Georgia State University

ScholarWorks @ Georgia State University

Computer Science Dissertations

Department of Computer Science

Summer 8-11-2020

Biomedical Data Classification with Improvised Deep Learning Architectures

Heta Desai

Follow this and additional works at: https://scholarworks.gsu.edu/cs_diss

Recommended Citation

Desai, Heta, "Biomedical Data Classification with Improvised Deep Learning Architectures." Dissertation, Georgia State University, 2020.
doi: https://doi.org/10.57709/18808634

This Dissertation is brought to you for free and open access by the Department of Computer Science at ScholarWorks @ Georgia State University. It has been accepted for inclusion in Computer Science Dissertations by an authorized administrator of ScholarWorks @ Georgia State University. For more information, please contact scholarworks@gsu.edu.

BIOMEDICAL DATA CLASSIFICATION WITH IMPROVISED DEEP LEARNING ARCHITECTURES

by

HETA DESAI

Under the Direction of Rajshekhar Sunderraman, PhD

ABSTRACT

With the rise of very powerful hardware and evolution of deep learning architectures, healthcare data analysis and its applications have been drastically transformed. These transformations mainly aim to aid a healthcare personnel with diagnosis and prognosis of a disease or abnormality at any given point of healthcare routine workflow. For instance, many of the cancer metastases detection depends on pathological tissue procedures and pathologist reviews. The reports of severity classification vary amongst different pathologist, which then leads to different treatment options for a patient. This labor-intensive work can lead to errors or mistreatments resulting in high cost of healthcare. With the help of machine learning and deep learning modules, some of these traditional diagnosis techniques can be improved and aid a doctor in decision making with an unbiased view. Some of such modules can help reduce the cost, shortage of an expertise, and time in identifying the disease.

However, there are many other datapoints that are available with medical images, such as omics data, biomarker calculations, patient demographics and history. All these datapoints can enhance disease classification or prediction of progression with the help of machine learning/deep learning modules. However, it is very difficult to find a comprehensive dataset with all different modalities and features in healthcare setting due to privacy regulations. Hence in this thesis, we explore both medical imaging data with clinical datapoints as well as genomics datasets separately for classification tasks using combinational deep learning architectures. We use deep neural networks with 3D volumetric structural magnetic resonance images of Alzheimer Disease dataset for classification of disease. A separate study is implemented to understand classification based on clinical datapoints achieved by machine learning algorithms. For bioinformatics applications, sequence classification task is a crucial step for many metagenomics applications, however, requires a lot of preprocessing that requires sequence assembly or sequence alignment before making use of raw whole genome sequencing data, hence time consuming especially in bacterial taxonomy classification. There are only a few approaches for sequence classification tasks that mainly involve some convolutions and deep neural network. A novel method is developed using an intrinsic nature of recurrent neural networks for 16s rRNA sequence classification which can be adapted to utilize read sequences directly. For this classification task, the accuracy is improved using optimization techniques with a hybrid neural network.

INDEX WORDS: Deep learning, Medical data analysis, AI in healthcare

BIOMEDICAL DATA CLASSIFICATION WITH IMPROVISED DEEP LEARNING ARCHITECTURES

by

HETA DESAI

A Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

in the College of Arts and Sciences

Georgia State University

2020

Copyright by
Heta Prakash Desai
2020

BIOMEDICAL DATA CLASSIFICATION WITH IMPROVISED DEEP LEARNING ARCHITECTURES

by

HETA DESAI

Committee Chair: Rajshekhar Sunderraman

Committee: Michael Weeks
Yanqing Zhang
Yi Jiang

Electronic Version Approved:

Office of Graduate Studies

College of Arts and Sciences

Georgia State University

August 2020

DEDICATION

To my loving, caring, kind and hard-working parents, who have sacrificed so much in life to provide me with opportunities to get a higher education and comforts of life. They are the epitome of loving and living life selflessly. My journey of becoming a scientist started long before I even started perusing my graduate studies. It started with my parents and older brother instilling me with curiosity towards nature's wonders, perseverance for finding answers to scientific questions, and hard work to achieve those answers. Most importantly, their positive outlook to life has aided me in keeping my optimism high during the toughest times of my education journey.

To my second set of parents, my brother and sister-in-law who have always been my support system and have given me their unconditional love throughout my college education. My brother gave up on his masters to help me with my undergraduate funds. To my nephews, whose innocence makes my world so much brighter and motivates me to continue working in scientific research that would aid in making this world a better place for them and many future generations.

To my husband, who encouraged me and motivated me to work harder than ever before. He patiently listened to my countless ideas and random research talks, no matter how funny they seemed. He helped me think clearly to achieve my goals one at a time and continues to do so. His desire for my success is ten times higher than my own.

Finally, to my daughter, who was in my womb while I finished a lot of this dissertation work. She is constantly reminding me of how beautiful being a nurturer to the wonder of this nature is. The amalgamation of nature and machines that enhance research and lives have always fascinated me, hence my research interests have always been in interdisciplinary studies of physical sciences and engineering.

ACKNOWLEDGEMENTS

This work would not have been possible without many people. First and foremost, my kind-hearted advisor for guiding me since last 6 years; allowing me to work full-time during my first three years of graduate school and answering thousands of silly questions I have about computer science. Without his utmost support, encouragement and guidance, this work would not have been possible. We have had many thoughtful conversations regarding various research fields, and I am thankful for him for trying his best to provide me with resources needed to perform my research.

Secondly, to my committee members Dr. Michael Weeks, Dr. Yanqing Zhang and Dr. Yi Jiang. Dr. Weeks has taught me to be thorough, precise and punctual on my research as well as writing. Taking his class was one of the best ways I improved on my scientific writing. I took a few classes with Dr. Zhang, first was Computational Intelligence, in which I was introduced to some of the most crucial machine learning algorithms. He has been one of the humblest and understanding human being I have come across at Georgia State University. Lastly, Dr. Jiang whom I met at a talk she organized on “Chimpanzee and its behaviors” in Math department, which eventually became my masters’ project. I look up to her as a successful woman in science for her research and interest in multidisciplinary topics, which motivates me to be in multidisciplinary research.

Thirdly, I also want to thank my husband’s parents for understanding and helping at home while I drift away writing or doing my research. My extended family and friends who have always supported me when in doubt. I want to thank all my uncles, aunts, cousins and extended family who have been part of this tough journey. My friends, “the three musketeers” for always being

there for me, like always. I could not have possibly reached here without their support and long phone conversations.

Also, I want to thank all the administrative staff at the CS department, Ms. Rice, Ms. Pittman, Ms. Martin, Mr. Bryan and most importantly Ms. Dudley. Their presence truly makes the department more cheerful. Ms. Dudley has always maintained a smile and supported me through the tough times as well as always encouraged me to do better. She has been a friend, and a sister whenever needed to be. Thank you for everything you do for every student who knock on your doors every day. Lastly, I would like to thank all my colleagues at the department of Computer Science (CS), and Ms. Anuja P. Parameshwaran, who have become one of the closest friends within a short period of time. She has provided me with some of the core image processing techniques' background during our brainstorming sessions.

I would like to acknowledge one of the quotes by a great scientist, engineer, artist and many other things, who have motivated me to explore many interdisciplinary areas. *"*Principles for the Development of a Complete Mind: Study the science of art. Study the art of science. Develop your senses- especially learn how to see. Realize that everything connects to everything else.*" -Leonardo da Vinci*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

Georgia State University (GSU)

Artificial Intelligence (AI)

Artificial Neural Network (ANN)

Backpropagation through Time (BPTT)

Computer-aided detection (CADe)

Computer-aided diagnosis (CADx)

Convolutional Neural Network (CNN)

Deep Learning (DL)

Deoxyribonucleic Acid (DNA)

Graphic Processing Units (GPUs)

Rectified Linear Units (ReLU)

ImageNet Large Scale Visual Recognition Challenge (ISLVRC)

Long Short-Term Memory (LSTM)

Machine Learning (ML)

National Institute of Health (NIH)

Medical Image Computing and Computer Assisted Intervention (MICCAI)

Medical Imaging with Deep Learning (MIDL)

Magnetic Resonance Image (MRI)

Structural Magnetic Resonance Image (sMRI)

Functional Magnetics Resonance Image (fMRI)

Positron Emission Tomography (PET)

Computerized Tomography (CT)

Open Access Series of Imaging Studies (OASIS-3)

Recurrent Neural Network (RNNs)

Ribonucleic Acid (RNA)

Ribosomal RNA (rRNA)

Visual Geometry Group (VGG)

1 INTRODUCTION

1.1 Overview

With the influx of large datasets and computational resources powerful enough to perform complex calculations, techniques of data analysis have also changed with broadened areas of applications [1]. With stronger graphic processing units (GPUs), scientists and researchers are now being able to collectively analyze data at a much larger scale than ever before [2], giving rise to the field of deep learning [3]. Today, deep learning has drastically changed the way images [4], videos, and textual data [5] have been studied collectively, quantitatively and qualitatively. More recently, the advancements in this field has also influenced applications in healthcare data analysis. In past before deep learning, it used to be very difficult to interpret, classify medical images or medical reports due to either limitations in publicly available dataset ; however, in last few years hundreds of papers have been published with advancement studies involving deep learning [6]; be automatic breast cancer detection [7], skin cancer lesion detection through a phone application [8] or diabetic retinopathy vessel segmentation [9]. The broad idea of artificial intelligence is that a computer can mimic a human behavior to aid in automizing a task. Figure 1.1 below illustrates an overall representation of artificial intelligence and its sub research fields of machine learning and deep learning.

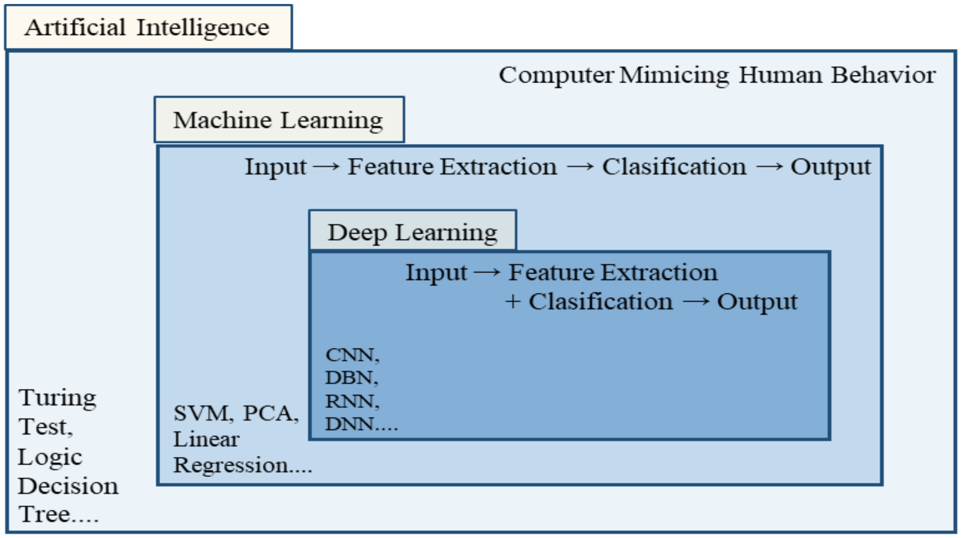

Artificial Intelligence
Computer Mimicing Human Behavior
Machine Learning
Input → Feature Extraction → Clasification → Output
Deep Learning
Input → Feature Extraction + Clasification → Output
CNN, DBN, RNN, DNN....
SVM, PCA, Linear Regression....
Turing Test, Logic Decision Tree....

Figure 1.1 Overview of Artificial Intelligence and its popular classifiers

1.2 Outline of contributions.

First focus of the study is to classify 16s rRNA bacterial gene based on its sequences. For this task below are the model architectures that are explored:

1) recurrent neural network (RNNs) are closely examined, especially Long Short-Term Memory (LSTM) dependent architectures.
2) Further, 1-dimentional convolutional neural networks (1D CNNs).
3) Combinational hybrid models consisting of both convolutional neural networks
4) Ensemble models involving hybrid models

This is one of the first study that investigates recurrent neural networks to classify 16S rRNA in their taxonomy. These pilot studies will help with future work in AI in healthcare implementing various deep learning architectures and by applying ensemble models of machine learning with deep learning.

Second focus of this study was to classify Alzheimer’s disease based on T1w MRI slices. For this study, an in-depth investigation is performed as below.

1) Eight convolutional neural networks for MRI medical images and two machine learning neural networks for clinical data are created.
2) Binary vs. Multiclass for all 10 models classification explored
3) Regression analysis to find highly correlated clinical data
4) A concatenated model architecture to utilize all slices from three planes, axial, sagittal and coronal.

1.3 Medical image analysis

There are mainly two type of applications of artificial intelligence in medical image analysis: 1) Computer-aided detection (CADe), 2) Computer-aided diagnosis (CADx). CADe is to identify an abnormality in region of interest and to improve detection rate of such regions with lowering the false negative rate. Typically, in CADe, the region of interest is detected with image processing techniques, features are represented as statistical information and features are fit. CADx are known for its discrimination of malign or benign lesions. There are both unsupervised and supervised learning methodologies used in networks with majority being supervised learning. With the rise of personalized medicine and electronic health records, National Institute of Health (NIH) is also bringing some standards with such technologies in order to be deployed and used in real time, including standards in radiology reports of such imaging [10, 11].

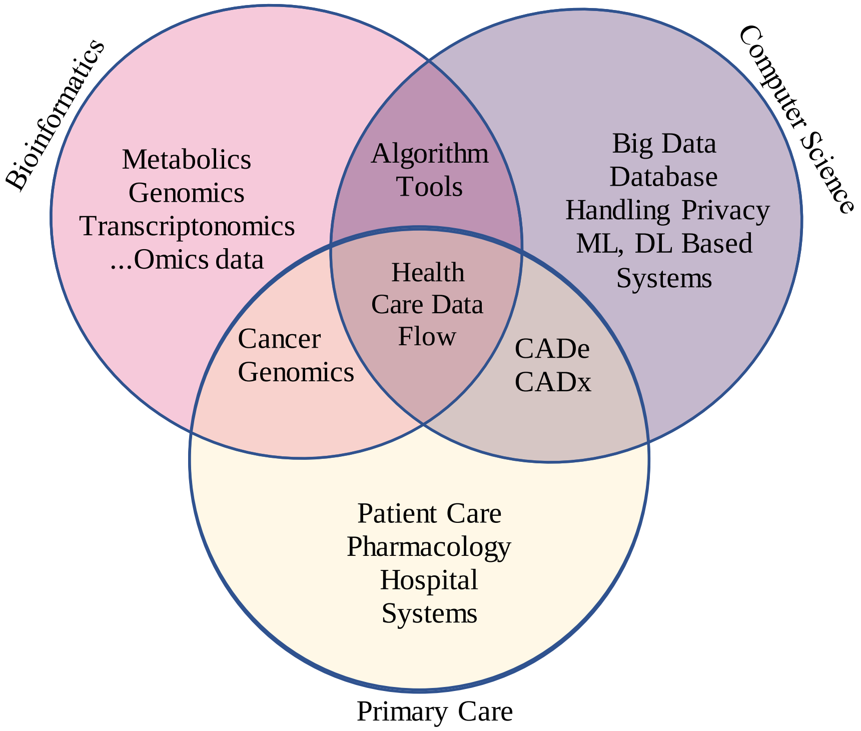


Figure 1.2 Overview of Health care Data type and Data Flow

Regardless of many successful studies of deep learning in medical image analysis, the biggest challenge right now is analyzing different sources of data collectively. One of the main ideas of this study is being able to collectively analyze and utilize all patient data that is available for AI in healthcare to aid in an end-to-end pipeline in future. Healthcare data today has three main sources, 1) any omics data such as genomics, proteomics, metagenomics, or transcriptomics etc. related to patient and/or diseases, 2) medical images from various modalities for disease detection/progression/classification/segmentation, and 3) textual data such as doctor's notes, chief complains, symptoms, pathology/radiology reports, hospital care details, etc. These three datatypes belong to three separate areas of research, 1) bioinformatics, 2) computer vision and 3) natural language processing respectively. Each research area is vast, but have to be studied in order to understand overall complexity of data analysis. Most of the models are either modality dependent or organ/disease dependent for the image processing tasks to work effectively. In order to help through an automatic detection, this study attempts to suggest some deep learning systems can come together and serve as a template.

1.4 Dissertation outline

This dissertation is organized in total of seven chapters including Chapter 1. In this dissertation we solved two classification problems from two different areas of 1) bioinformatics and 2) medical image processing by developing several neural network architectures and overcoming some of the challenges pertaining to tasks. We provide a foundation for understanding neural network architectures and problems at hand in Chapters 2 and 3. In Chapter 4, 5 and 6, we provide the contributions made in this dissertation. Lastly, in Chapter 7, we provide a brief reflection on this research work and future directions.

Chapter 2, we offer an overview of deep architectures, CNNs in chronological order of their more recent development utilized in medical image classification, and RNNs with explanation of basic architectures utilized in later chapters of genomics sequence classification. **Chapter 3**, we provide background information necessary to lay foundation for understanding medical image classification and genomics sequence classification tasks. In this chapter we also provide related works of both aforementioned classifications along with background information on application of deep learning in different areas of medical images and genomics.

