models were applied and compared. Since this is a classification task, binary cross-entropy was chosen as the loss function, and accuracy was the metric. 3000 instances (20 patients) were training and validation data (2400 and 600 respectively), and 600 instances (4 patients) were testing data. The losses and accuracies of training data and validation data were displayed in Figure 52. According to the performances, the best result of pre-trained models is less than 70%. Overall, MobileNet was chosen as the backbone of the model to be further fine-tuned. Based on MobileNet, various combinations were conducted and compared, from different optimizers, learning rate, the number of layers, the number and size of filters, dropout rate, batch size to the number of epochs. The accuracies of the training and validation data ranged from

50% to 70%, and those of the testing data were slightly smaller. One of the best performances is shown in Figure 53, and its corresponding accuracy for testing data is 71.9%. Other models' performances are shown in Figure 54.

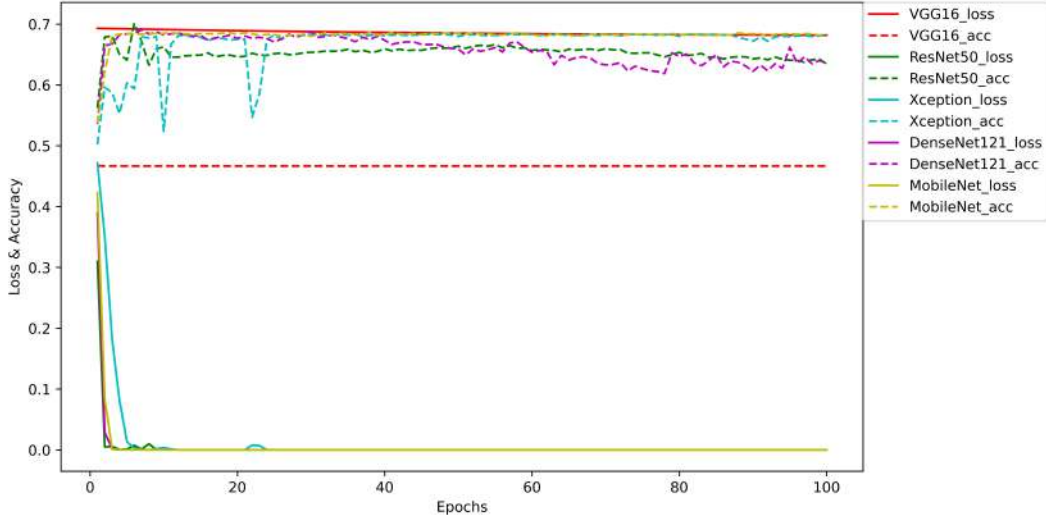


Figure 52: Loss and accuracy during training process. Each model has a unique color representation as it is shown in this figure. The solid line indicates the loss (binary cross-entropy), and the dotted line indicates the accuracy.

- Experiment 2: 3D CNN based models

In experiments, 5 fold cross-validation method was then applied on 3000 (20 patients) training data, and 600 (4 patients) instances were used as testing data. Manifold combinations with parameters and layers were applied and tried out. However, the performances were poor (Fig 55). The accuracies of models mainly ranged between 40% and 60% for both training and validation data, and accuracies for testing data were merely around 40 %.

- Experiment 3: GCN based models

The coherence data shape is $116 * 116$ and for each, which represents the coherence information among 116 brain regions. Each region has its coherence information with the other 115 regions. In this way, coherence data is symmetric along its diagonal, and the values fall on the diagonal are all zeros, which makes it an adjacent matrix data. The direct information about the region itself thus is neglected. The typical spectral GCN method is applied to deal with this drawback, as mentioned in Chapter 2.7.1. Equation 14 is the mathematical form. In this paper, the GCN model is based

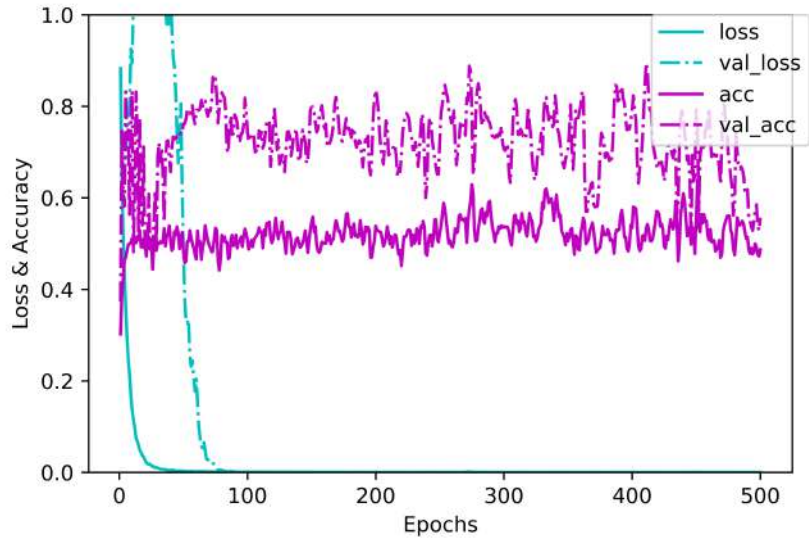


Figure 53: An example of the best performance model.

on this theory, using coherence data as the adjacent matrix and degree, strength, clustering coefficient and betweenness centrality data as the features. Together these two sets of data were fed into the GCN block to get the feature maps, which were further sent to the designed FC layers, as shown in Figure 56. However, the model's result is poor, with 60% to 70% accuracy on training data and 30% to 40% on testing data, which is much lower than the 2D CNN model.