In **Chapter 4**, we tackle the problem of deoxyribonucleic acid (DNA) sequence classification. That is, provided a sequence of bacterial 16S rDNA sequences, predict the corresponding taxonomy at family, genus and species levels. We approach this problem as a textual classification problem with fixed length input lengths of input sequences. In this initial chapter, we used basic recurrent neural networks and convolutional neural networks to for classification bacterial sequences into taxonomical representation. The content of this chapter is based on initial model architectures from Desai et.al [12] and comparison study of initial architectures with convolutional neural networks from Desai et. al [13]. **Chapter 5** is an extension of chapter 4, where we compare ensemble and hybrid models of recurrent neural networks and convolutional neural networks to see if an ensemble models without any further modifications in data can achieve state-of-the art performances. This chapter is based on methodology section of Desai et. al [14]. **Chapter 6** is multi-data study performed on brain MRI dataset to classify Alzheimer's disease for imbalanced dataset which is normally seen in real world applications especially in healthcare and biology. In this chapter we create a model architecture to separately observe clinical data information based on heavily utilized machine learning algorithms, as well as deep learning architectures that utilizes all three planes of a structural magnetic resonance imaging (sMRI). We also anticipate a major improvement on model architecture that combines both approaches to bring both model in same multimodal feature space. Finally, **Chapter 7** concludes this dissertation with challenges and some of the ways to avoid bias in such deep architecture dependent technologies in healthcare. We also lay a path forward for some of the improvements to solve existing challenges.

2 DEEP LEARNING ARCHITECTURES

2.1 Convolutional Neural Networks (CNNs)

From the moment we wake up in the morning and open our eyes, our brain starts processing visual information around us. Unconsciously, all the information is compartmentalized and identified by just looking at a scenery. One looks outside a window and can identify tree, sunshine, clouds, birds, buildings etc. We also identify minute things, make correlational examination and make assumptive decisions from what we see. All these things are going on in our brains every day, intuitively we link objects with what they are linguistically called and make an instant label. These are all very complex processing, but we grow up learning things by labeling and making connections of objects to language. For instance, a child is shown a picture of a bird and is taught the spelling as well as verbal pronunciation of the word. These object identification task has been taught to us by repetitive example correlational research from an early age. Such model is an inspiration to machine learning paradigm or training and testing data to find correlation and patterns. Eye is the first contact point of this visual system; however, processing happens in area of the posterior part of brain called as primary visual cortex as seen in figure 2.1 below [15].

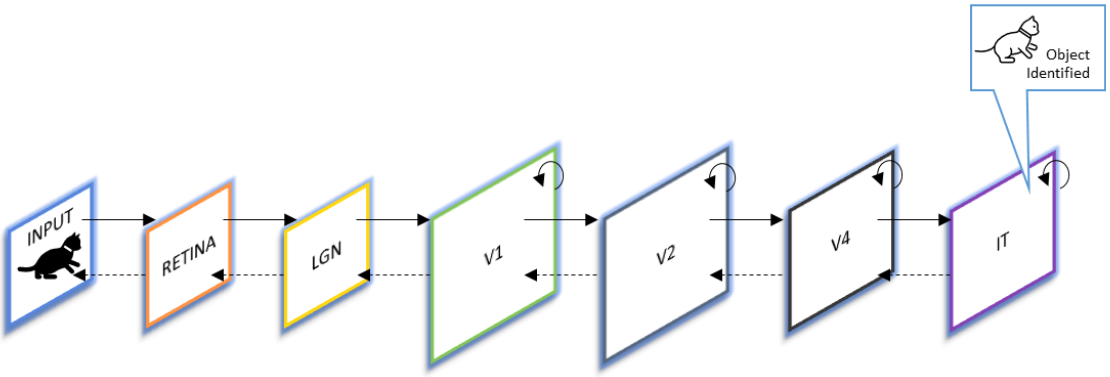


Figure 2.1 Primary visual cortex and its functionality in image analysis.

The inspiration behind CNN can be better understood by observing animal's visual cortex. In 1960s, two researchers from Harvard Medical School, Dr. Hubel and Dr. Wiesel, first published a model on mammalian visual system (studies done of cats) showing how cells in primary visual cortex perceives surrounding world visually. Eyes see a small sub-regional scene, called receptive (visual) fields, normally divided in left visual field and right visual field. Visual cortex of brain receives these visual fields as an input. These small inputs are normally put together in series of slides to cover the entire receptive field. In visual cortex, there are two type of biological cells that play a very crucial role in the way our brain processes an image. These two cells are known as simple cells (S cells), and complex cells (C cells) [16, 17]. The simple cells are activated during edge detection, pattern tasks at a fixed angle view while complex cells are activated during larger receptive field without any restrictions on the view angle or position. This cascading model of S cells and C cells works together in pattern recognition. The receptive field in retina receives a stimulus that activates neurons in that field, which in return sends a somatic signal to downstream neuron bodies. Such information is passed in hierarchy and is stored in the order it is received, in terms sequential. The part of the brain that is involved in memories called neocortex, stored such information hierarchically.

2.1.1 A history of Convolutional Neural Networks

Recently, CNNs architecture have been making headlines with its various applications in robotics, disease/cancer classification, self-driving cars etc. However, the history of CNNs rather started in late 1980s Fukushima first proposed a neural network called neocognitron, a network with "an ability of unsupervised learning, learning without a teacher" [18]. This model was indeed inspired by previously mentioned "S cell/C cell" model proposed by Dr. Hubel and Dr. Wiesel, where an architecture was made with Simple cells, where parameters were modified with a layer of complex cells, where pooling is performed [18, 19]. Much later in 1998, a group of researchers, Le Cun, Bottou, Bengio and Haffner, published a "Gradient-based learning" applied for document recognition. This study revealed the first ever Convolutional Neural Network, which they called LeNet-5.

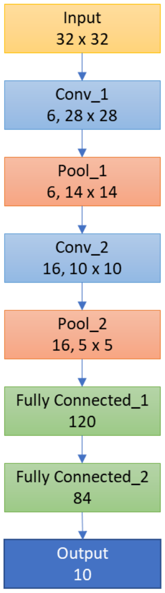


Figure 2.2 The very first LeNet-5 architecture with 3 convolutional and 2 subsampling layers, with last one fully connected with classifier and output layer figure simplified to portray LeNet-5 from [20].

They proposed many versions of CNNs in their paper, with LeNet-5 being the best architecture, which was used to identify digits from hand-written numbers [20]. LeNet-5 as shown in the figure 2.2 above, have three main layers, convolutions, subsampling (pooling) and non-linearity (with tanh or sigmoid). This basic components of CNNs is used in image classification tasks in deep learning till date. Each layer of CNNs is explained in later sections in detail.

After a decade long gap, with the rise of graphical processing units (GPUs); automatic image classification, object detection, and speech recognition tasks came about in lime-light as well as rise of large image datasets. In 2009, a very well-known computer science professor Fei-Fei Li opened an ImageNet Large Scale Visual Recognition Challenge (ISLVRC), consisting of labeled image dataset available to researchers, professors and students. In 2010, Ciresan and Schmidhuber came up with the first implementations of neural networks using NVDIA GTX 280 GPU with up to nine-layer neural network. Shortly after that, in 2012, Alex Krizhevsky, Sutskever and Hinton, proposed a much deeper architecture than LeNet and called it AlexNet in the ImageNet competition [21]. The architecture won the competition with 16.4 % error rate on classification task using AlexNet [22].

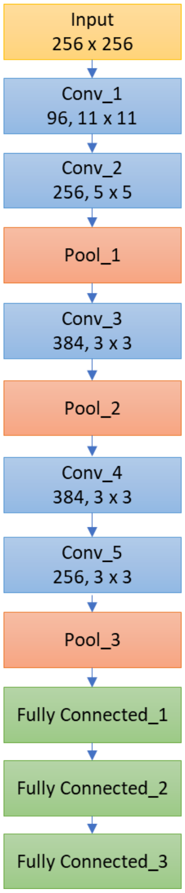


Figure 2.3 Main architecture of AlexNet, with 5 convolutional layers and 3 fully connected layers with two separate Graphic Processing Units (GPUs), this is a simple illustration of AlexNet.

One of the major contributions in architecture was of using rectified linear units (ReLU) as non-linearity functions after the convolutional layer instead of tanh or sigmoid, used data augmentation techniques on input data, dropout layer to reduce overfitting on training data, and overlapping max-pooling. In 2013, the winner of ImageNet competition was ZFNet, found by Zeiler and Fergus. The architecture was very similar to AlexNet with some modifications involving filter size changes, introduced cross-entropy loss for error function and training using batch stochastic gradient descent. The model's final error rate was 11.2% almost 5% less than AlexNet, by mainly fine-tuning the existing model. The researchers also developed a visualization technique called Deconvolutional Networks (DeConvNet) for different feature activation at each layer. In 2014, two architectures became popular, one GoogLeNet (the winner of the competition) and the other VGG [23]. GoogLeNet was a 22-layer CNN achieved 6.7 % of error rate. This was one of the first CNN architectures that created a very different architecture and deviated from stacking of the core of three-layer CNN architecture, Convolution-pooling-nonlinearity together. The main contribution of this paper was to introduce an inception module, which consists of different filter combinations of small convolutions 1 x 1, 3 x 3 and 5 x 5. These convolutions are concatenated at the end each inception module to pass to the next layer hierarchically. Figure 2.4 below shows one of the inception modules from GoogLeNet.

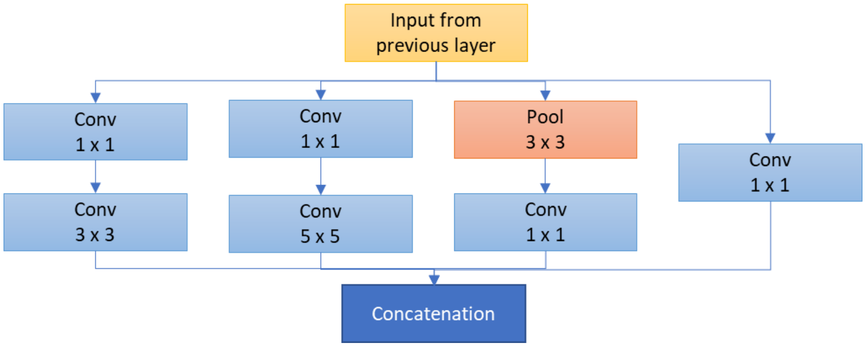


Figure 2.4 This figure shows one of the inception modules introduced in GoogLeNet architecture, which is shown stacked approximately nine time in overall architecture [11].

In this architecture, several inception layers are stacked together to create the final architecture. Another popular architecture, a runner-up from 2014 ILSVRC competition was VGGNET, developed by the Visual Geometry Group (VGG), University of Oxford. This network contributed in identifying network's depth as one of the potential hyperparameter to achieve better recognition. The networked used two convolutional layers followed by an activation layer of ReLU; however, used three fully connected layers at the end in all proposed architectures [22]. Six variants of VGGNET were proposed the same year, VGG A through E; although, three of these six are now widely known as VGG-E (A), VGG-16(C) and VGG-19(E) [24]. All of them had similar architecture, except varying numbers of two convolutional layer + ReLU units; with VGG-E having total of eight convolutional layers, VGG-16, thirteen and VGG-19, sixteen such layers [24].

In 2015, another popular CNN architecture – Residual Network (ResNet) won the competition ILSVRC, proposed by Kaiming He from Microsoft Research Asia. It brought error rate down to 3.57%, much lower than the human error rate of 5 % [22, 25]. ResNet was proposed with many varying numbers of convolutional layers: 34, 50, 101, and 152.

This architecture deviated from the rest vastly as it introduced "a residual block". This block goes through the series of convolutional layer – ReLU – convolutional layer, gives you some output, let's say F(x). Now this output is added to the input x, so finally you get F(x) + x, whereas, in popular CNN, the output would be only F(x).

After the release of such powerful architectures, a lot more variants came out most recently. CNNs architectures performs the best, when the input data has a very structural component, such as an image or an audio with repeating patterns [26].

2.2 Recurrent Neural Networks (RNNs)

2.2.1 Simple Vanilla RNN

Recurrent neural networks (RNNs) are known to be temporally deep networks i.e. the RNNs are usually unrolled or unfolded in time. There are certain formulas that govern the computations of the RNN and they are: 1) input x_t which is associated at time step *t*. 2) hidden state h_t, or in other words the memory which is calculated by taking the previous hidden state (at the previous time step (*t-1*)) and present input into consideration. i.e. $h_t = f\ (Ux_t + Wh_{t-1})$, where *U, W* are shared parameters associated with the different layers of RNN and *f* is a nonlinearity functions which is generally a tanh function or a ReLU function. 3) output o_t at time step *t*.

The vanilla RNN cell unit is a simple unit where the previous hidden state h_{t-1} and current input x_t is passed through the *tanh* non-linearity function to update the current hidden state, i.e. equation 2.1 as below.

$$h_t\ = tanh\, W\binom{x_t}{h_{t-1}} \qquad \textit{Equation 2.1}$$

The drawback of vanilla RNNs is that it is difficult to train these networks. The updating of parameters for Vanilla recurrent neural networks happen the same way as that of artificial neural networks i.e. through the back-propagation algorithm. The catch with respect to the calculation of gradients at every time step is with respect to that fact that vanilla RNNs have shared parameters between all-time steps of a layer, as opposed to independent parameters of an ANN. Thus, to calculate a gradient in a current time step there is a need to backpropagate to previous time steps until the present one making the vanilla RNN have difficulties in learning long term dependencies i.e. dependencies between far away time steps. This backpropagation algorithm is called Backpropagation through Time (BPTT) and this can lead to the vanishing/exploding gradient problem. [27] Better variants of the RNNs like the LSTMs and the GRUs have been designed to solve this problem, thus making these variants popular for tasks like sequence classification and prediction.

2.2.2 LSTM and Bidirectional LSTM

More variants of RNNs were developed to address the shortcomings of a simple vanilla RNN. LSTMs are a popular variant of RNNs, each unit of the LSTM is associated with memory typically called as a cell. In a LSTM cell unit, the memory is regulated with the help of three gates namely input gate (i_t), output gate (o_t, hidden state h_t) and forget gate (f_t), which helps determine which information needs to be added to the current cell state (C_t) and which information can be forgotten to update the cell state. The equations 2.2 to equation 2.7 below represents the data flow from the current cell state, previous cell state and the next state. [28]

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \qquad \textit{Equation 2.2}$$
$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i) \qquad \textit{Equation 2.3}$$
$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C) \qquad \textit{Equation 2.4}$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t \qquad \textit{Equation 2.5}$$
$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o) \qquad \textit{Equation 2.6}$$
$$h_t = o_t * tanh\ (C_t) \qquad \textit{Equation 2.7}$$

The advantages of the LSTMs over vanilla RNNs are that they were specifically designed to overcome the vanishing gradient problems and are deemed efficient in capturing long-term dependencies. Due to the vanishing gradient problem in Vanilla RNN models, LSTM were introduced since its ability to update its own cell-state. In LSTM unit, the horizontal line on top acts as a “conveyer belt – with some minor linear interactions,” making backpropagation task much simpler [28].

Bidirectional LSTMS capture the idea that the output of recurrent unit at a time step not only depends on its past instances (past elements of the sequence) but also on the future instances. The idea of such a network is developed by stacking 2 layers of LSTMs over each other thus making the output dependent on the computation of hidden states from both the LSTM layers as opposed to one as in the unidirectional LSTM network.

3 REVIEW DEEP LEARNING APPLICATIONS IN MEDICAL IMAGING AND GENOMICS

3.1 Relevant Deep Learning applications for biomedical data in Literature

With impressive breakthroughs in computer vision and natural language processing, and due to its power of being able to identify intrinsic features within the dataset, deep learning has drawn attention of almost every interdisciplinary researcher even out of computer science domain. This chapter focuses on background of two main applications of deep learning in, 1) sequence classification in genomics and 2) disease diagnosis/classification with medical imaging data. Within genomics, this dissertation focuses on bacterial taxonomy classification, and within medical imaging, this dissertation focuses on Alzheimer's disease classification using structural MRI. For each section, dataset, methods, architectures and results are discussed in detail.

3.1.1 Bacterial taxonomy classification

There have been many machine learning approaches in solving genomics problems, more recently, deep learning approaches are also appearing with tools to solve many popular problems related to human genomes, especially in functional and regulatory genomics. In functional genomics, prediction of DNA sequence specificity to RNA binding protein cis regulatory cites, gene expression, methylation status and chromatin immunoprecipitation (ChIP) sequencing are some of the main applications of deep learning [29]. Whereas in regulatory genomics, due to DNA being a double helix with its strand and reverse complement of the same strand can give different sequences, which in deep learning can cause misinterpretation of data [29].

Bacterial classification is a crucial bioinformatics application in public health. More and more efforts are being made to correctly identify bacteria at genus and species level in an environment sample. This classification task is mostly carried out with gene that codes for 16S rRNA – known to be a conservative region amongst domain of bacteria. According to definition of taxonomy, it is a systematic way of classifying living organisms that fall under a specific kingdom. For kingdom of bacteria, it is harder to classify an organism as you move to least inclusive class such as order, family, genus and species, the genetic material shared by species within the same genus taxa has a very high percentage of sequence identity. Figure 3.1 below shows the bacterial taxonomy order at level depicts an example of this taxonomy.

Level	**Example**
Domain	Bacteria
Phylum	Protobacteria
Class	γ – Protobacteria
Order	Enterobacterials
Family	Enterobacteriaceae
Genus	*Escherichia*
Species	*E. coli*

Figure 3.1 Bacterial Taxonomy Classification, the focus of this study is on last three taxa, Family, Genus and Species.

There are many applications of genus level identification of bacteria in a sample, especially in metagenomics, infectious disease, and material identification – where a company called Phylagen is trying to map the origin of a product through its manufacturing profile for transparency is global supply chain and to fight counterfeit materials. As metagenomics sample sequencing involves sequencing entire contents of a sample, it can be very difficult to identify a very low abundance of DNA origin, especially at a genus and species level at which it is more informative and discriminative.

There are many taxonomy classifiers developed with machine learning algorithms such as support vector machine (SVM), random forests, preprocessed nearest neighbor (PLSNN – based on partial least squares) and naïve Bayes [30, 31, 32, 33]. One of the other notable work came from Fioravanti et.al. who developed phylogenetic convolutional neural network (Ph-CNN) for metagenomics dataset [33]. Even though this application is used for metagenomics classification, its main goal is to classify human gut microbiome of Inflammatory Bowel Disease (IBD) patient's vs healthy controls [33]. This section, however, focuses on classifiers that are based on deep learning models, most of which came after the year 2015. Fiannaca et.al. developed a convolutional neural network (CNN) as well as deep belief network (DBN), whereas Busia et.al. developed a deep neural network (DNN) for this bacterial classification task [34, 35]. In forthcoming sections, Fiannaca et. al.'s study is referred as study 1 and Busia et. al.'s study is referred as study 2.

3.3.1.1 Datasets

Datasets used for bacterial taxonomy classification mainly includes of four main sources 1) complete curated genes of 16S rRNA, 2) whole genome or shot-gun sequencing reads from next generation sequencing (NGS) platforms 3) metagenomics samples and 4) amplicon sequencing reads from primer enhanced libraries. Metagenomics samples are mainly used when environmental samples or community/human gut microbiome samples are being analyzed [33]. Most of metagenomic sample reads available in public dataset, don't have any corresponding taxonomy classification associated with it, hence normally, a mock metagenomics read datasets are created using popular tools like REAGO [35, 34, 36, 37]. Fiannaca et. al uses this approach to first create a mock dataset and then uses open source tools to isolate reads belonging to 16S gene with 99%

accuracy, mainly for comparisons and testing purposes. The authors of this paper, also create another dataset, called shot-gun (SG) and amplicon (AMP) by downloading over 57788 16S gene sequences from RDP database (dated on 16th September, 2016) by filters using such as length greater than 1200 bps, quality – good , and source – isolate [34, 38]. However, to tackle imbalance dataset, authors limit this dataset by selecting sequences only belonging to Protobacteria Phylum, hence having 1000 sequences from 100 genera and 10 species per genus. The final number of sequences belonged to 3 classes, 20 orders, 39 families, and 100 genera. After selecting these sequences, Grinder was used to generate both shotgun and amplicon sequences with mutations rate with replacement at $3 \times 10^{-3} + 3$, $3 \times 10^{-8} \times i^{4}$, where i is the position of nucleotide [39]. For amplicon sequences, primers for only V3-V4 regions that are approximately 469 bp long combined, were considered, which resulted in 28000 short reads, and losing 86 sequences that didn’t match with the primers used described in [40]. The reasons for choosing V3 was that it has the most amount of single nucleotide polymorphisms (SNPs) and V4 region has been regarded as most discriminatory against V5-V6 for phylogenetic variance [40, 41, 42, 43].

The dataset was obtained from National Center for Biotechnology Information (NCBI) using reference sequence database (RefSeq) to generate the mock reads dataset (reference NCBI). More specifically 18,902 sequences of bacterial 16S rRNA were obtained from the RefSeq BioProjects 33175 and 33317 (downloaded on 27th November, 2017). The average length was approximately 1455 base pair long, and the sequences varied from 302 to 3600 base pairs [35]. Busia et.al used a multi-length read approach where mock read lengths of 25, 50, 100, 150 and 200 were generated from reference sequence, using a sliding window fashion. In order to get the corresponding taxonomy class information from superkingdom to species level, NCBI Taxonomy Browser was used [44]. The final dataset involved represented 38 phyla, 91 classes, 202 orders, 479 families, 2768 genera(genus) and 13,838 species. Despite of having a lot of number of sequences, for species level there is not a lot of representation for each individual species.

3.3.1.2 Different Methods applied

To prepare short-reads for an input to deep learning architectures such as CNN and RNN, normally, short-reads are converted into one-bit encoding for each of the four bases of A, T, G and C. This creates an array of four bit and replaces 1 to represent each of the above four letters, for example, A is represented as [1,0,0,0], and this is considered one-hot encoded raw vector. Fiannaca et. al. and Busia, et.al both studies represent the four bases in this similar fashion of four-bit array representation. Fiannaca et. all uses k-mer representation to extract features from short-reads that are used in sequence classification tasks [34]. K-mer patterns, occurrences and combinations is heavily used in bioinformatics. There are a few drawbacks of k-mer representations, 1) the positional origin of k-mer in the sequence is not maintained or known and 2) depending on the k in k-mers, meaning the the length of a k-mer, the vector space representation of k-mers suffers from very high dimensionality that can grow exponentially. Fiannaca et. al uses k-mers with size of 3 to 7 [34]. Similar to study 1, study 2 also implemented one-hot encoding for all four bases, A, C, T, G as a four-dimensional vector along with International Union of Pure and Applied Chemistry (IUPAC) ambiguity codes. This dataset is further split into NCBI-0/1/2 for model performance. NCBI-0 contained 90% of the species in each genus by random sampling and by getting second sampling of 90% reads for each selected species in the first sampling. Rest of the 10% reads went to NCBI-1/2. The sequence length for this dataset was set to 100 base pairs. To evaluate the resulting DNN classifier, authors in study 2 created 16S sequences of synthetic community samples from [45, 46, 40]. For these mock communities, the read length selected was 250 base pairs and included 49 bacterial strains and 10 strains from archaea.

3.3.1.3 Main Architectures Used

There are only a few papers that utilizes a true deep learning architecture for bacterial taxonomy classification using 16S rRNA gene or reads belonging to the gene. This review mainly discusses architectures used in aforementioned study 1 and study 2, first uses CNN and DBN as a classifier [34] and second uses DNN (with some convolutions) as a classifier [35]. First published study 1 has a pipeline that starts with classified 16S short-reads, which are then converted into k-mer one-hot encoded vector representation. For learning process, a deep learning architecture both CNN and DBN are deployed and final trained model for each class, order, family and genus are obtained [34]. There is no species level classification utilized since for taxonomy classification genus level classification is highly regarded as a final taxon. In DBN, at least a few Restricted Boltzmann Machine (RBM) are utilized as layers. RBM usually can represent an input in a lower dimensional space to be utilized in following layers [34]. It is an unsupervised learning at first, and then fine-tuning happens using a multilayer perceptron (MLP) and finally a logistic regression layer is utilized as a supervised classifier. Fiannaca et.al uses a derived CNN from LeNet-5 for their architecture design, containing of two convolutional layers, each followed by a max-pooling and a ReLU layers. At the end, two fully-connected layers are utilized with an unspecified classification layer [34].

The authors in study 2, utilizes a DNN structure consisting of depth-wise separable convolutions first introduced by Sifre and Mallat in [47]. The main difference in such convolutions is that they can separate ‘task of learning spatial features’ from ‘integrating information across channels’ [48]. Separable convolutions involve of first depthwise convolutions *i* of some weight W_d and some input X, which then is an input of a pointwise convolution *i* of some weight W_p. Main model architecture involved three layers of such depthwise separable convolutions, followed by two to three layers of combination of fully connected layer – leaky rectified-linear (LReLU) function – a dropout regularization and a pooling layer. Finally, a softmax classification layer is utilized to compute final probability distribution of over 13,000 species. The network was implemented using TensorFlow library [49]. Batch size was set to 500 reads as an input x compromising of corresponding species class as a label y with ADAM optimizer. The authors in this study 2 also introduced random base-flipping noise in input sequence at different rates between 0% to 16%.

3.3.1.4 Result Comparisons

First study claims to have achieved a 91.3% accuracy at genus lever compared to RDP classifier that achieved a 83.8% accuracy on the same dataset [34]. The authors performed two separate experiments to first comprehend the ideal k-mer size and parameters of networks, and second to find the classifier performance against the RDP classifier. In the first experiment, they observed k-mer of length 7 performed the best with 99% accuracy for class and up to 80% accuracy for genus level in general. However, in comparison to two datasets, the best performing combination was of AMP dataset with CNN classifier, which obtained 91.33% accuracy for about 100 genus taxa compared to SG dataset with CNN classifier obtained 85.50% accuracy. DBN performed well with AMP dataset, achieving about 91.37% accuracy, whereas; with SG dataset it only achieved 81.27% accuracy. In conclusion, the authors conclude that k-mers of the hypervariable regions of V3-V4 used in AMP dataset serve as a more discriminatory feature than k-mers from shotgun sequencing resulting in higher accuracy with CNN classifier.

The second study showed much promising results despite having a very low representation of initial number of sequences belonging to each individual 13,838 species since the authors used 18.907 initial 16S rRNA genes. The results were promising for all taxon especially from superkingdom to phylum, achieving greater than 95.7% accuracy expect for species taxa where accuracy was 99.9%, 90.0% and 84.4% respectively. However, as real length goes from 200 to 25, for more lower level taxon, such as order, family, genus and species, the accuracies decrease sharply.

3.2 Brief Introduction to Medical Imaging Technology

Over the last few years there have been an exponential growth in number of published papers of machine learning applications in medical image processing. Applications of deep learning especially in medical imaging have been on the rise since 2015. Open medical image data challenges for disease diagnosis, tumor classifications, brain segmentations etc. at popular conferences such as International conference on Medical Image Computing and Computer Assisted Intervention (MICCAI) and on Kaggle online platform have only boosted the popularity of such data analysis amongst the interdisciplinary researchers. Deep learning applications in medical imaging has influenced the research community to such extent that a new international conference named "Medical Imaging with Deep Learning" (MIDL) was initiated in 2018 to accommodate the pace for this research area [50].

3.3 Brief Introduction of Deep Learning applications in Medical Imaging

As indicated in [6], trend in deep learning papers in medical imaging technology is on the rise with most popular model being a CNN due to its ability of being able to learn features from images. However, lately RNN are also being utilized in time-series analysis in medical imaging especially in longitudinal studies that involve of prediction in disease progression. As far as modality of imaging is concerned, the most popular modalities have been MRI and microscopy, in areas of pathology and brain, while segmentation and detection of objects in imaging have topped in area of interest amongst deep learning. Despite of this drastic shift seen in this area, after

2017, there has been an exponential growth, where the number of papers in this field have reached in thousands from hundreds. Some of the core deep learning application areas are discussed further.

3.3.1 Localization/Detection

Localization and detection in deep learning is defined as a task of find an object in an image and drawing a bounding box around the entire object as shown in figure 3.2. One of the most crucial application of such task is in self-driving cars where surrounding objects need to be identified and tagged for its position compared to the focal point. In medical imaging, such task is useful for identifying a tumor in brain in space and time or identifying an abnormal cell growth in pathology microscopic images for intervention and planning [6]. In medical imaging, this task normally requires processing of 3D volumetric data. Identifying objects or lesions in images have been one of the cumbersome, tedious yet important part for clinicians.

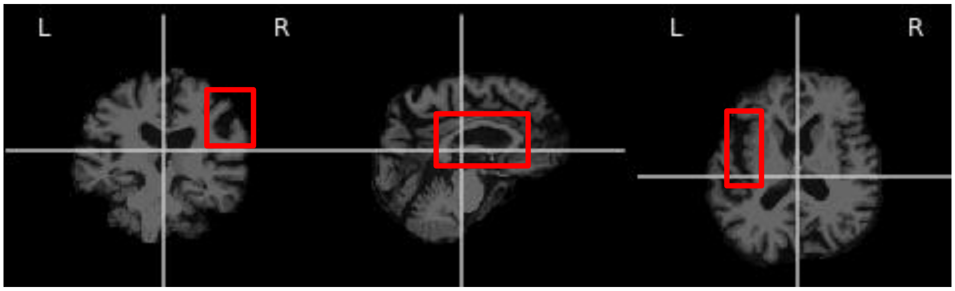


Figure 3.2 An example of localization task for brain lesions in Axial, hippocampus detection in sagittal and lesion in coronal sections

Generally, CNN is used to extract features from every pixel with some postprocessing that can identify the possible objects embedded in pixel representation. The bounding box around object is one of the crucial parts of the algorithm in computer vision, which separates it from segmentation task and classification task.

3.3.2 Classification

This was one of the first areas of application of deep learning in medical imaging since CNN's direct application and popularity for image classification. Earlier in the years, finding a large, balanced and public medical image dataset was an extremely difficult task, hence, most of the applications for clinical detection systems were based on traditional machine learning techniques. After transfer learning algorithms, researchers were able to utilize pre-trained network weights which are trained on millions of generic images, and transfer the learning of features on medical image data classification task despite of having a small dataset. Many studies showed promising results through this, and Litjen's et.al. described such studies in their review [6].

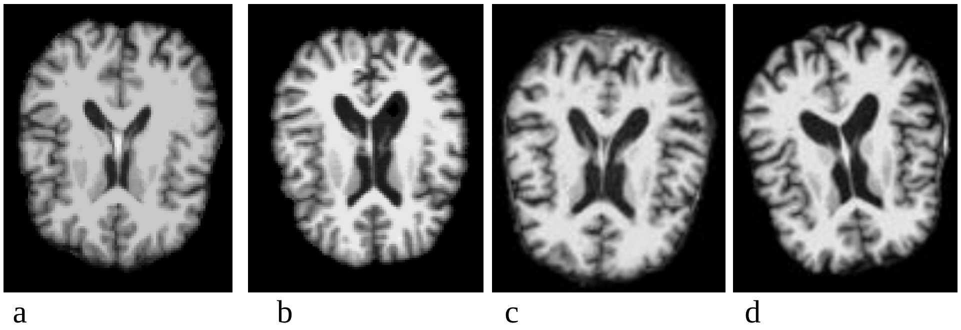
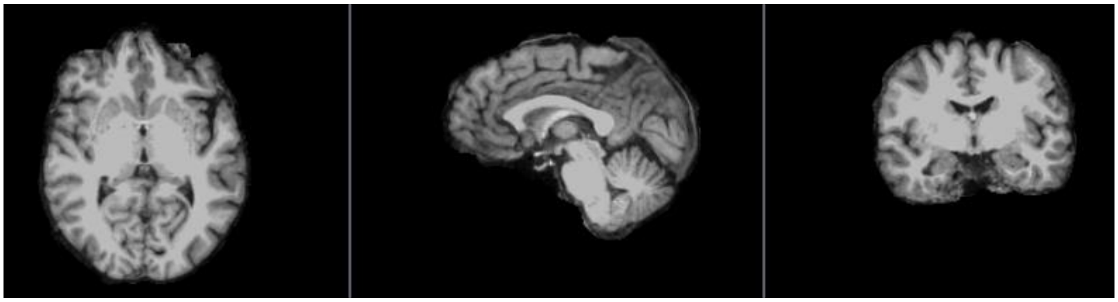


Figure 3.3 Classification is achieved by either detecting disease vs normal or by classifying the disease at different stages based on an input image normally a T1-MRI or FDG-PET scan. Above image is from OASIS-3 dataset example of four classes a) Cognitive normal, b) very mild dementia c) mild cognitive dementia and d) Alzheimer's disease.

There are two types of classification tasks that are popular, first is more superficial or level 1 classification which can identify if an image of a normal patient's or a disease positive patient. At this level, there is no interest in identifying the disease of the stage of the disease, it is simply asking a question if the image is negative or positive. The deeper classification or level 2 classification as shown in above figure 3.3, where the progression of disease becomes a class or known as label y in neural network prediction outcome. Both areas of classification now have seen a tremendous growth in research interest. In the earlier days, almost all of the popular CNN networks discussed in chapter 2 were deployed for medical image classification, however, currently most deployed CNN versions have been a 3D CNN, an ensemble or U-net. One such study by Islam and Zhang et.al. achieved a very high accuracy of 93% for normal vs positive Alzheimer's patient using MRI and FDG- PET scan data from OASIS-2 dataset. The level 1 classification task has been a simpler task to achieve high accuracy; however, level 2 task that required matching is more difficult as disease progression standards varies by physicians to physicians that provide with ground truth values for training of the algorithms.

3.3.3 Segmentation

Segmentation have been one of the most popular application areas in medical imaging. It is used in organ or cell structure segmentation that can serve as clinical features related to area shape in finding abnormalities. Another application of lesion segmentation combines the application of segmentation and detection. More recently, RNNs have become more popular for completing this task, for example, Xie et.al. used a spatial RNN for segmentation in histopathology images, while Stollenga et. al. used 3D LSTM-RNN with convolutional layers. Another popular deep learning architecture for this task has been fCNNs and 2D U-nets in combination with gated recurrent units (GRUs) for 3D segmentation [6].