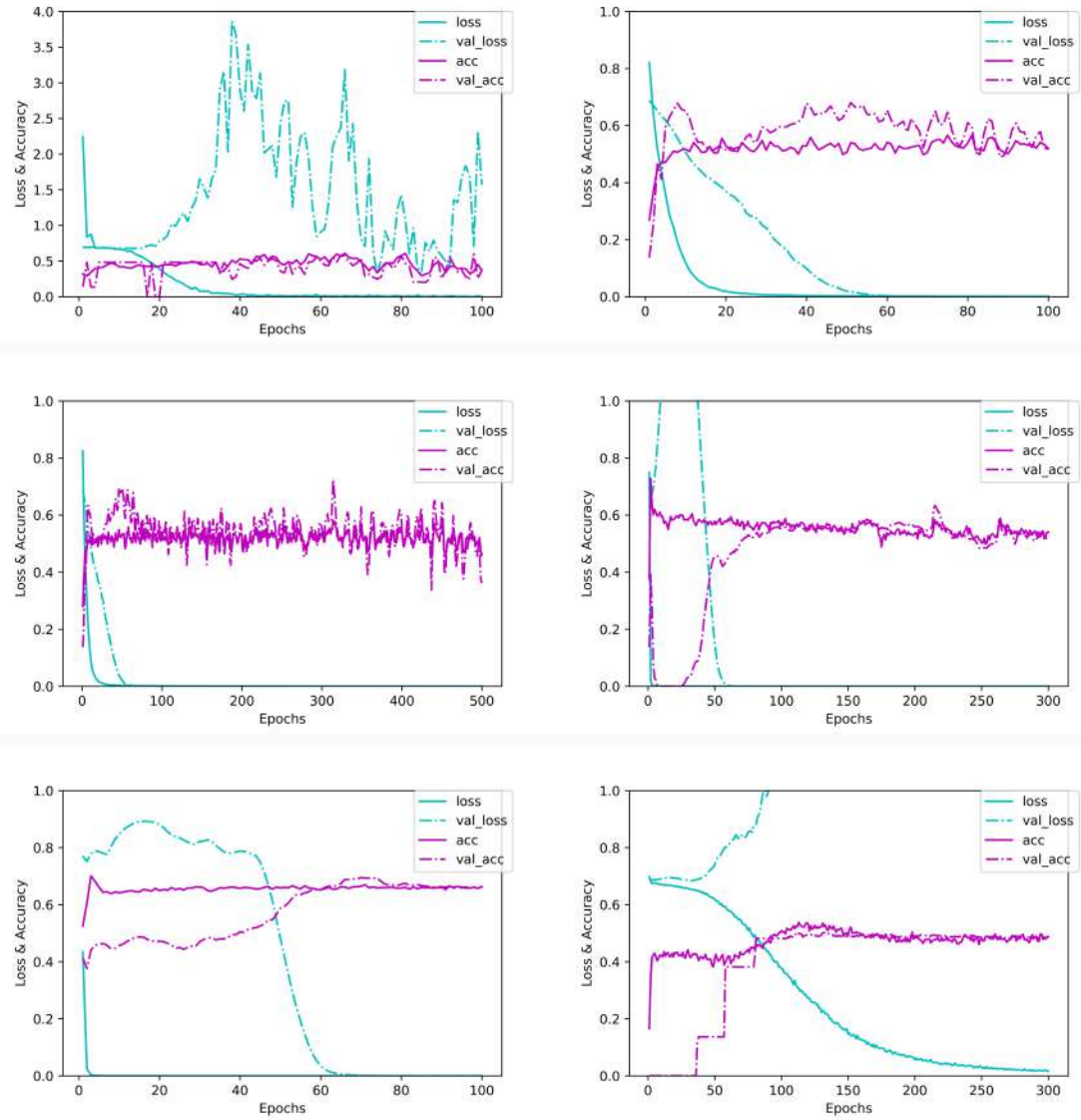


Figure 54: Performances of different cases of 2D CNN models.