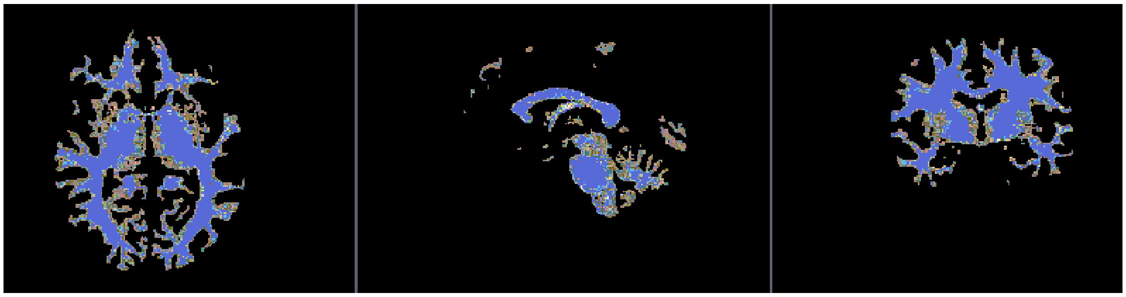


Figure 3.4 Example of OASIS-3 dataset with ground truth of white matter segmentation. Bottom image shows the corresponding of above image for segmented white matter.

Figure 3.4 above illustrates an example of segmentation of white matter in brain axial, sagittal and coronal planes. This segmentation ground truth can differ from physicians to physicians and from different software. Therefore, it is very expensive to minimize this bias by having more than a few medical annotation tasks by physicians that can serve as a good variance for training on deep learning architecture. Hence, the need of more unsupervised learning has been on the rise.

3.3.4 Registration

Lastly, image registration task is spatial alignment of medical images where coordinate transforms is calculated from one medical image to another [6]. Image registration has shown to be beneficial as a preprocessing step in achieving good accuracies of image classification and image segmentation tasks when multi-modal medical image data has been used as an input. One of the ways deep learning techniques are deployed in image registration is to find similarities in two images which then can be optimized further in an iterative fashion. Another way is to predict transformation features using deep regression networks [6]. Nevertheless, in any of the approaches, two way stacked auto-encoders, U-nets and regular CNNs have been the most popular architectures in dealing with image registration tasks.

In [51] one of the more recent tool BIRNet tools that utilizes dual supervised fCNN to solve the image registration in brain MRI images for any of the axial, sagittal, and coronal planes. Also, the authors Fan et.al. uses gap filling, hierarchical loss and multiple sources uses to refine the training processes.

4 16S RIBOSOMAL GENE CLASSIFICATION USING RECURRENT NEURAL NETWORK AND CONVOLUTIONAL NEURAL NETWORK MODELS

Bacterial 16S ribosomal gene is used to classify bacteria because it consists of both highly conservative region as well as a hypervariable region in its sequence [52, 53]. This hypervariable region serves as a discriminative factor to differentiate bacteria at taxonomic levels. In past, many efforts have been made to correctly identify a bacterial species from environmental samples or human gut microbiome samples, yet this identification and subsequent classification task is challenging. For such bacterial taxonomic classification, several studies in the past have been performed based on k-mer frequency matching, assembly-based clustering, supervised/unsupervised machine learning models and a very few studies with deep learning architectures [31, 34, 35]. In this chapter, we study and propose six different deep learning architectures involving Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) to classify bacteria at a family, genus and species taxonomic level using approximately 12,900 16S ribosomal DNA sequences. The best classification accuracies achieved are 92%, 86% and 70% at family, genus and species taxonomic level respectively by variants of RNN.

4.1 Introduction

With the rise of refined, cheap and effective sequencing technologies, it is possible to perform targeted sequencing for identification of bacteria obtained from different environmental samples [52]. Due to this phenomenon, a lot more sequencing projects/dataset have been submitted in National Center for Biotechnology Information (NCBI) database and have been made publicly available. Metagenomics studies have lately drawn a lot of attention since it doesn’t require cell cultures in order to characterize bacterial species [53]. Much more standardize approach is yet to be set for processing such large dataset without creating bias in these analyses. Currently bacterial taxonomic classification tasks depend on read-based sequence matching using tools like mothur and Quantitative Insights Into Microbial Ecology (QIIME) [54, 55]. These tools perform matching of sequences using algorithms involving k-mer frequency, Basic Local Alignment Search Tool (BLAST) and suffix trees. With rise of machine learning algorithms in past decade, various studies emerged involving naïve Bayes approaches such as RDP classifier, hierarchical clustering, random forests and support vector machines (SVM) [31, 34, 35, 38, 56, 57]. To an extent all these studies rely on sequence similarity matching or k-mer frequency counting [38, 56, 31, 57]. Such k-mer based approaches can be limiting as they are independent of a motif position information along with initial sequencing errors/biases which potentially creates noisy input data leading to an inaccurate analysis.

With accessible computational resources and large public data repositories, deep learning techniques are in-demand for classification tasks involving medical image analysis, cancer genomics as well as correcting sequencing errors [58, 59]. There are various kind of neural networks such as the Convolutional Neural Networks (CNNs), the Recurrent Neural Networks (RNNs) and simple Deep Neural Networks (DNNs). While CNNs are generally considered to be a class of feed-forward artificial neural networks in which connections between nodes do not form a cycle and are trained to recognize patterns across the entire spatial domain. Whereas in RNNs, connections between nodes form a directed graph along a temporal sequence allowing this class of neural networks to exhibit temporal dynamic behavior which is highly beneficial while processing sequences, for instance, time-series data. Thus, RNNs are trained to recognize patterns over the time domain. CNNs are normally fixed input size architectures that have found most of their applications to be in the computer vision domain. However, recently scientific community has shown promising results for natural language processing (NLP) related classification tasks using CNNs. Some of the applications of such NLP related classification tasks involves sentiment analysis, spam detection or topic categorization as CNNs can identify patterns (in form of n-grams) from regular expressions in any text data regardless of their position [60, 61, 62, 63, 64, 65, 66, 67].

4.2 Dataset

The dataset was downloaded from Genomic-based 16S ribosomal RNA Database (GRD), which is a manually curated 16S ribosomal DNA sequences [68]. The dataset has sequences that varies in lengths from 65 to 2900, which means some of the sequences are not complete genes and some of the sequences have more basepairs than 16S rDNA sequence, which normally is ~1500 bp long. The dataset has two separate files, one sequence fasta file, and another metadata file containing tab-delaminated fasta header tag matched to bacterial taxonomy of which corresponding sequences belong to. This study focuses on bacterial classification at Family, Genus and Species levels only since normally Phylum, Class and Order level classification achieves > 99% accuracy in most of the models due to sequences belonging to lesser categories [34, 35]. This dataset had sequences belonging to 272, 840 and 2456 categories for family, genus and species level respectively. For input files, each sequence was first separated with respective family, genus and species category comma separated files. Each file was then further processed to cut each sequence in 100 bp non-overlapping length to compare it with variable full-length sequences (results not included in this study). To ensure each subsequence length is 100, if the last cut fragment is greater than 30 bp, it was then padded with Ns else the fragment was discarded. Ambiguous characters other than ATCG were treated as N for model characters. The sequences were randomly divided in 80% training and 20% testing(validation) dataset. This random split may cause an imbalance of classes in train vs test data; however, for our initial comparison, our approach was to proceed with minimizing the data-preprocessing tasks in deep learning.

4.3 Deep Learning Approaches

Fiannaca *et al*. [34] developed CNN based models for bacterial classification task. The CNN in its basic architectural form consists of convolutions and pooling operations. These operations generally lead to loss of sequence information related to data; for example, text data loses information about local order of words in a sentence. RNNs are popularly known as neural networks with memory, and are designed to keep such orderly information intact, allowing the model to also learn the context (semantics). Thus, RNNs are a natural choice for sequence modeling applications which have a time component, making them a powerful and preferred architecture for NLP applications. This work compares the performances of various neural networks for task at hand involving 1) three variants of RNNs, 2) two variants of CNNs and 3) a combinational model which makes use of the convolution and the recurrent layers. In this study, all models are directly applied towards classification of DNA sequences without any prior feature engineering. Three different variations of RNN models namely, vanilla RNN (model 1), Long Short-Term Memory (LSTM) (model 2), Bidirectional LSTM (BiLSTM) (model 3) as illustrated in figure 4.1; two different variations of CNN namely, simple CNN (model 4), multi-filter (MF) CNN (model 5) along with a combinational model (model 6) were used to classify DNA sequence for achieving a higher classification accuracy. In this study, each category – family, genus and species – was evaluated with all three aforementioned RNN and CNN variants trained on GRD dataset [68].

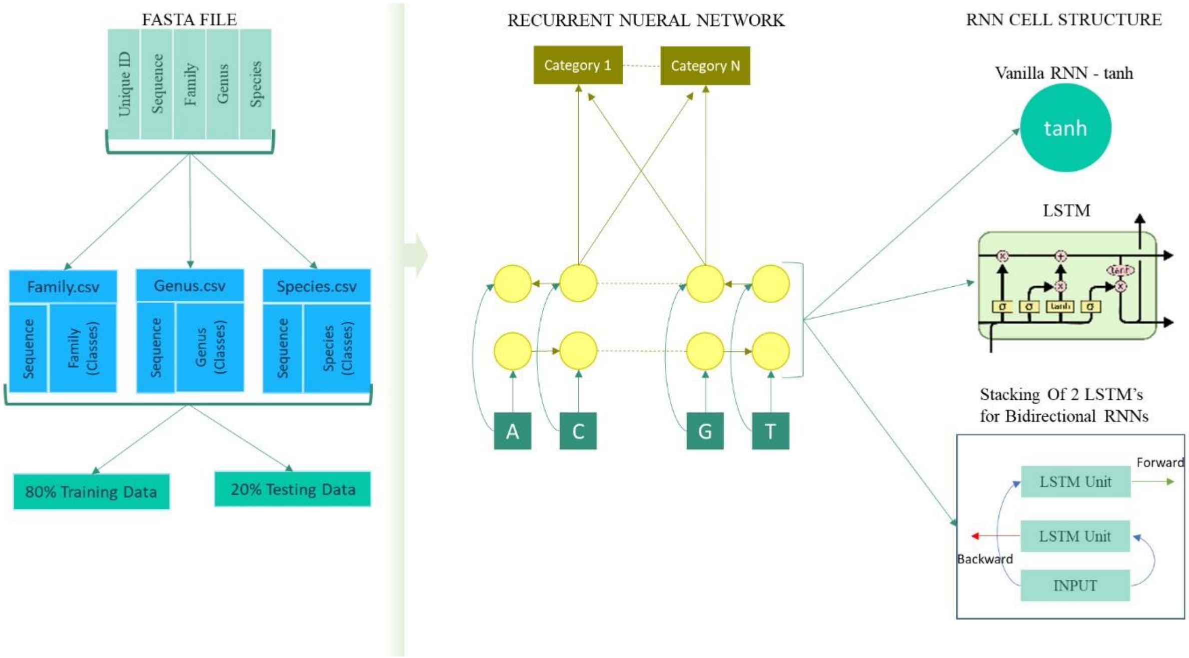


Figure 4.1: Overall architecture of data flow and RNN cell structures used in Model 1 (Vanilla RNN), Model 2 (LSTM) [28], and Model 3 (BiLSTM).

DNA sequence classification can be modeled as a predictive modeling problem. For such problems, models generally take some sequence of inputs and predict a suitable category or class that better defines the sequence as an output. RNNs are an inherent choice of architecture for solving this problem primarily for two important reasons: 1) because of the presence of internal memory and 2) their intrinsic ability to deal with variable length sequences as input. These neural networks are expected to take in variable length sequences as input and learn long-term dependencies between the various symbols of the input sequence, making them very suitable architectures for sequence classification as well as sequence prediction tasks. On the other hand, CNNs are known to be faster i.e. their computational times is shorter than RNNs. CNNs are a better architecture to extract features (feature detection tasks) compared to RNNs, which are intuitively suited for classification tasks involving a long range of semantic dependency [69].

Therefore, both RNN models along with CNNs and combinational architecture are explored used in this study.

4.3.1 Simple vanilla RNN

Recurrent neural networks (RNNs) are known to be temporally deep networks i.e. the RNNs are usually unrolled or unfolded in time. There are certain formulas that govern the computations of the RNN. First, input x_t which is associated at time step t. Second, hidden state h_t, or in other words the memory which is calculated by taking the previous hidden state (at the previous time step (t-1)) and present input into consideration. i.e. $h_t = f(Ux_t + Wh_{t-1})$, where U,W are shared parameters associated with the different layers of RNN and f is a nonlinearity functions which is generally a tanh function or a rely function. Finally, output o_t at time step t.

The vanilla RNN cell unit is a simple unit where the previous hidden state h_{t-1} and current input x_t is passed through the *tanh* non-linearity function to update the current hidden state, i.e.

$$h_t \ = \tanh W\binom{x_t}{h_{t-1}}$$

The drawback of vanilla RNNs is that it is difficult to train these networks. The updating of parameters for Vanilla recurrent neural networks happen the same way as that of artificial neural networks i.e. through the back-propagation algorithm. The catch with respect to the calculation of gradients at every time step is with respect to that fact that vanilla RNNs have shared parameters between all-time steps of a layer, as opposed to independent parameters of an ANN. Thus, to calculate a gradient in a current time step there is a need to backpropagate to previous time steps until the present one making the vanilla RNN have difficulties in learning long term dependencies i.e. dependencies between far away time steps. This backpropagation algorithm is called Backpropagation through Time (BPTT) and this can lead to the vanishing/exploding gradient problem [27]. Better variants of the RNNs like the LSTMs and the GRUs have been designed to solve this problem, thus making these variants popular for tasks like sequence classification and prediction. Figure 4.2 below is a simplified model architecture to show recurrent neural networks in normal flow.

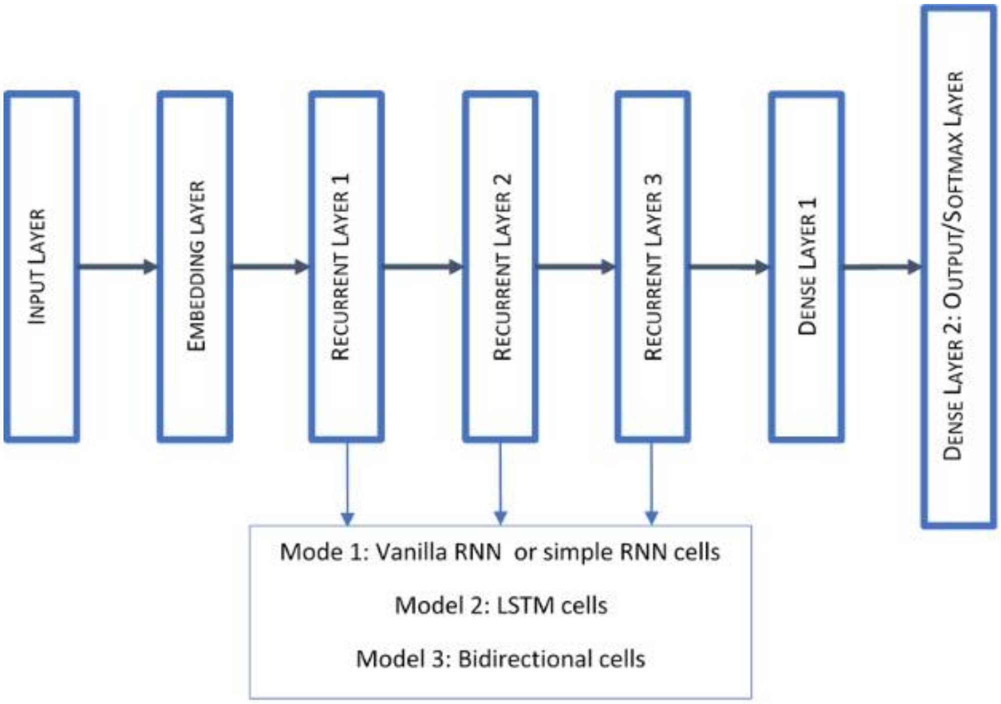


Figure 4.2: Illustration of the simplified model architecture where variable recurrent layer is dependent on RNN cell used.

4.3.2 LSTM and Bidirectional LSTM

More variants of RNNs were developed to address the shortcomings of a simple vanilla RNN. LSTMs are a popular variant of RNNs, each unit of the LSTM is associated with memory typically called - a cell. In a LSTM cell unit, the memory is regulated with the help of three gates namely input gate (i_t), output gate $(o_t$, hidden state $h_t)$ and forget gate (f_t), which helps determine which information needs to be added to the current cell state (C_t) and which information can be forgotten to update the cell state. The equation below represents the data flow from the current cell state, previous cell state and the next state [28].

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right)$$

$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

$$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

The advantages of the LSTMs over vanilla RNNs are that they were specifically designed to overcome the vanishing gradient problems and are considered efficient in capturing long-term dependencies. Due to the vanishing gradient problem in Vanilla RNN models, LSTM were introduced since its ability to update its own cell-state. In LSTM unit (Figure 4.1), the horizontal line on top acts as a “conveyer belt – with some minor linear interactions,” making backpropagation task much simpler [28].

Bidirectional LSTMS capture an idea, an output of recurrent unit at a particular time step not only depends on its past instances (past elements of the sequence) but also on the future instances. The idea of such a network is developed by stacking two layers of LSTMs over each other thus making an output dependent on the computation of hidden states from both LSTM layers as opposed to one in the unidirectional LSTM network.

4.3.3 Convolutional Neural Network and Multi-Filter CNN

Convolutional Neural Networks or ConvNets are popular neural network models that gained prominence in the field of computer vision for their ability to have minimal pre-processing of data compared to other classification algorithms. Additionally, they are able to eradicate the use of primitive hand engineered filters for singular instances of training data and maximize the usage of these neural networks to automatically learn the features from an image/video. ConvNets are generally used as good feature extractors that is successfully able to capture spatial/temporal dependencies of an image/video through the application of filters. Successive applications of these filters in the convolutional layers on any 2D/3D data helps in the reduction of size of the data by retaining all the critical features which are important to make a final prediction. Recently, ConvNets have made a transition to NLP based applications. Instead of a 2D/3D convolutional layer, these applications rely on using 1D convolutional layer. The filter size specified works as a sliding window that rolls over the entire sequence length which is specified as a parameter in the convolutional layer.

The multi filter convolutional neural network was originally proposed by Kim, Yoon in [60]. The filter size mentioned in the convolutional layer of the CNN defines the number of characters to consider in an iteration across the string of characters. The multi filter CNN model in this study only includes different sized filters on the standard two convolutional layered model as illustrated in figure 4.3. Such architectural changes allow the simple CNN model to integrate the different interpretations resulted from processing nucleotide sequence at different resolutions or different n-grams (groupings of characters with sliding window similar to k-mers) due to the variable filter sizes being used.

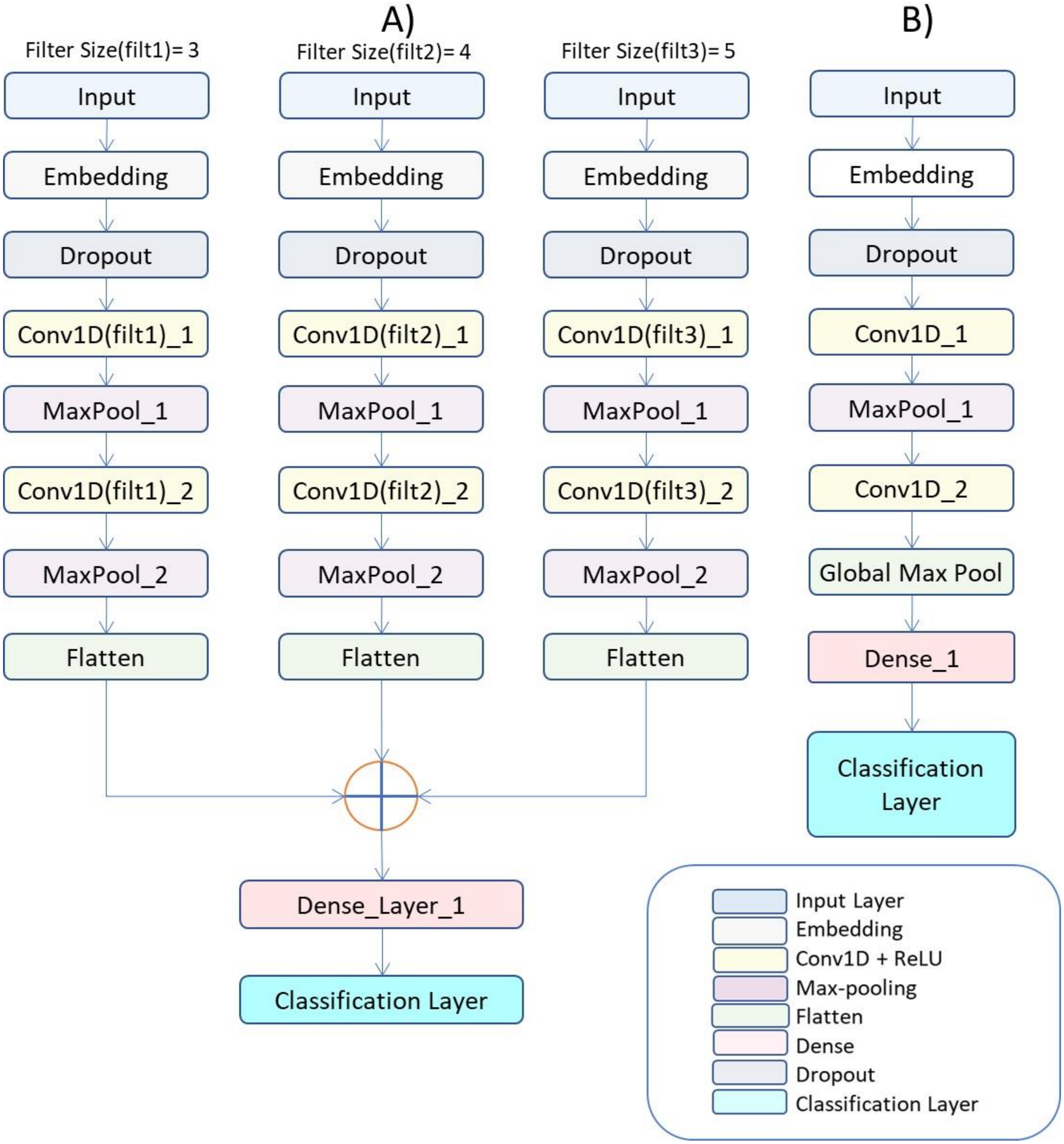


Figure 4.3: A model architecture depicting A) Multi-filter (MF) CNN and B) simple CNN. Global MaxPool layer in B) serves as a flatten layer in this model.

4.3.4 Combinational Model

In addition to aforementioned models, we compared RNN models with a combinational model for each category – family, genus and species. This simple combinational model consists of one single layer of one-dimension convolution, one layer of pooling following with one layer of LSTM unit. These types of combinational models are highly used in text classification tasks and they perform better than a simple CNNs and MF CNN as they combine both the convolution and the recurrent layers in its architecture. This model is depicted in figure 4.4 as below.

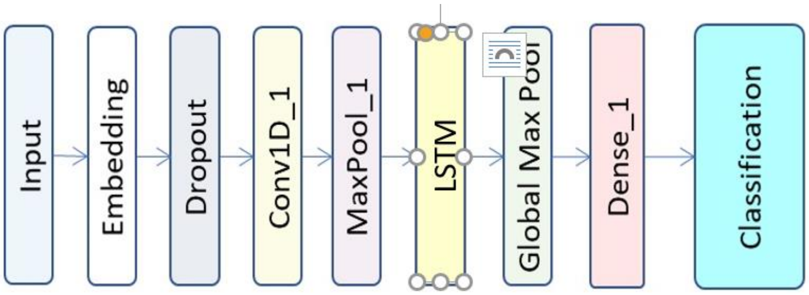


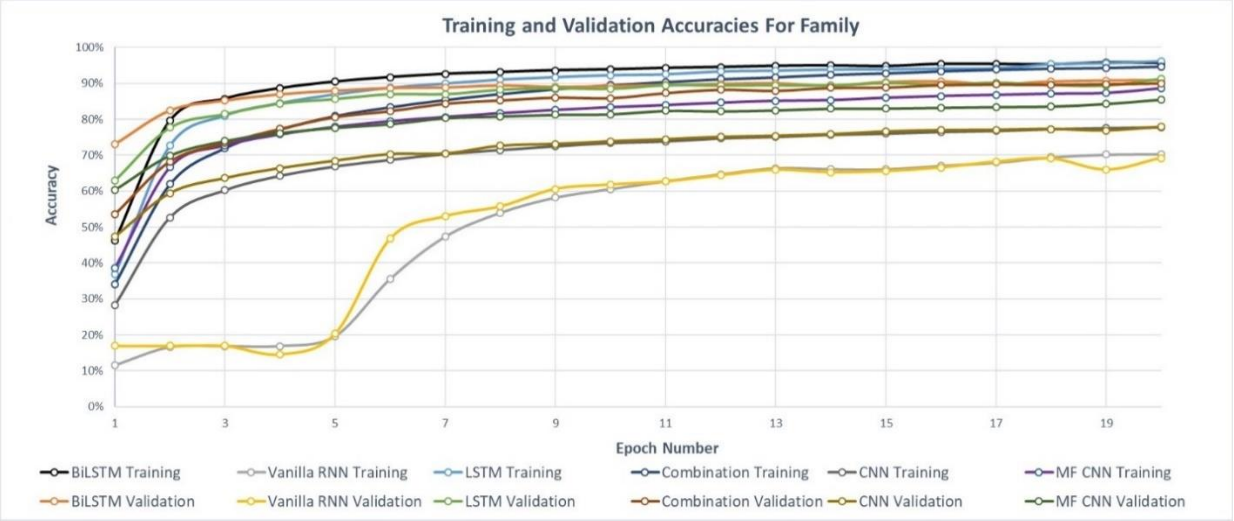
Figure 4.4: A model architecture depicting combinational model. Global MaxPool layer serves as a flatten layer in this model.

4.4 Results

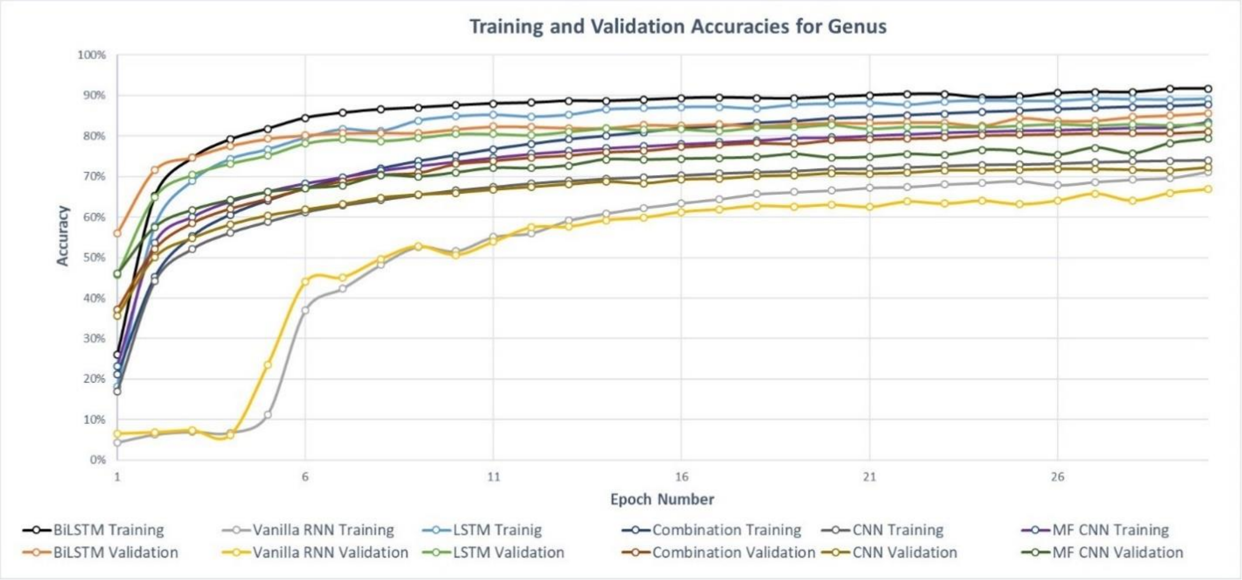
For performance comparisons, each model was evaluated with same hyperparameter for family and genus level classification, whereas species level classification, the model was run for longer number of epochs. Since the number of classes at species level is almost three times higher than genus lever, the complexity of classification also increases exponentially; hence, there is a stark difference in number of epochs hyperparameter. In addition, at species level only five to six sequences are present in certain classes, attributing to additional complexity in achieving better classification accuracies. Each of the below graphs shows the initial results which are comparable to other deep learning models, achieving >85% accuracies for both family and genus and ~70%

accuracy for species level classification [34, 35]. The figure below shows the loss and accuracies of each model for each taxonomic level.

A)



B)



C)

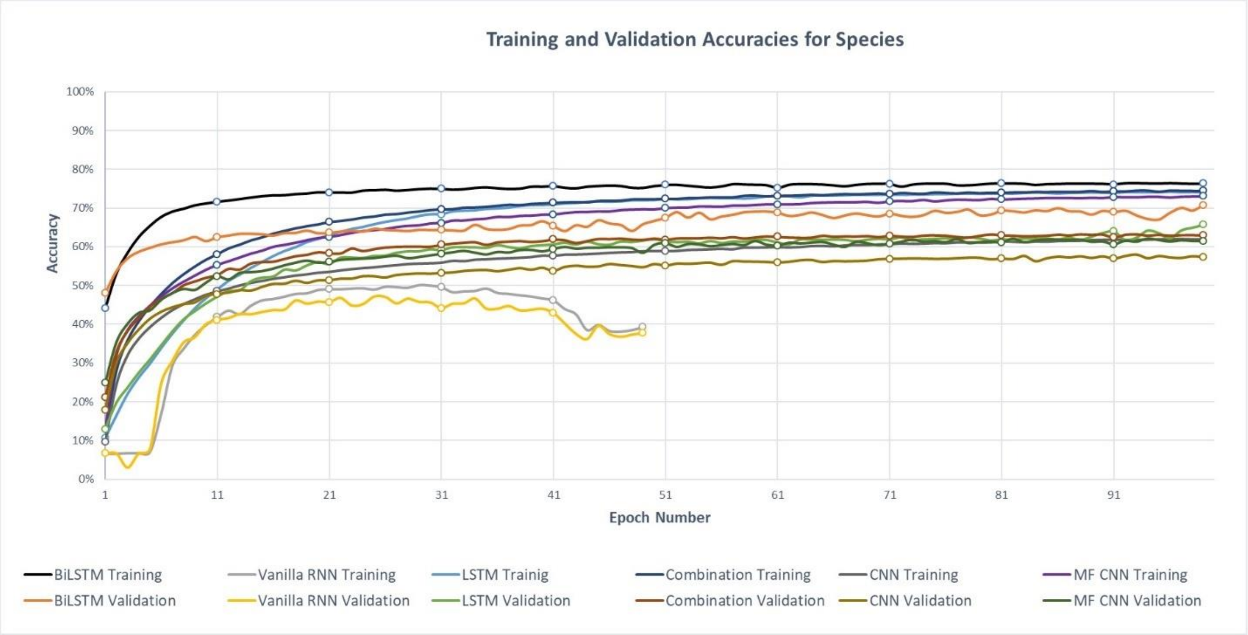


Figure 4.5 The training and validation(testing) accuracies of A) family, B) genus and C) species level classification for all six models.

*Table 4.1 Final accuracies and losses of all six models for each family, genus and species taxonomic levels. Accuracies highlighted in bold are the highest classification accuracy achieved within each level. *Stopped at 50 epochs since this model's performance accuracy was dipping below 40%.*

		Family		Genus		Species	
Model	Model Name	Validation Loss	Validation Accuracy	Validation Loss	Validation Accuracy	Validation Loss	Validation Accuracy
1	Vanilla RNN	1.446	69.21%	1.742	66.91%	3.754*	37.7%*
2	LSTM	0.328	**91.24%**	0.933	82.92%	1.762	65.72%
3	BiLSTM	0.531	90.85%	0.774	**85.63%**	0.914	**70.78%**
4	CNN	1.010	77.91%	1.536	72.25%	3.031	57.13%
5	Multi-Filter	0.746	85.52%	1.275	79.27%	2.686	61.71%

	(MF) CNN						
6	Combination	0.4742	90.00%	1.010	81.11%	2.659	63.01%

The exact accuracies for family models were 90.85%, 91.24% and 69.21%; for genus models were 85.63%, 82.92% and 66.91% and for species were 70.78%, 65.72% and 37.7% achieved with BiLSTM, LSTM and Vanilla RNN respectively. At species level, model was stopped at 50 epochs since the accuracy was descending below 40%. The number of hidden units used in BiLSTM and LSTM models were 500, ran for 20 epochs with batch size of 200, with ‘adam’ optimizer. The values for learning rate was 0.001, beta of 0.9, decay of 0.0 for optimizer. The number of hidden units used in Vanilla RNN models were 500, ran for 100 epochs (model 2, 3) and for 20 epochs (model 1) with batch size of 200, with ‘adadelta’ optimizer. The values for learning rate was 1.0, rho of 0.95, decay of 0.0 for optimizer.

For the fixed input architecture models, that are CNNs, the accuracies achieved did not the outperform the RNN models, especially BiLSTM (model 3) and LSTM (model 2). However, the interesting outcome observed was that combination model (model 6) surpassed the performance of multi-filter CNN model (model 5) as well as simple CNN model (model 4) at all three taxonomic levels included in this study. Comparing the simplistic models from the two architecturally different networks, one can make a conclusion that simple CNN (model 4) outperformed Vanilla RNN (model 1) at all three taxonomic levels. All of the hyperparameters were exactly same as RNNs except for the optimizer, which was ‘adam’ for all CNN models. Table 1 further elaborates the results achieved by all six models.

4.5 Summary

The need of faster, and accurate bacterial sequence classification is a critical task in analysis of metagenomics sequencing for gut microbiome and environmental samples. In recent years, the data repositories for such sequencing projects have been on the rise. The goal of this work was to perform detailed comparative analysis of various deep models and their performances. In general, species level sequences are categorized in almost three times more classes than genus and twelve times more classes than order, hence, the model performance reduces drastically compared to other levels. This is also because bacterial dataset is not balanced; data for some species are easily available than the rare species. Though stacking of deeper LSTM units sounds lucrative in terms of performance for RNNs, the hyperparameter optimization for such network can get difficult. BiLSTM models (model 3) achieved the highest accuracies for genus and species levels; whereas, LSTM models (model 2) achieved the highest accuracy for family level amongst all the models studied. Considering all the CNN models, the combination model (model 6) achieved the highest accuracies at all levels. This work doesn't provide a final tool for analysis of metagenomic samples, but rather serves as a study of possible classification architectures to be included in future endeavors. This work doesn't provide statistical make-up of each metagenomics sample, but provides a comparison of which deep models can be deployed for this classification task. In future, three approaches are being taken to improve accuracies separately and simultaneously, 1) compare our results using a larger dataset as well as raw read dataset, 2) use full gene sequences instead of 100 bp length sequences to create an ensemble model, and 3) stricter data cleansing approaches even if few of the classes are lost. In all the machine learning and deep learning approaches for taxonomy classification task, the relationship of hierarchical data is oversimplified and, in most cases, ignored. In future we plan to incorporate these relationships to improve final accuracy at species level.