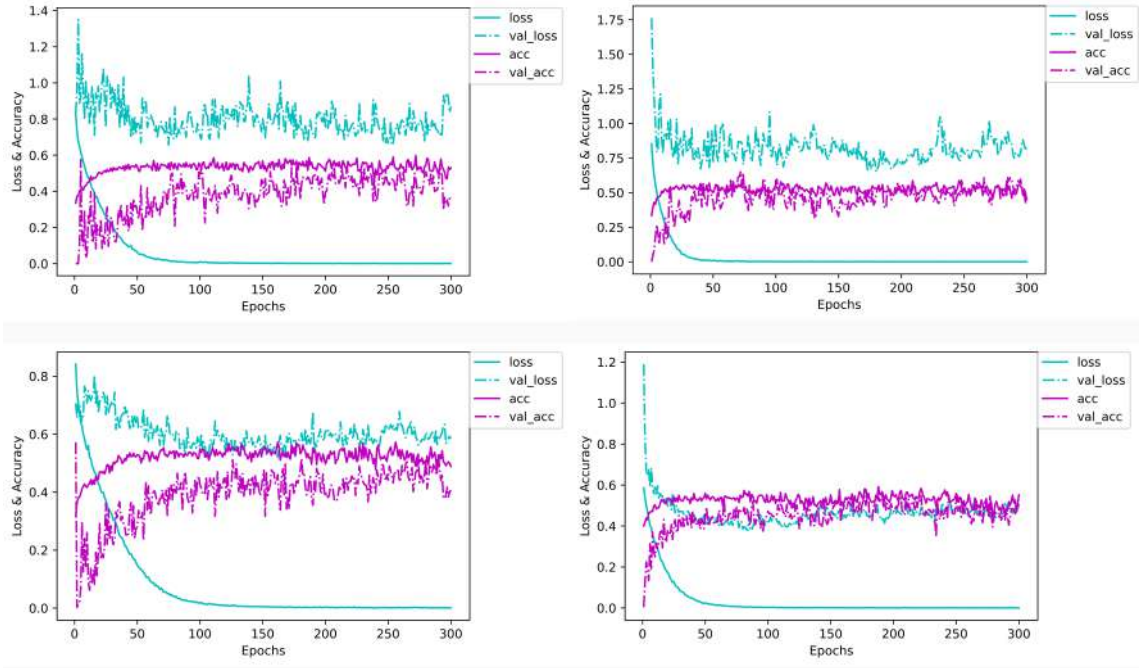


Figure 55: Performances of different cases of 3D CNN models.

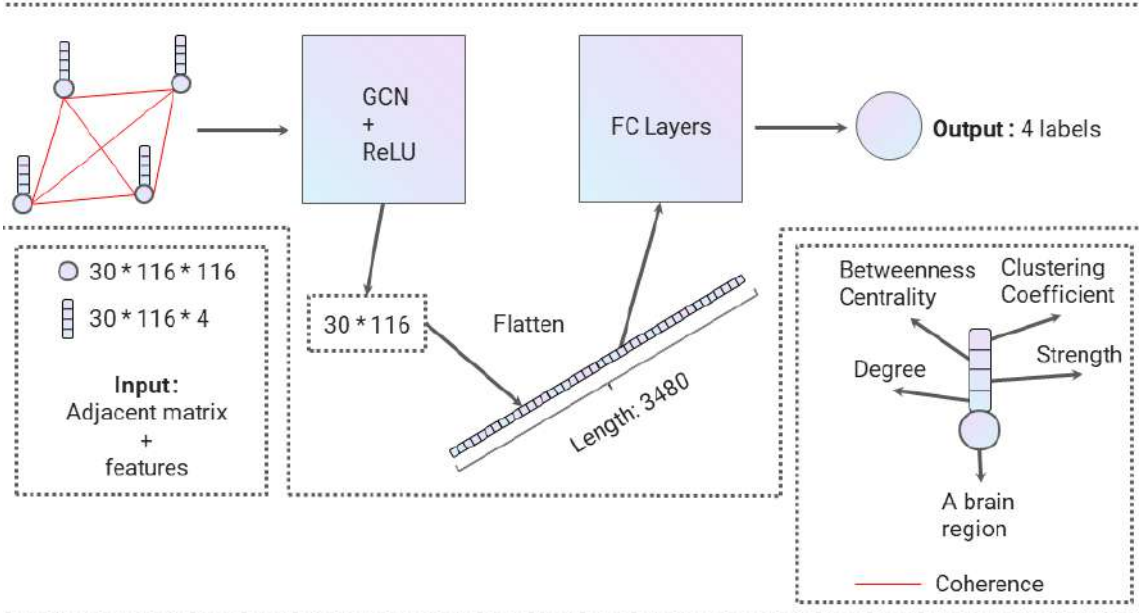


Figure 56: Architecture of the proposed model.

5 Discussion

In this thesis, various deep learning models were proposed to seek solutions to the research questions. For question 1, it is possible to predict that the patients will have improvements in vision before they take the treatment, given their coherence matrix data. The accuracy of the testing data is 95%. For question 2, it is possible to predict that the subject is a healthy control or a stroke patient given their coherence matrix data, and the accuracy of the testing data is 72%. For question 3, it is barely possible to predict a patient's percentages of vision condition given their coherence matrix data. For question 4, it is possible to predict stroke patients' vision distribution given their coherence matrix data, and the accuracy of testing data is 71.9%.

Several possible reasons lead to the poor performance of the models. The first reason is the size of the available data. In total, there are 24 patients and 24 healthy controls. Though each stage of each subject, 50 pieces of data were recorded (3600 instances for patients and 1200 for healthy controls), these data pieces are similar to each other within the same subject (Fig. 57).

The second reason is that the data quality might not be very accurate, which will influence the performance as well. The brain connectome on the source level was converted from EEG data, and EEG data might introduce much noise during the recording process itself. The vision data acquisition was also prone to have errors since it is a manual operation that requires the subject's great attention for a long time.

The third reason is that the connections between the brain connectome and the percentages of vision conditions are not direct and straightforward. The percentages of the vision conditions only reflect the pure numeric values of the vision conditions. For a single value, it could have infinite combinations. For example, the normal vision percentage is 54.4% (240 out of 441 areas are normal). In this case, the normal areas' locations could be distributed anywhere within the 21×21 matrix.

The fourth reason might be the problems of the proposed models. Building and training a model from scratch requires much experience. The original models proposed in this thesis might not reasonably fit for the data set; however, from the

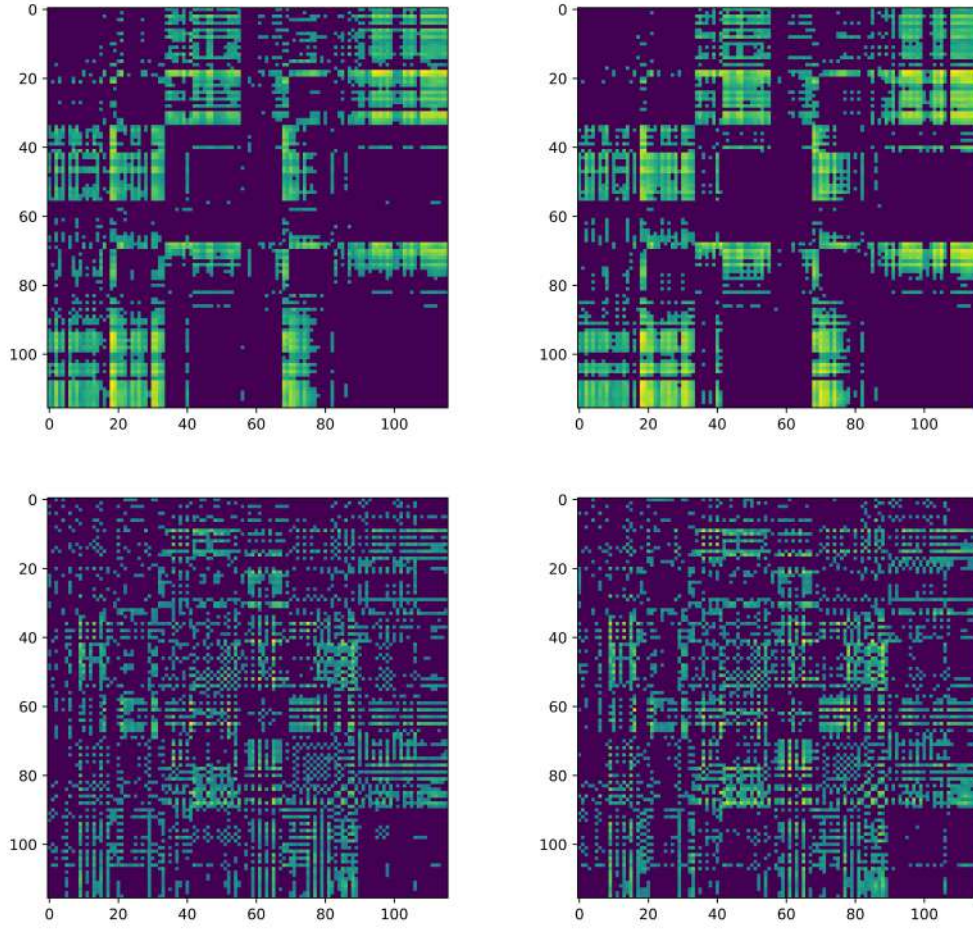


Figure 57: Coherence of the same frequency band in two trials of the same patient in pre-treatment stage (First row: 1Hz; Second row: 10 hz).

feature maps depicted in Figure 58, the model could extract feature patterns.