5 DEEP ENSEMBLE MODELS FOR 16S RIBOSOMAL GENE CLASSIFICATION

In bioinformatics analysis, the correct identification of an unknown sequence by subsequent matching with a known sequence is a crucial and critical initial step. One of the constantly evolving open and challenging areas of research is understanding the adaptation of microbiome communities derived from different environment as well as human gut. The critical component of such studies is to analyze 16S rRNA gene sequence and classify it to a corresponding taxonomy. Thus far recent literature discusses such sequence classification tasks being solved using many algorithms such as early methods of k-mer frequency matching, and assembly-based clustering or advanced methods of machine learning algorithms – for instance, random forests, naïve Bayesian techniques, and recently deep learning architectures. Our previous work focused on a comprehensive study of 16S rRNA gene classification by implementing simplistic singular neural models of Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs). The outcome of this study demonstrated very promising classification results for family, genus and species taxonomic levels, prompting an immediate investigation into deep ensemble models for problem at hand. In this study, we attempt to classify 16S rRNA gene using deep ensemble models along with a hybrid model that emulates an ensemble in its early convolutional layers followed by a recurrent layer.

5.1 Introduction

In the early millennia, the first ever human genome was successfully sequenced. Ever since, a plethora of sequences including that of microbes, archaea and plants, have been sequenced and publicly made available for various genomic studies. In more recent decades, progressive trend in emerging next generation sequencing technologies have been seen, which vastly enhanced accuracy and rapidness of not only the whole genome shotgun sequencing (WGS) but also targeted gene sequencing or amplicon sequencing (AS) [52]. This phenomenon is noticeable in many areas of bioinformatics, especially in metagenomics. Metagenomics focuses on studying the composition of environmental and human gut samples for abundance and identification of microbiome community and its chronological comparisons [70]. Metagenomics studies are crucial due to their applications in various fields such as ecology, biomedicine, environmental sciences, and microbiology. They are also important for studying gut microbiota for its role in maintaining healthy weight, blood sugar, cholesterol and immune system [71, 72, 73, 74]. One of the most commonly used markers to correctly identify the composition of a microbiome community is 16S ribosomal ribonucleotide acid (rRNA) gene sequence [75]. In every cell of prokaryotic organisms, 16S rRNA gene is part of 30S subunit [75, 76]. This 30S subunit together with 50S subunit makes 70S ribosome –a site of protein synthesis [75, 76]. Because 16S rRNA gene is present in all bacteria and archaea, it serves as an identification card or a biological marker to study the presence of a species/taxa in biological samples. The sequence of 16S rRNA consists of nine hypervariable regions wrapped in between highly conserved regions. These hypervariable (V1-V9) regions make 16S rRNA gene to be rendered as a biological marker [43]. 16S rRNA gene sequencing is preferred due to it having low sequencing cost per Gigabyte, not requiring laboratory cell culture [53, 77] and requiring relatively low input DNA at the beginning [75]. On the contrary to popular belief that metagenomics and 16S rRNA are similar, metagenomics differs from 16S rRNA gene study on an important instance; while 16S rRNA gene study is an examination of relationship among different taxa based on a single gene, metagenomics is a study of all translated genes (entire translated genome) of all microbiomes in a sample [70]. While 16S rRNA gene study allows one to identify underlying taxa composition, it has limitation when the taxa composition of two different samples is predicted to be exactly the same or when two species have a very high sequence identity of >99.5% such as Streptococcus mitis and Streptococcus pneumoniae [75]. In this case, metagenomics whole genome shot gun sequencing may provide with a much deeper resolution of abundance as it sequences all translated genes of all present species including that of low fraction taxa, virus, and fungi. Figure 5.1 depicts an overview knowledge graph of 16S rRNA motivation and classification techniques.

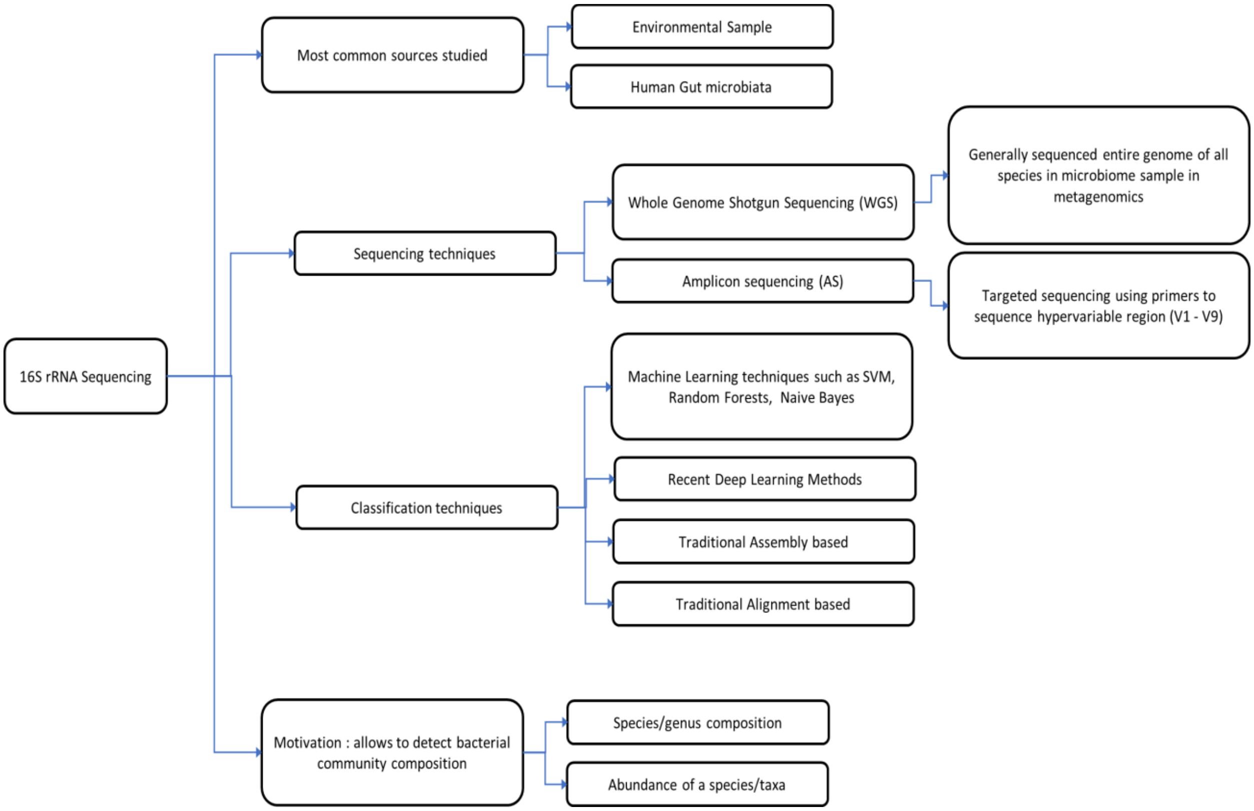

16S rRNA Sequencing
Most common sources studied
Environmental Sample
Human Gut microbiata
Sequencing techniques
Whole Genome Shotgun Sequencing (WGS)
Generally sequenced entire genome of all species in microbiome sample in metagenomics
Amplicon sequencing (AS)
Targeted sequencing using primers to sequence hypervariable region (V1 - V9)
Classification techniques
Machine Learning techniques such as SVM, Random Forests, Naive Bayes
Recent Deep Learning Methods
Traditional Assembly based
Traditional Alignment based
Motivation : allows to detect bacterial community composition
Species/genus composition
Abundance of a species/taxa

Figure 5.1: Overview of 16S rRNA sequencing application and motivation in bioinformatics.

Some of the basic techniques applied for classification of bacterial taxonomy are based on alignment, assembly [34], machine learning, and more recently deep learning. Many bioinformatics applications involve finding sequence similarity and correctly mapping sequences to sequences in known databases. Finding sequence similarity and correct sequence labeling require sequences to be mapped to databases with known sequence taxonomy known as reference genomes. Metagenomic sequences or 16S rRNA gene sequences are thus mapped to reference genomes using alignment algorithms such as Basic Local Alignment Search Tool (BLAST) to classify and measure abundance of taxa; for example, mothur and kraken known to perform read based sequence matching [54, 78]. Second widely utilized technique is assembly based in which first sequence is assembled into entire genome or large contigs and then gene curation is performed by matching predicted genes from contigs to known database. In either case, sequence matching requires some bioinformatics sequence manipulations and analysis. However, in machine learning or deep learning-based techniques, sequence reads or k-mers from sequences can be directly tested on previously trained models, reducing analysis duration. Some of the known machine learning based techniques such as naïve Bayesian, hierarchical clustering, random forests, and support vector machines, also have shown comparable results to aforementioned classifying techniques [38, 56, 57].

The recent advances of various affordable sequencing technologies coupled with the advancements of fast hardware (general-purpose graphic processing units (GPGPUs)), categorical big datasets, open source libraries and improved algorithms have enabled researchers, and scientists to develop multi-disciplinary studies [2]. This hardware acceleration aided in refinement of very powerful deep learning architectures for image and text classification; these discoveries then resulted in the rise of deep learning applications in medical imaging and genomics [34, 58]. Thus far, only a few studies have been published including ours that studies direct classification of 16S rRNA using deep learning architectures. Fiannaca et.al implemented a CNN and deep belief network (DBN) based classifiers for both targeted sequencing and whole genome sequencing taxonomy classification [34]. More recently, Busia et.al. published a study with deep neural network (DNN) classifier that looked at various length sequences to note the performance [35]. Previously, published study’s main goal was to compare performances of deep learning architectures especially of RNNs such as LSTM, BiLSTM with CNNs for 16S rRNA classification task(hpd).

5.2 Method

Method development focuses on dataset preprocessing and proposed deep learning models for 16S rRNA classification task. For all proposed models, input dataset is exactly the same, and tested on same training and validation data split. The overall goal of this study is to be able to create a model that can take raw reads with minimum pre-quality check and trimming requirements. This work implements architecturally four different models, three ensemble models and a hybrid model. The ensemble models average three different deep models, while hybrid model consists of both convolution and recurrent layers. The hybrid model, however, emulates the Multi-Filter model in figure 4.3a of published study [13] for its early convolutional layers with one striking difference: variable length of kernels in Multi-Filter model versus the same kernel sizes in the parallel convolutional branches of the hybrid model. For ensemble models, there are three different intrinsic models involved in making three different combination of models. Next two sections further discuss dataset and implemented models.

5.3 Dataset

The dataset used in this study remained same as previously published study [13]. This manually curated dataset is obtained from Genomic-based 16S ribosomal RNA Database (GRD) [68]. 16S rRNA gene or rDNA sequence length is approximately 1500 base pairs long; however, some of the bacteria can have multiple copies of 16S rRNA gene, hence input sequences from this dataset varies in length from 65 to 2900. Input files are same as [13], consists of two raw files; one containing tab delaminated fasta header with its corresponding bacterial taxonomy and other is fasta header tag with a fasta sequence containing all of the sequences in database. This study also focuses family, genus and species taxa levels as opposed to phylum, class, and order that are known to have >99% classification accuracies. Number of classes at each taxonomic level were 272, 840 and 2456 for family, genus, and species respectively. Approximately ~13,000 sequences were used for training the model and ~3500 for validation, which is 80% - 20 % split for training vs validation dataset. Preprocessing of sequences for input sequences is exactly as first published study, for further details please refer to chapter 4 [13]. The main focus of this study is to demonstrate the effectiveness of ensemble and hybrid models in achieving better classification accuracies compared to simpler deep models.

5.4 Deep Learning Approaches

As discussed in introduction, deep learning models are on the rise with many applications in medical and biological fields. Architectures presented in this study are driven from previous study's results. In study [13], we observe a trend where recurrent models, Bidirectional LSTM and LSTM, outdo convolutional models. The outcome in this study [13] shows singular BiLSTM achieving highest accuracies for genus and species taxa; whereas, LSTM achieved the best accuracy for family taxa. The run time of BiLSTM for ~13,500, 100-character long sequences in training was much higher than of simple LSTM and simple CNN. Hence, in this study, the proposed model architectures are explored to grasp whether proposed models can achieve comparable accuracies as BiLSTM.

One type of proposed model is an ensemble model. Ensemble models have multiple classification algorithms incorporated, allowing them to perform better upon completion as oppose to an individual model [79]. Generally, ensemble model is able to improve accuracy if there is a good amount of variety in model architectures that makes up an ensemble model. In this study four different models – model 1, model 2, model 3 and model 4 – are developed. Model 1-3 are an averaging ensemble models, which are made up using combinations of four intrinsic sub-models : 1) a simple CNN model with two convolutional layers, 2) multi-filter CNN as described in Chapter 4 a hybrid model with two convolutional layers followed by a LSTM layer, lastly, 4) a simple two layers LSTM model. Specifically, model 1 – CNN-MultiFilterCNN-LSTM, consists aforementioned sub-models 1, 2 and 4; model 2 – CNN-CNN-LSTM, consists of two sub-models 1 and one sub-model 4; while, model 3 – CNN-hybrid-LSTM, consists of sub-model 1,3, and 4. These three ensemble models average the output weights of its intrinsic sub-models. However, model 4 is a hybrid model, which is a single model that imitates the multi-filter CNN architecture from [13] in its earlier convolutional layers followed by a recurrent LSTM layer before the softmax classifier. Figure 5.2a illustrates the ensemble model particularly showcasing the model 3, while Figure 5.2b illustrates model 4.

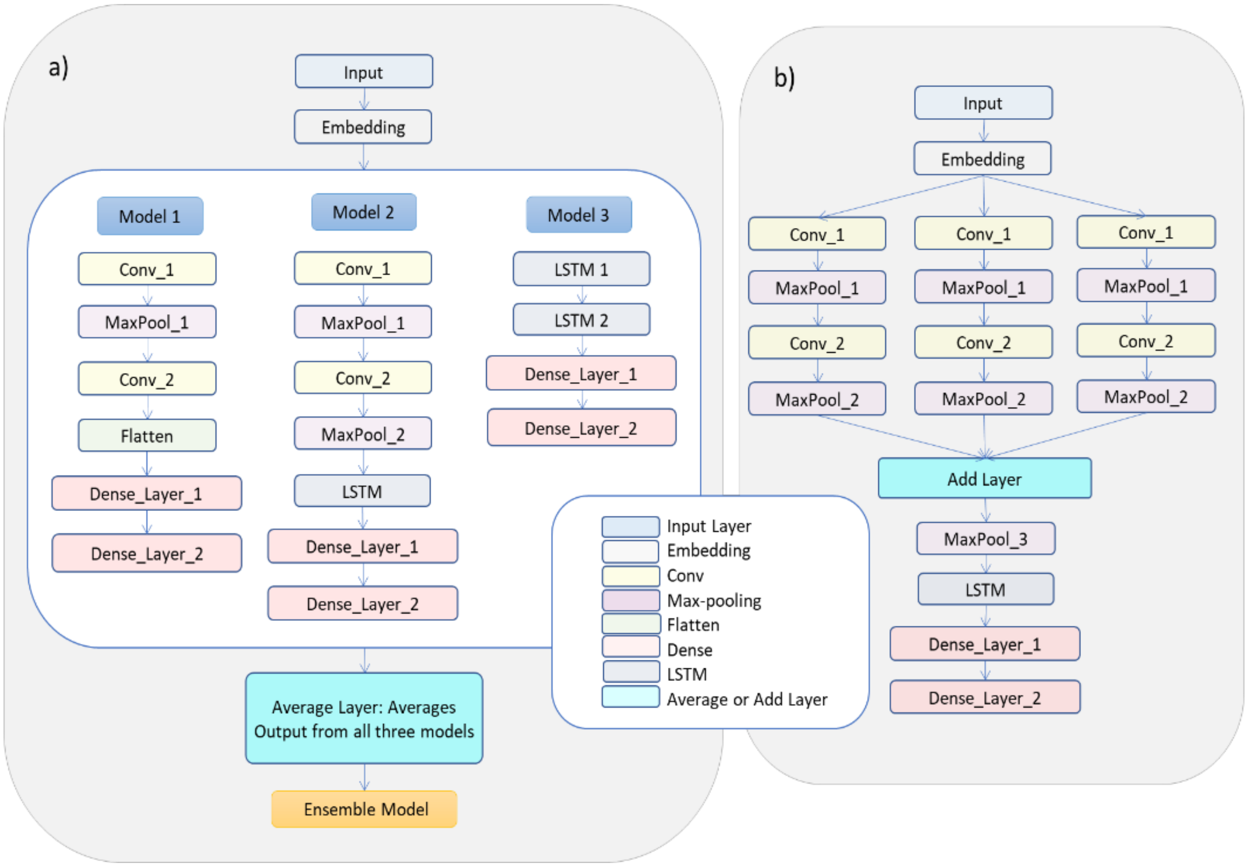


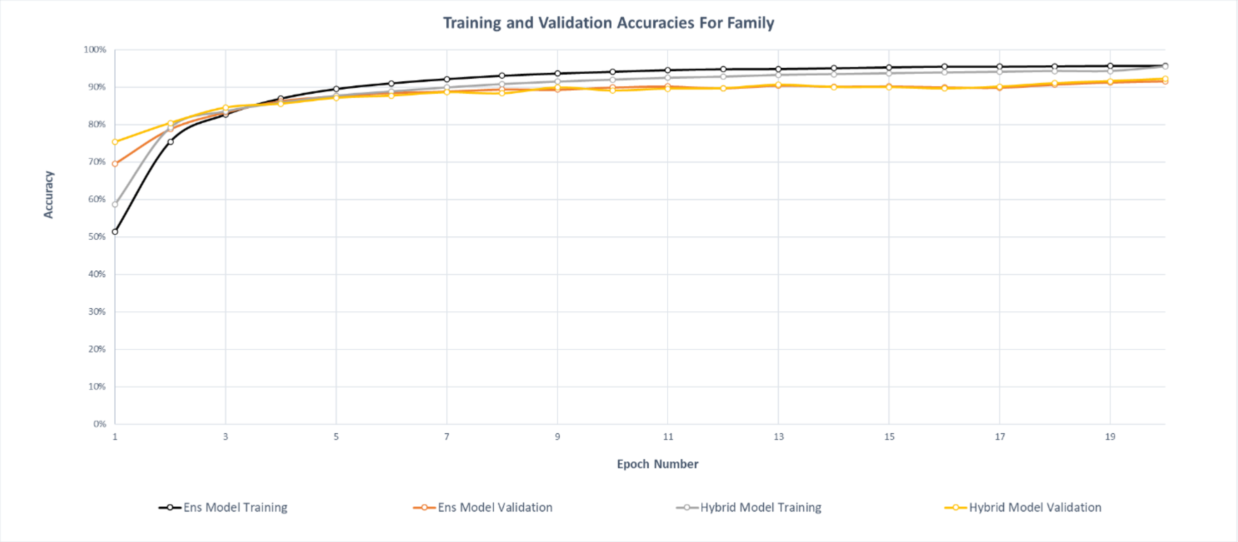
Figure 5.2: Overall architecture of data flow of a) an ensemble model (depicted 3rd combination CNN-hybrid-LSTM), and b) hybrid model used for the classification task at hand.