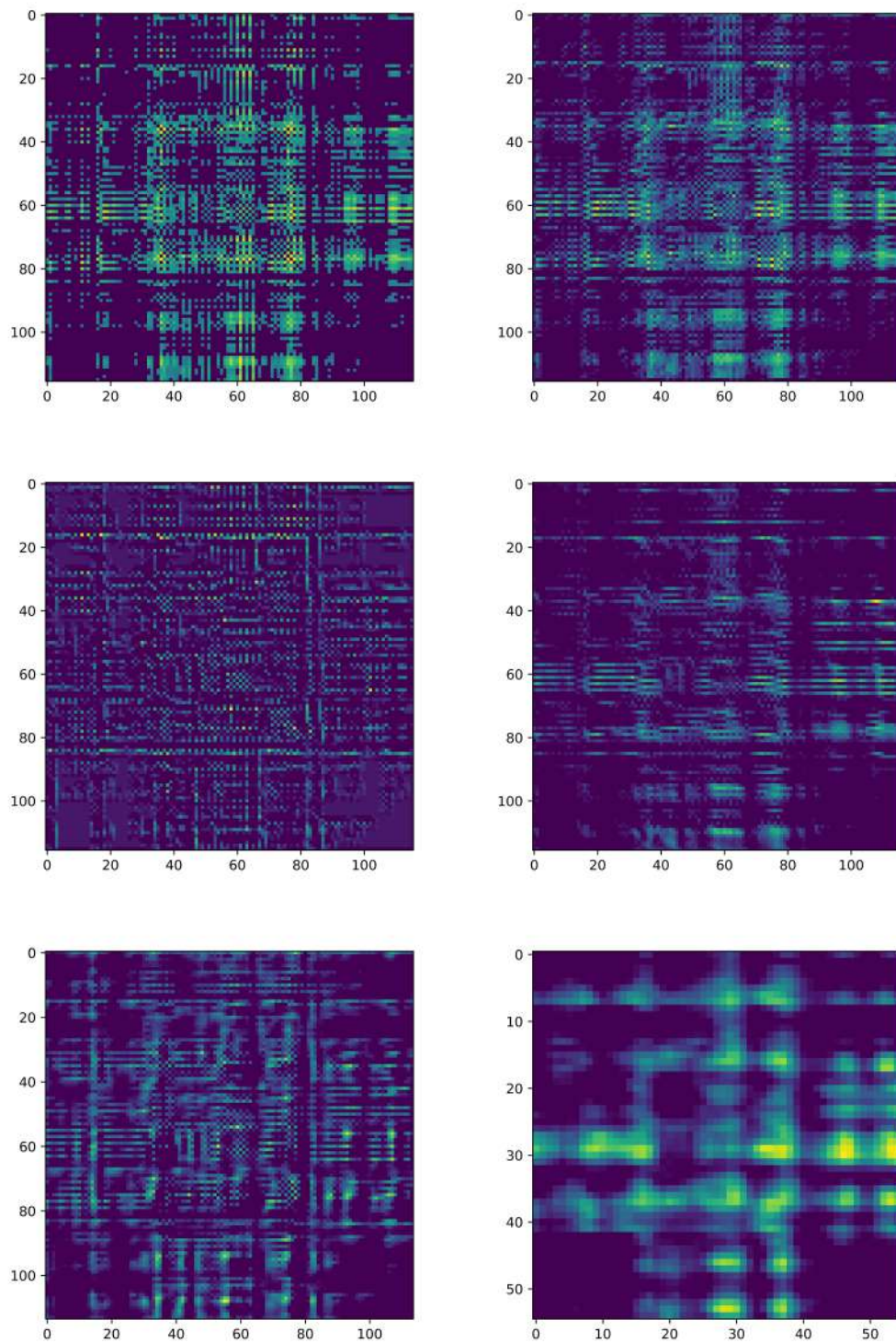


Figure 58: Visualization of feature maps of the proposed 3D CNN model.

6 Conclusion

In this thesis, the results prove that some correlation between the brain connectome and stroke patients' vision exists by using deep learning technology. Multiple deep learning models were proposed and applied to explore specific research questions. Given the currently available dataset, it is possible to discriminate between patients and healthy controls with an accuracy of 72%, and the corresponding model is 3D CNN; The possibility to predict whether a stroke patient can have notorious recovery in vision is 95%, which is also the performance of a 3D CNN model; Predicting vision distribution has an accuracy of 71.9% by fine-tuning predefined 2D CNN classic models; Predicting vision percentage conditions remains a formidable challenge.

Overall, due to the current size of data, this result is most likely suitable for the present given data (24 patients), and it remains uncertain whether it is universal for external data. From the clinical perspective, predicting whether a stroke patient's vision could be well benefited from the rehabilitation has great practical values. If more data could have been used for the experiment, this research would have a promising developing prospect.