Model 4 draws its architecture inspiration from the sequence to sequence deep model, which is staple model used for machine translation tasks deployed in many Natural Language Processing (NLP) such as speech recognition, language translation, and Computer Vision (CV) applications like video captioning [80, 81]. Sequence to sequence models are broadly made up with one model that acts as an encoder and another that decodes the output of an encoder. Model 4, however, does not have an encoder-decoder arrangement; it is a singular model that incorporates the convolutional and recurrent layers within its instance.

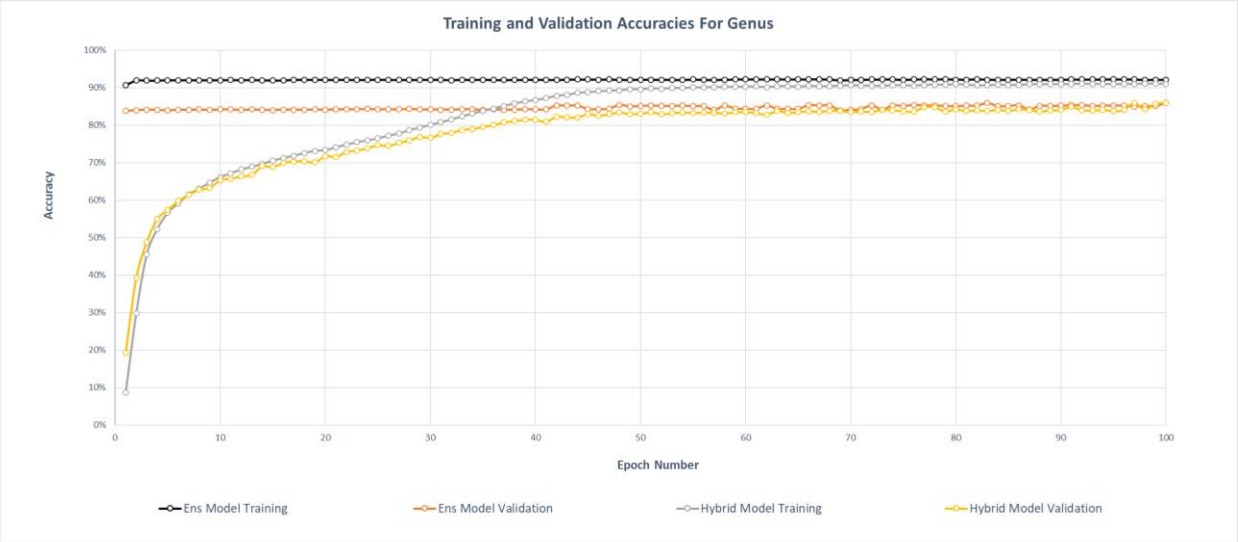
5.5 Results

Neural networks are known for their ability to learn very complex underlying pat-terns from large dataset; however, at the same time, their performance heavily relies on initial training weights as well as balanced un-biased training data. Due to such initial conditions, neural networks are susceptible to high variance, and ensemble models are one of the ways to reduce this variance by combining prediction accuracies of different models. Compared to previous studies, performance of the ensemble models and hybrid model aligns with accuracy greater than 85% for family and genus taxa [34, 35]. For species level, however, these models didn't surpass 70% accuracy achieved in our previous study [13].

a)


Training and Validation Accuracies For Family
Accuracy
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
1
3
5
7
9
11
13
15
17
19
Epoch Number
Ens Model Training
Ens Model Validation
Hybrid Model Training
Hybrid Model Validation

b)


Training and Validation Accuracies For Genus
Accuracy
100%
90%
80%
70%
60%
50%
40%
30%
20%
10%
0%
0
10
20
30
40
50
60
70
80
90
100
Epoch Number
Ens Model Training
Ens Model Validation
Hybrid Model Training
Hybrid Model Validation

c)

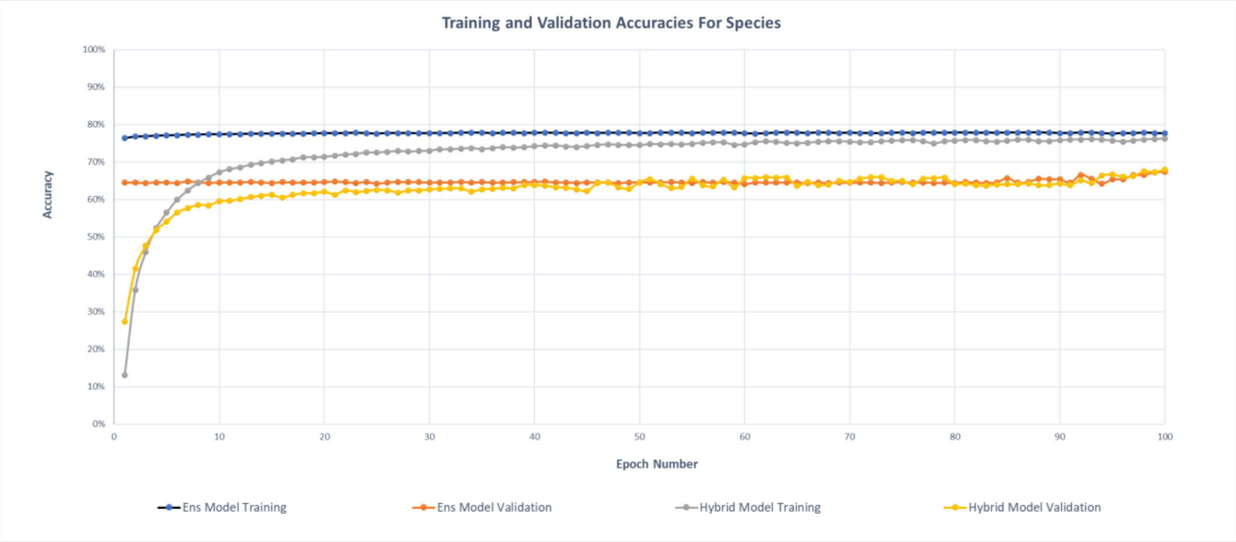


Figure 5.3 The training and Validation(testing) accuracies of classification for family (a), genus (b) and species (c) level including both hybrid model and ensemble model. For ensemble model, accuracies shown here are from the best performing CNN-hybrid-LSTM model.

Figure 5.3a, 5.3b and 5.3c shows loss and accuracy curves for family, genus and species taxa respectively. These figures only show model 3 (averaging ensemble) and model 4 (hybrid) curves since they achieved the highest accuracies. As described in table 5.1 below, the highest validation accuracies for family and species taxa are 92.22% and 67.95%, achieved with hybrid model. However, at genus level, the highest validation accuracy achieved was 85.98% with CNN-hybrid-LSTM model and second highest validation accuracy of 85.94% with hybrid model. Even though, model 3 and model 4 outperformed previously obtained classification results for family and genus taxa, both models failed to outperform at species level, but stayed within 3% percentile range. All of the models in this study have comparable outcomes within 1-2% accuracies obtained for all taxa amongst each other, unlike our previously explored simplistic single models [13]. This agrees to the notion that ensemble and/or hybrid models tend to achieve better performance predictions than any singular model.

Table 5.1 Final accuracies and losses of all four models for each family, genus and species taxonomic levels. Accuracies highlighted in ***bold*** *are the highest classification accuracy achieved within each level. Precision, Sensitivity, F1 Score, Specificity, and Accuracy Data of Binary Classification Models*

Model Info		Family		Genus		Species	
No	Name	Val_Loss	Val_Acc	Val_Loss	Val_Acc	Val_Loss	Val_Acc
1	CNN-MF-CNN-LSTM Ensemble Model	0.5760	90.20%	1.2330	85.76%	2.6654	66.86%
2	CNN-CNN-LSTM Ensemble Model	0.7239	91.33%	1.1342	85.80%	2.2033	67.15%
3*	CNN-hybrid-LSTM Ensemble Model	0.4670	91.60%	1.0007	**85.98%**	2.0057	67.39%
4*	Hybrid model	0.5231	**92.22%**	0.9988	85.94%	1.9226	**67.95%**

All models are trained using the same hyperparameters for all taxa classification at hand, except for epochs and non-linearity function. For family taxa classification, all four models are run with 20 epochs; whereas, for genus and species taxa, the models ran for 100 epochs until we saw no further improvements on the outcome loss and accuracy curves. For all LSTM layers, the number of hidden states is set to 500. For all models, the batch size used is 128, with ‘adadelta’ optimizer, learning rate applied is 0.01, and momentum of 0.0. Further hyperparameter optimization is an open avenue for ongoing improvisation.

5.6 Future work for this study

In this study, input reads are hundred base pair long, in other words, input is hundred characters long string; however, the model can easily be adapted for longer or shorter read lengths. With deep learning architectures such as recurrent neural networks, longer strings may provide a finer representation of features in recognizing underlying patterns. These taxonomical data consist of a hierarchical relationship which cannot be used to find abundance of sequences in a sample, but it can certainly be used for sequence classification tasks. Next steps involve of using such information along with higher dimension input feature vector of different sequence regions to further improve accuracies at Family, Genus and Species taxa levels.

6 ALZHEIMER DISEASE CLASSIFICATION USING DEEP LERNING

6.1 Introduction

After the rise of powerful deep learning architectures, researchers started making more and more datasets available to public after anonymizing patient data. This led to major dataset availability in brain imaging for Alzheimer’s disease detection, such as, Alzheimer’s Disease Neuroimaging Initiative (ADNI) [82] , Open Access Series of Imaging Studies (OASIS) [83], and Australian Imaging Biomarkers and Lifestyle Study of Ageing (AIBL) [84]. In last four years, there have been over forty research studies have been published that takes different deep learning approaches for solving Alzheimer’s disease classification.

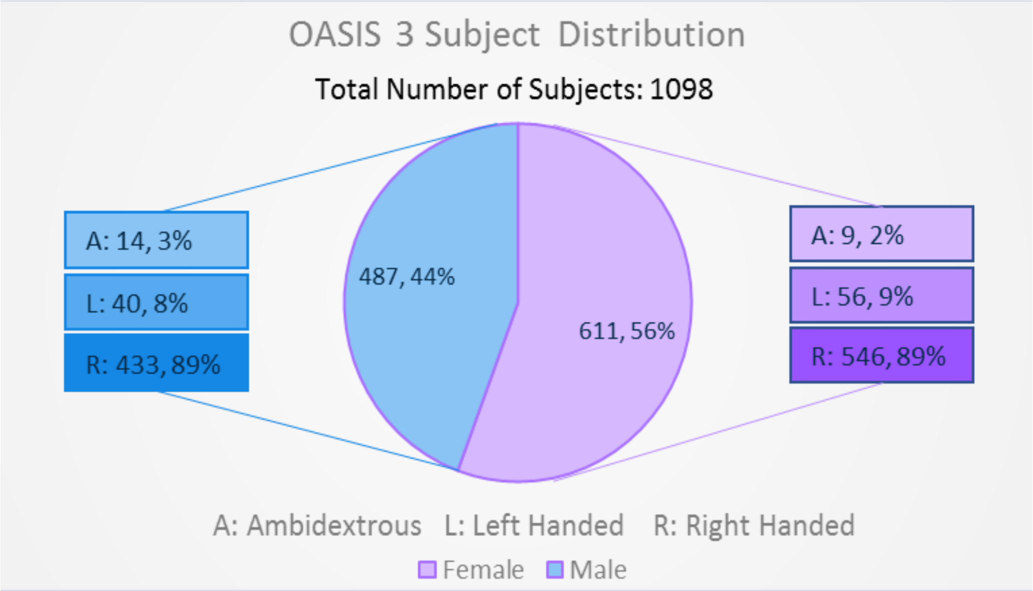
Alzheimer’s disease classification task mainly divided in three different ways, 1) studies which have multi-class classification and not just superficial presence of disease classification, 2) studies that have separate classification to look at different progression of classes against normal and against each other and lastly, 3) studies that uses classification to predict the progression level of disease based on previous image scans using longitudinal data. Another approach that can split such studies is whether the studies use multi-modality dataset or single modality dataset. In multi-modality data, the most common dataset involves of T1-weighted MRI scans, mostly axial plane, and FDG-PET scans. There is only one such study that utilizes CSF biomarkers, demographic information, cognitive evaluation along with medical imaging to predict the progression of the disease [85]. Benefit of involving such pathological markers besides using imaging resources is that such data contains relevant and complementary clinical outcome that can only enhance the performance of the model theoretically [86]. Majority of studies have binary classification of progression classes against each other or against normal class.

In our approach, we perform multi-data, multi-class and binary classification separately with necessary preprocessing steps and several neutral network architectures for performance optimization. Multi-data as discussed further, involves of all three planal data of T1w MRI scans, and other demographic, cognition and brain mass data that is available for each subject.

6.2 Raw Dataset Preprocessing and Input Dataset Preprocessing

Detailed preprocessing of dataset handing has been described in this section. The dataset was obtained from OASIS-3 as it is least utilized and consists of wide-variety of larger scale different imaging modalities as well as multiple scans per subjects [83]. The original dataset has 1098 subjects, involving of 56% female to 44% male.

a)



b)

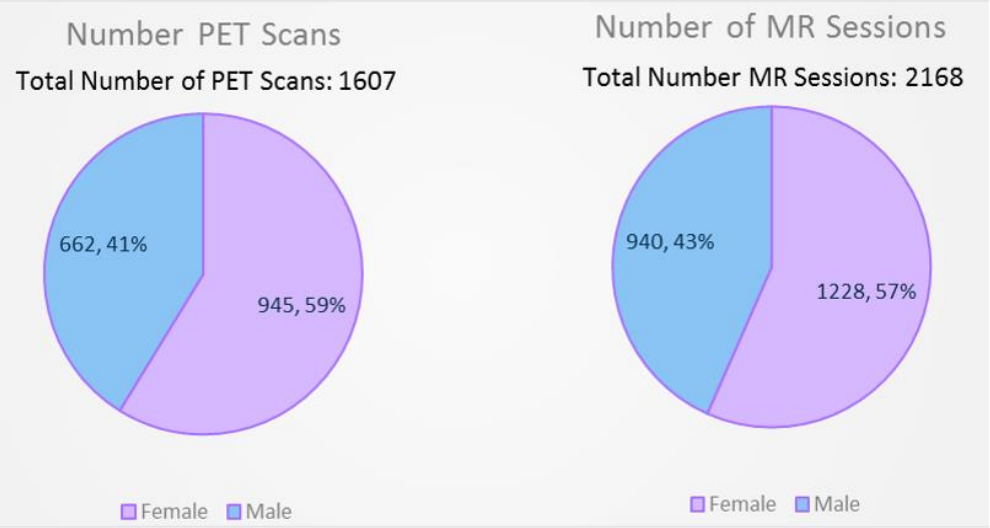


Figure 6.1 a) OASIS-3 dataset original subject distribution b) OASIS-3 number of PET and MRI sessions amongst male vs female distributions

Out of those, both female and male share very similar ratio of ambidextrous, left and right handedness at ~2%, 9% and 89% as shown in figure 6.1 section a). Looking further into imaging sessions, figure 6.1-part b), there are 1607 PET sessions and total of 2168 MRI sessions available in overall data, again well distributed amongst male and female.