References

- [1] *A simplified schema of the human visual pathway*. URL: <https://commons.wikimedia.org/w/index.php?curid=37868501>.
- [2] Xingwei An et al. “Dynamic Functional Connectivity and Graph Convolution Network for Alzheimer’s Disease Classification”. In: (June 2020).
- [3] *Anatomy of the Brain*. URL: <https://mayfieldclinic.com/pe-anatbrain.htm>.
- [4] Salim Arslan et al. “Graph Saliency Maps Through Spectral Convolutional Networks: Application to Sex Classification with Brain Connectivity”. In: *Graphs in Biomedical Image Analysis and Integrating Medical Imaging and Non-Imaging Modalities*. Ed. by Danail Stoyanov et al. Cham: Springer International Publishing, 2018, pp. 3–13.
- [5] Jan Claassen Brandon Foreman. “Annual Update in Intensive Care and Emergency Medicine 2012 Non- invasive Ventilation Outside the ICU”. In: *Updat. Intensive Care Emerg. Med.* January 2011 (2012), pp. 29–37.
- [6] Joan Bruna et al. “Spectral networks and locally connected networks on graphs”. English (US). In: *International Conference on Learning Representations (ICLR2014), CBLS, April 2014*. 2014.
- [7] Hannelore van der Burgh et al. “Deep learning predictions of survival based on MRI in amyotrophic lateral sclerosis”. In: *NeuroImage: Clinical* 13 (Oct. 2016).
- [8] N. R. Carlson. *Structure and Functions of the Nervous System*. Boston, Mass. London: Allyn and Bacon, 2002.
- [9] Hongyoon Choi. “Functional connectivity patterns of autism spectrum disorder identified by deep feature learning”. In: *CoRR* abs/1707.07932 (2017).
- [10] François Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *CoRR* abs/1610.02357 (2016). arXiv: 1610.02357.
- [11] Özgün Çiçek et al. “3D U-Net: Learning Dense Volumetric Segmentation from Sparse Annotation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2016*. Ed. by Sebastien Ourselin et al. Cham: Springer International Publishing, 2016, pp. 424–432.

- [12] FERNANDEZ COLIN. *Partially blind patients can boost their vision by doing a series of simple eye exercises*. Accessed: 2020-04-30.
- [13] Haskell B. Curry. “The Method of Steepest Descent for Non-linear Minimization Problems”. In: *Quart. Appl. Math.* 2 (3): (1986).
- [14] Hao Wen Dong et al. “Musegan: Multi-track sequential generative adversarial networks for symbolic music generation and accompaniment”. In: *32nd AAAI Conf. Artif. Intell. AAAI 2018* (2018), pp. 34–41.
- [15] John C. Duchi, Peter L. Bartlett, and Martin J. Wainwright. “Randomized smoothing for (parallel) stochastic optimization”. In: *Proc. IEEE Conf. Decis. Control* 12 (2012), pp. 5442–5444. ISSN: 01912216.
- [16] EEG. URL: <https://mehtahospital.com/index.php/gallery-item/eeg/>.
- [17] Pedro F Felzenszwalb et al. “Object Detection With Partbase”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 32.9 (2010), pp. 1627–1645.
- [18] Angela Ferreira et al. “Colaboração terapêutica: Estudo comparativo dois casos de insucesso terapêutico – um caso finalizado e de um caso de desistência”. In: *Anal. Psicol.* 33.2 (2015), pp. 165–177. ISSN: 08708231.
- [19] Linton C. Freeman. “Centrality in social networks: Conceptual clarification”. In: *Social Networks* 1.3 (1979), pp. 215–239.
- [20] W. Freeman. “Origin, structure, and role of background EEG activity. Part 2. Analytic phase”. In: *Clinical Neurophysiology* 115 (2004), pp. 2089–2107.
- [21] Walter Freeman. “Origin, structure, and role of background EEG activity. Part 1. Analytic amplitude”. In: *Clinical neurophysiology : official journal of the International Federation of Clinical Neurophysiology* 115 (Oct. 2004), pp. 2077–88.
- [22] Liuhao Ge et al. “3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation From Single Depth Images”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. July 2017.
- [23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016, p. 204.
- [24] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015).

- [25] C. J. Honey et al. “Predicting human resting-state functional connectivity from structural connectivity”. In: *Proc. Natl. Acad. Sci. U. S. A.* 106.6 (2009), pp. 2035–2040. ISSN: 00278424.
- [26] Andrew G. Howard et al. “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications”. In: *CoRR* abs/1704.04861 (2017).
- [27] Gao Huang, Zhuang Liu, and Kilian Q. Weinberger. “Densely Connected Convolutional Networks”. In: *CoRR* abs/1608.06993 (2016).
- [28] Andreas A. Ioannides. “Dynamic functional connectivity”. In: *Curr. Opin. Neurobiol.* 17.2 (2007), pp. 161–170. ISSN: 09594388.
- [29] M. Jiménez Muro et al. “Stroke patients in South Madrid: Function and motor recovery, resource utilization, and family support”. In: *Stroke* 31.6 (2000), pp. 1352–1359. ISSN: 00392499.
- [30] Henrik S. Jørgensen et al. “Outcome and time course of recovery in stroke. Part I: Outcome. The Copenhagen stroke study”. In: *Arch. Phys. Med. Rehabil.* 76.5 (1995), pp. 399–405. ISSN: 00039993.
- [31] Diederik P. Kingma and Jimmy Lei Ba. “Adam: A method for stochastic optimization”. In: *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* (2015), pp. 1–15.
- [32] Thomas N. Kipf and Max Welling. “Semi-Supervised Classification with Graph Convolutional Networks”. In: *Proceedings of the 5th International Conference on Learning Representations (ICLR)*. ICLR ’17. Palais des Congrès Neptune, Toulon, France, 2017.
- [33] Yazhou Kong et al. “Classification of Autism Spectrum Disorder by Combining Brain Connectivity and Deep Neural Network Classifier”. In: *Neurocomputing* 324 (May 2018).
- [34] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. “ImageNet Classification with Deep Convolutional Neural Networks”. In: (2012). Ed. by F. Pereira et al., pp. 1097–1105.
- [35] Sofia Ira Ktena et al. “Distance Metric Learning Using Graph Convolutional Networks: Application to Functional Brain Networks”. In: *Medical Image Computing and Computer Assisted Intervention MICCAI 2017*. Ed. by Maxime Descoteaux et al. Cham: Springer International Publishing, 2017, pp. 469–477. ISBN: 978-3-319-66182-7.

- [36] Sofia Ira Ktena et al. “Metric learning with spectral graph convolutions on brain connectivity networks”. In: *NeuroImage* 169 (2018), pp. 431–442.
- [37] Siddique Latif et al. “Deep Representation Learning in Speech Processing: Challenges, Recent Advances, and Future Trends”. In: (2020), pp. 1–25.
- [38] Y. Lecun et al. “Gradient-based learning applied to document recognition”. In: *Proceedings of the IEEE* 86.11 (1998), pp. 2278–2324.
- [39] Y. Li et al. “Supervised Graph Representation Learning for Modeling the Relationship between Structural and Functional Brain Connectivity”. In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2020, pp. 9065–9069.
- [40] Jialin Liu et al. “Stock Prices Prediction using Deep Learning Models”. In: (2019), pp. 1–15.
- [41] Zhiyuan Liu and Jie Zhou. “Introduction to Graph Neural Networks”. In: *Synthesis Lectures on Artificial Intelligence and Machine Learning* 14 (Mar. 2020), pp. 1–127.
- [42] Xin Ma, Guorong Wu, and Won Hwa Kim. “Multi-resolution Graph Neural Network for Identifying Disease-specific Variations in Brain Connectivity”. In: *ArXiv* abs/1912.01181 (2019).
- [43] Daniel Maturana and Sebastian Scherer. “VoxNet: A 3D Convolutional Neural Network for real-time object recognition”. In: Sept. 2015, pp. 922–928.
- [44] Vinod Nair and Geoffrey E. Hinton. “Rectified Linear Units Improve Restricted Boltzmann Machines”. In: (2010), pp. 807–814.
- [45] Mathias Niepert, Mohamed Ahmed, and Konstantin Kutzkov. “Learning Convolutional Neural Networks for Graphs”. In: *Proceedings of The 33rd International Conference on Machine Learning*. Ed. by Maria Florina Balcan and Kilian Q. Weinberger. Vol. 48. Proceedings of Machine Learning Research. New York, New York, USA: PMLR, 2016, pp. 2014–2023.
- [46] P. L. Nunez and R. Srinivasan. *Electric Fields of the Brain: The Neurophysics of EEG*. New York, NY: Oxford University Press, 2006.
- [47] Hae-Jeong Park and Karl Friston. “Structural and Functional Brain Networks: From Connections to Cognition”. In: *Science* 342.6158 (2013). ISSN: 0036-8075. DOI: 10.1126/science.1238411.

- [48] Chun-Ren Phang et al. “Classification of EEG-Based Brain Connectivity Networks in Schizophrenia Using a Multi-Domain Connectome Convolutional Neural Network”. In: *CoRR* abs/1903.08858 (2019).
- [49] A. Polaski, S.E. Tatro, and J. Luckmann. *Luckmann’s Core Principles and Practice of Medical-surgical Nursing*. Saunders, 1996. ISBN: 9780721659947.
- [50] D. Purves et al. “Neuroscience”. In: Sinauer associates Inc., 2004.
- [51] H. Robbins and S. Monro. “A stochastic approximation method”. In: *Annals of Mathematical Statistics* 22 (1951), pp. 400–407.
- [52] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. “U-Net: Convolutional Networks for Biomedical Image Segmentation”. In: *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. Ed. by Nassir Navab et al. Cham: Springer International Publishing, 2015, pp. 234–241.
- [53] Abhimanyu Roy et al. “Deep learning detecting fraud in credit card transactions”. In: *2018 Syst. Inf. Eng. Des. Symp. SIEDS 2018* (2018), pp. 129–134.
- [54] Mikail Rubinov and Olaf Sporns. “Complex network measures of brain connectivity: Uses and interpretations”. In: *Neuroimage* 52.3 (2010), pp. 1059–1069. ISSN: 10538119.
- [55] Sebastian Ruder. “An overview of gradient descent optimization algorithms”. In: (2016), pp. 1–14.
- [56] David E. Rumelhart, Geoffrey E. Hinton, and Ronald J. Williams. “Learning Representations by Back-Propagating Errors”. In: *Neurocomputing: Foundations of Research*. Cambridge, MA, USA: MIT Press, 1988, pp. 696–699. ISBN: 0262010976.
- [57] Bernhard A. Sabel, Hans Strasburger, and Erich Kasten. “Programs for diagnosis and therapy of visual field deficits in vision rehabilitation”. In: *Spatial Vision* 10.4 (1Jan. 1997), pp. 499–503.
- [58] Saeid Sanei and J.A. Chambers. “Introduction to EEG”. In: *EEG Signal Processing*. John Wiley Sons, Ltd, 2013. Chap. 1, pp. 1–34. ISBN: 9780470511923.
- [59] F. Scarselli et al. “The Graph Neural Network Model”. In: *IEEE Transactions on Neural Networks* 20.1 (2009), pp. 61–80.

- [60] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.* (2015), pp. 1–14.
- [61] S. Siuly, Y. Li, and Y. Zhang. “EEG Signal Analysis and Classification”. In: *Health Information Science*. 2016.
- [62] Shuran Song and Jianxiong Xiao. “Deep Sliding Shapes for Amodal 3D Object Detection in RGB-D Images”. In: *CoRR* abs/1511.02300 (2015).
- [63] Olaf Sporns. “The human connectome: A complex network”. In: *Ann. N. Y. Acad. Sci.* 1224.1 (2011), pp. 109–125. ISSN: 17496632.
- [64] Heung-Il Suk, Chong-Yaw Wee, and Seong-Whan Lee. “State-Space Model with Deep Learning for Functional Dynamics Estimation in Resting-State fMRI”. In: *NeuroImage* 129 (Jan. 2016).
- [65] Kah Kay Sung and Tomaso Poggio. “Example-based learning for view-based human face detection”. In: *IEEE Trans. Pattern Anal. Mach. Intell.* 20.1 (1998), pp. 39–51. ISSN: 01628828.
- [66] “The magnetic lead field theorem in the quasi-static approximation and its use for MEG forward calculation in realistic volume conductors”. In: *Physics in medicine and biology* 48 (Dec. 2003), pp. 3637–52.
- [67] N. Tzourio-Mazoyer et al. “Automated Anatomical Labeling of Activations in SPM Using a Macroscopic Anatomical Parcellation of the MNI MRI Single-Subject Brain”. In: *NeuroImage* 15.1 (2002), pp. 273–289. ISSN: 1053-8119.
- [68] Peter J. Uhlhaas et al. “The role of oscillations and synchrony in cortical networks and their putative relevance for the pathophysiology of schizophrenia”. In: *Schizophr. Bull.* 34.5 (2008), pp. 927–943. ISSN: 05867614.
- [69] Ragav Venkatesan et al. *Convolutional Neural Networks*. Elsevier Inc., 2018, pp. 89–116. ISBN: 9780128157398.
- [70] *Visual problems after stroke*. URL: <https://www.stroke.org.uk/>.
- [71] Akihiko Wada et al. “Differentiating Alzheimer’s Disease from Dementia with Lewy Bodies Using a Deep Learning Technique Based on Structural Brain Connectivity”. In: *Magnetic Resonance in Medical Sciences* 18 (July 2019), pp. 219–224.
- [72] D.J. Watts and S.H. Strogatz. “Collective dynamics of ‘small-world’ networks”. In: *Nature* 393 (1998), pp. 440–442.

- [73] Xin Wen et al. “Detecting the Information of Functional Connectivity Networks in Normal Aging Using Deep Learning From a Big Data Perspective”. In: *Frontiers in Neuroscience* 13 (2020), p. 1435. ISSN: 1662-453X.
- [74] *What is homonymous hemianopsia?* URL: <https://my.clevelandclinic.org/health/diseases/15766-homonymous-hemianopsia->.
- [75] *What’s to know about hemorrhagic stroke?* URL: <https://www.medicalnewstoday.com/articles/317111>.
- [76] Jennifer Wu et al. “Utility of EEG measures of brain function in patients with acute stroke”. In: *J. Neurophysiol.* 115.5 (2016), pp. 2399–2405. ISSN: 15221598. DOI: 10.1152/jn.00978.2015.
- [77] Adwan Yasin and Fadi AbuAlrub. “Enhance RFID Security Against Brute Force Attack Based on Password Strength and Markov Model”. In: *Int. J. Netw. Secur. Its Appl.* 8.5 (2016), pp. 19–38. ISSN: 09752307.
- [78] Matthew D. Zeiler. “ADADELTA: An Adaptive Learning Rate Method”. In: *CoRR* abs/1212.5701 (2012).
- [79] Shuai Zhang et al. “Deep learning based recommender system: A survey and new perspectives”. In: *ACM Comput. Surv.* 52.1 (2019), pp. 1–35. ISSN: 15577341.
- [80] Yanfu Zhang and Heng Huang. “New Graph-Blind Convolutional Network for Brain Connectome Data Analysis”. In: *Information Processing in Medical Imaging*. Ed. by Albert C. S. Chung et al. Cham: Springer International Publishing, 2019, pp. 669–681.