Additionally, all clinical data that was provided in clinical sessions for each subject was closely looked at for machine learning models. A correlation between multiple clinical raw data was performed to select some of the demographics that would impact the outcome like education years, age at entry, Mini Mental State Exam score (MMSE), total cortex volume total gray matter volume, etc

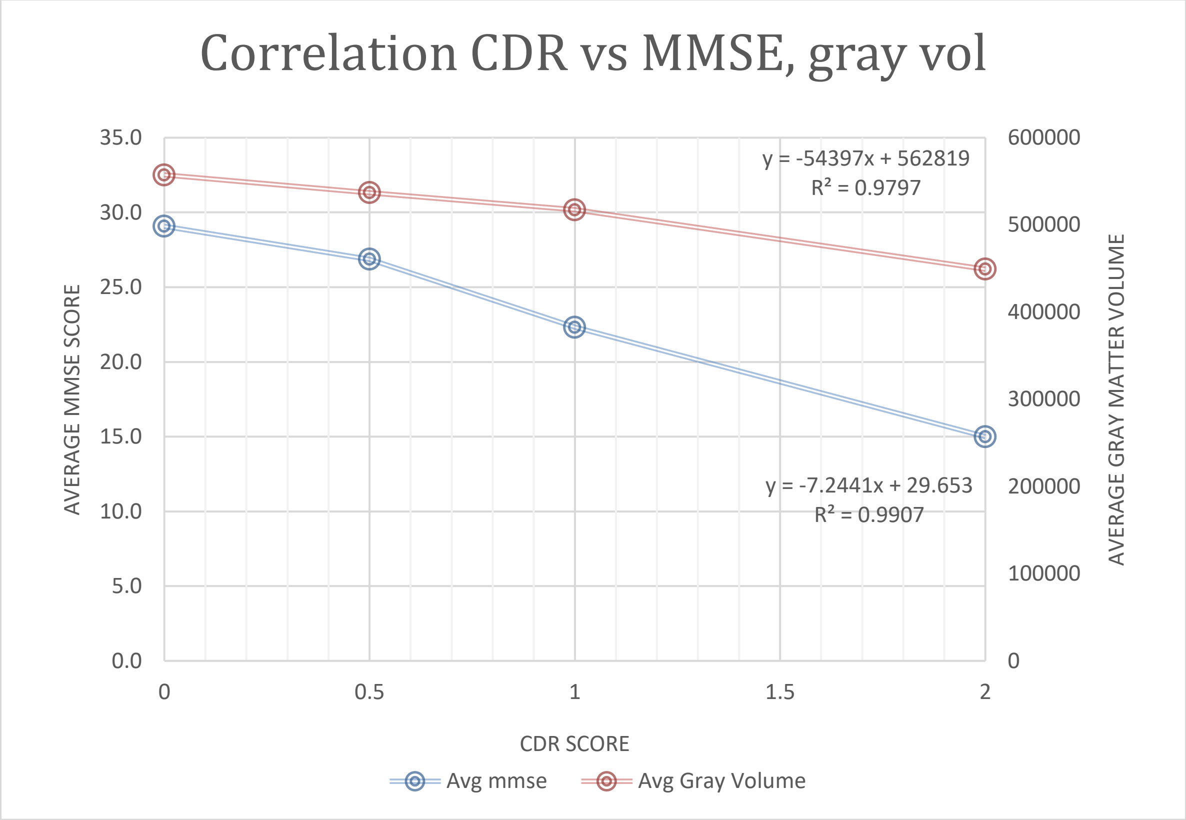


Figure 6.2 Correlation of CDR vs Mini Mental State Exam (MMSE) and gray matter volume that shows a very high r-squared value of 0.9797 and 0.9907 respectively. A lot of other datapoints were measured for similar correlation, however, not all data had high correlation values.

One such correlation is shown in figure 6.2 as seen above. Two of such graphs shows that average MMSE score and gray matter volume are highly correlated with average CDR score at correlation coefficient of 99% and 97.9% respectively. Hence, finally three of such additional data parameters, MMSE score, total cortex volume and gray matter volume, which serve as additional clinical information related to disease prognosis were included in machine learning model 5M as inputs for classification.

OASIS dataset does not provide class-labels for each scan sessions, it has a separate clinical session information which has class related information based on five-point scale Clinical Dementia Rating (CDR) score. This scale defines the class labels, 1) 0 – cognitively normal (CN),

2) 0.5 – very mild dementia (vMCI), 3) 1 – mild cognitive dementia (MCI), 4) 2 – moderate dementia (Alzheimer's disease) (AD) and last, 5) 3 – severe dementia. To match MRI/PET sessions with clinical sessions, there is another processing step, that can relate clinical diagnosis to imaging scans. The authors of OASIS dataset suggest using a window of six months to a year before or after session date as a criterion to select corresponding clinical sessions. Selected clinical session match to be considered positive if it was closest to three hundred and sixty-five days before or after the imaging session date. By this pre-processing step, finally 1952 sessions out of the original 2168 MRI sessions.

Raw Data Pre-Processing
Input Data Pre-Processing
Model Architecture
Test Data Analysis

Figure 6.3 Process flow of deep neural network models

As shown in figure 6.3 above, pipeline of neural network models for Alzheimer's disease classification involve following steps: 1) raw data pre-processing, 2) input data pre-processing, 3) model architecture and 4) test data analysis. Along with dataset also provides the processed MRI and processed PET files of FreeSurfer and PUP, which is beneficial to assess preprocessing of raw images as well as to eliminate and use directly pre-processing files.

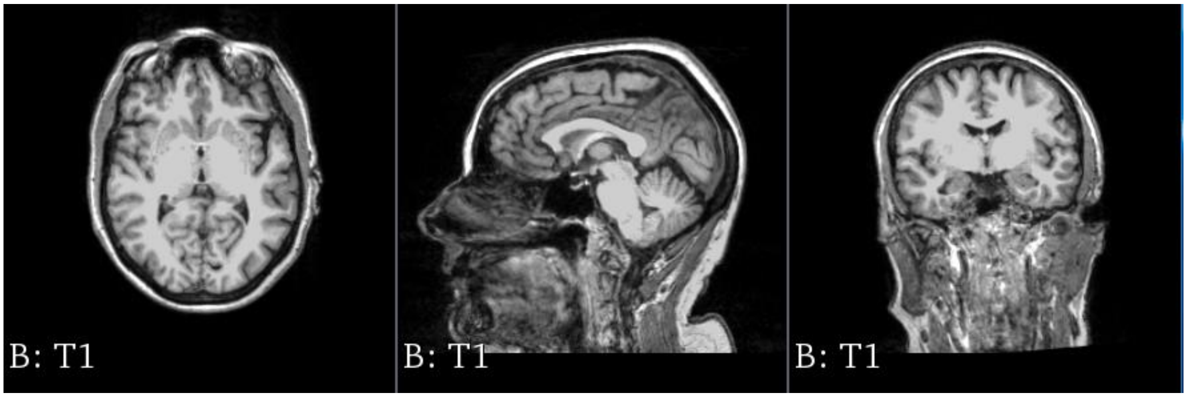


Figure 6.4 An example of raw T1 MRI scan from OASIS-3 dataset without all of the pre-processing steps of intensity corrections, noise reduction, motion correction, skull stripping and

normalization. Left most section shows axial plane, middle section shows sagittal plane and right most section shows coronal plane.

Preprocessing of nii raw data files for each structural MRI modality was performed using state-of-the-art algorithms for intensity non-uniformity, noise reduction, motion correction, skull stripping and intensity normalization. Figure 6.4 above shows the representation of a raw MRI planes. These results are then compared with FreeSurfer brainmask.auto.mgz file for correctness. Raw data pre-processing involves taking all selected clinical session's MRI data files (*.mgz) and extracting slices for Axial, Sagittal, and Coronal planes. For each of the axial, coronal and sagittal planes, 40 middle slices (90-129) were picked as shown in figure 6.5, containing most of the hippocampus region of cortex in brain as it is known to shrink in Alzheimer's patient [87].

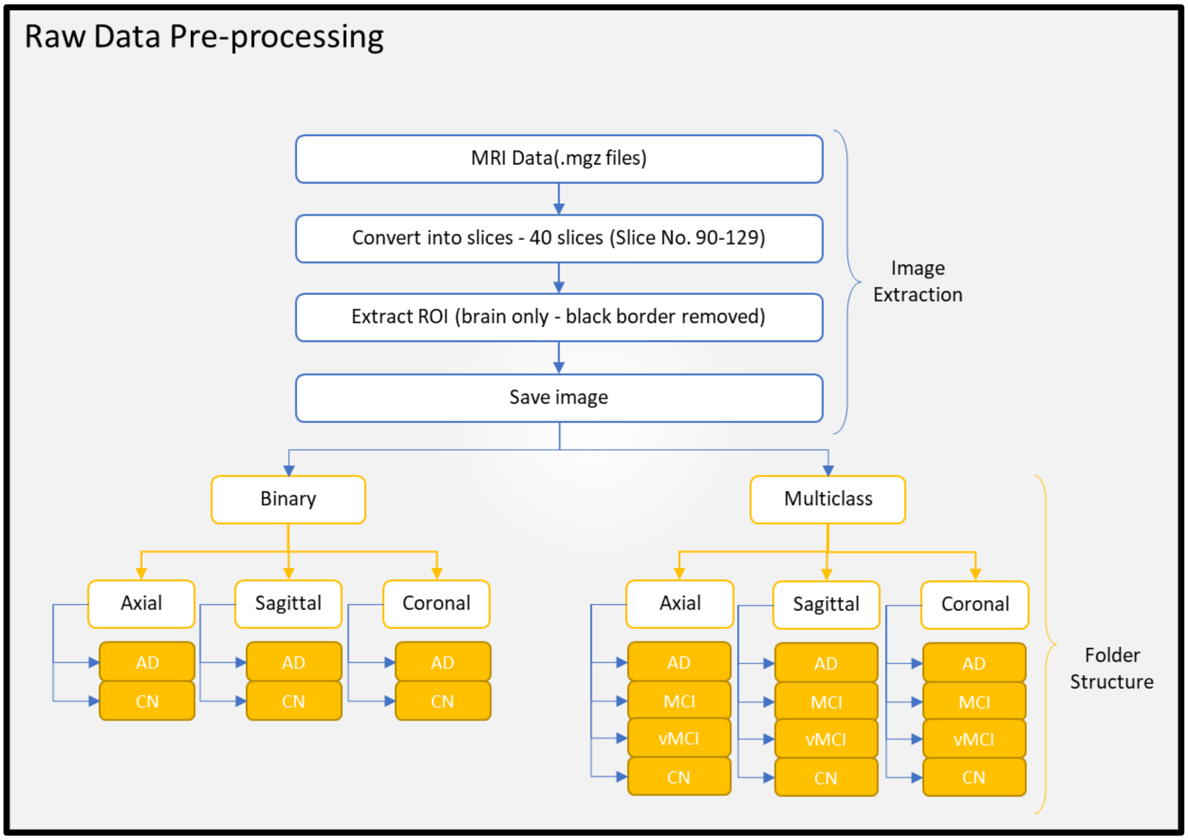


Figure 6.5 Raw data preprocessing: image extraction from MRI (.mgz) files and image storage folder structure

Once the slices are extracted, region of interest (ROI) is extracted; black boarder pixels are removed from the picture in order to extract just the area of brain image pixels. These ROI extracted image files are then saved to an organized folder structure. Folder structure contains 3 levels of subdirectories. Level 1 directory contains subdirectories called “Binary” and “Multiclass” to distinguish binary from multiclass data. Level 2 directory contain subdirectories called “Axial”, “Sagittal”, and “Coronal’ directories. Level 3 directory consists of “CN” and “AD” for binary classification; whereas, for multiclass classification, the directory consists of “CN”, “vMCI”, “MCI”, and “AD”. These level 3 subdirectory names correspond to CDR score of 0, 0.5, 1 and 2 respectively, which is the y input label for the neural network models.

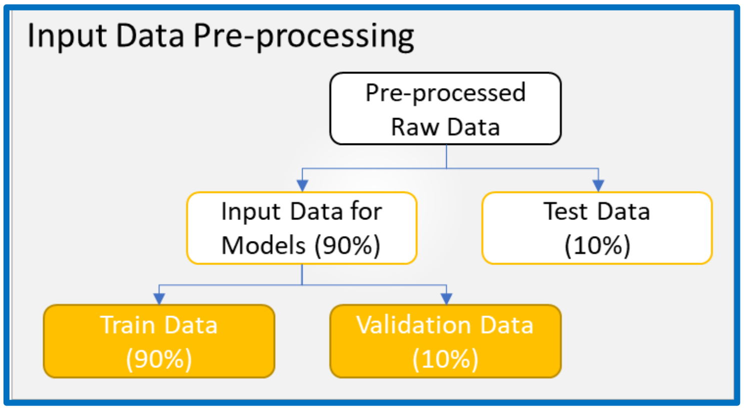


Figure 6.6 Input data preprocessing: splitting of training and validation data

As show in figure 6.6 above, these saved images for individual planes are then converted into an individual pixel-array for further processing in model 1B-3B and 1M-3M. For model 4B and 4M, images are matched based on the slice number for all three planes to ensure x-label doesn’t have any inconsistencies while creating an array input. The data have been first split in to input data (90%) and test data (10%). The input data is further split into training data (90%) and validation data (10%). Hence, the dataset is a 3-way split with 10% of data as an unseen test data. The details of data pre-processing and input data pre-processing for deep learning architecture are further discussed in section 6.3.

6.3 Multi-Data Deep Neural Network Model Architecture and Machine Learning Models

To include T1w MRI modality as well as other clinical information separately in deep learning models and machine learning models, we designed our architecture as below in figure 6.7. There are eight neural network models investigated in this chapter, 4 models (model 1B, 2B, 3B, 4B) are for binary classification and 4 models (model 1M, 2M, 3M, 4M) are for multiclass. Additionally, two machine learning models (model 5B, and 5M) are investigated for clinical data related to MRI image sessions; one for binary and another for multiclass.

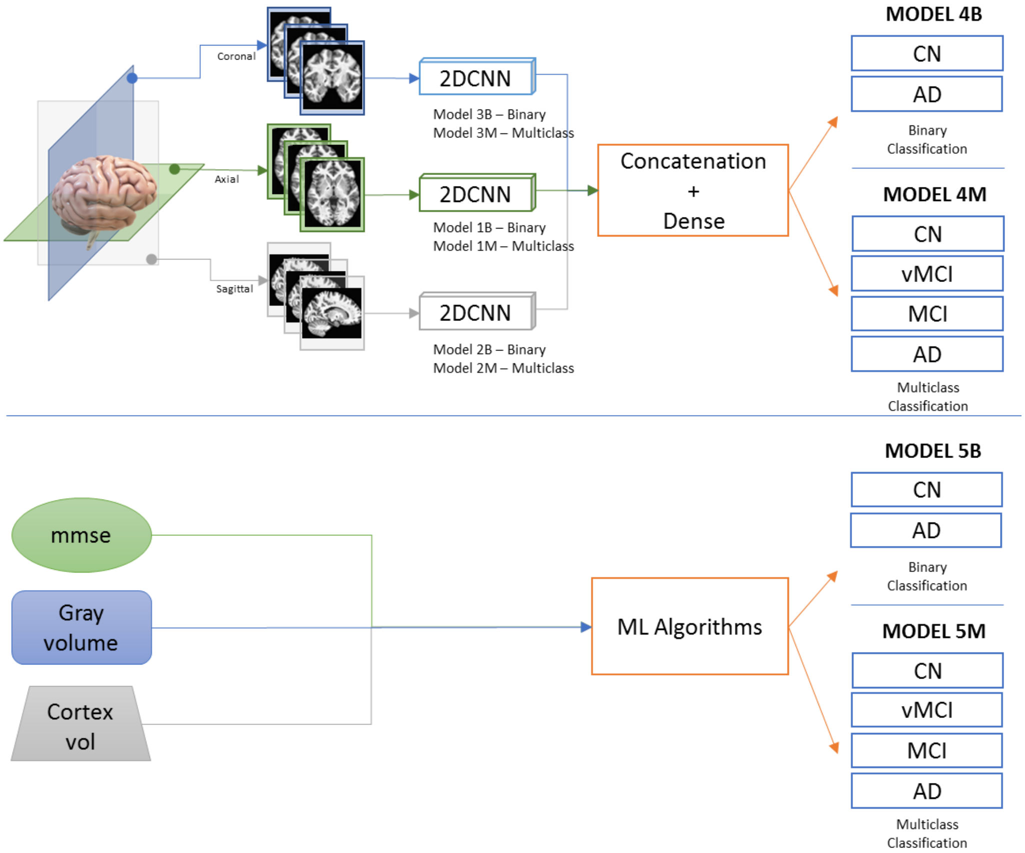


Figure 6.7 An overview of multi-data convolutional neural network framework

Firstly, Model 1B-1M, 2B-2M and 3B-3M take T1w MRI planar slices Axial, Sagittal, and Coronal images, respectively, as data for input data preprocessing. Each of those slices then go through a 2DCNN unit, which has two convolutional layers, dropout, max-pooling and ReLU non-linearity layers as feature extractors. Many classifier validation experiments are run to find the optimal values of hyperparameter for each binary and multiclass model. On top of it, two loss functions are utilized: cross-entropy sigmoid and cross-entropy softMax. Model architectures of model 1B, 2B, 3B, 1M, 2M and 3M are shown in figure 6.8.

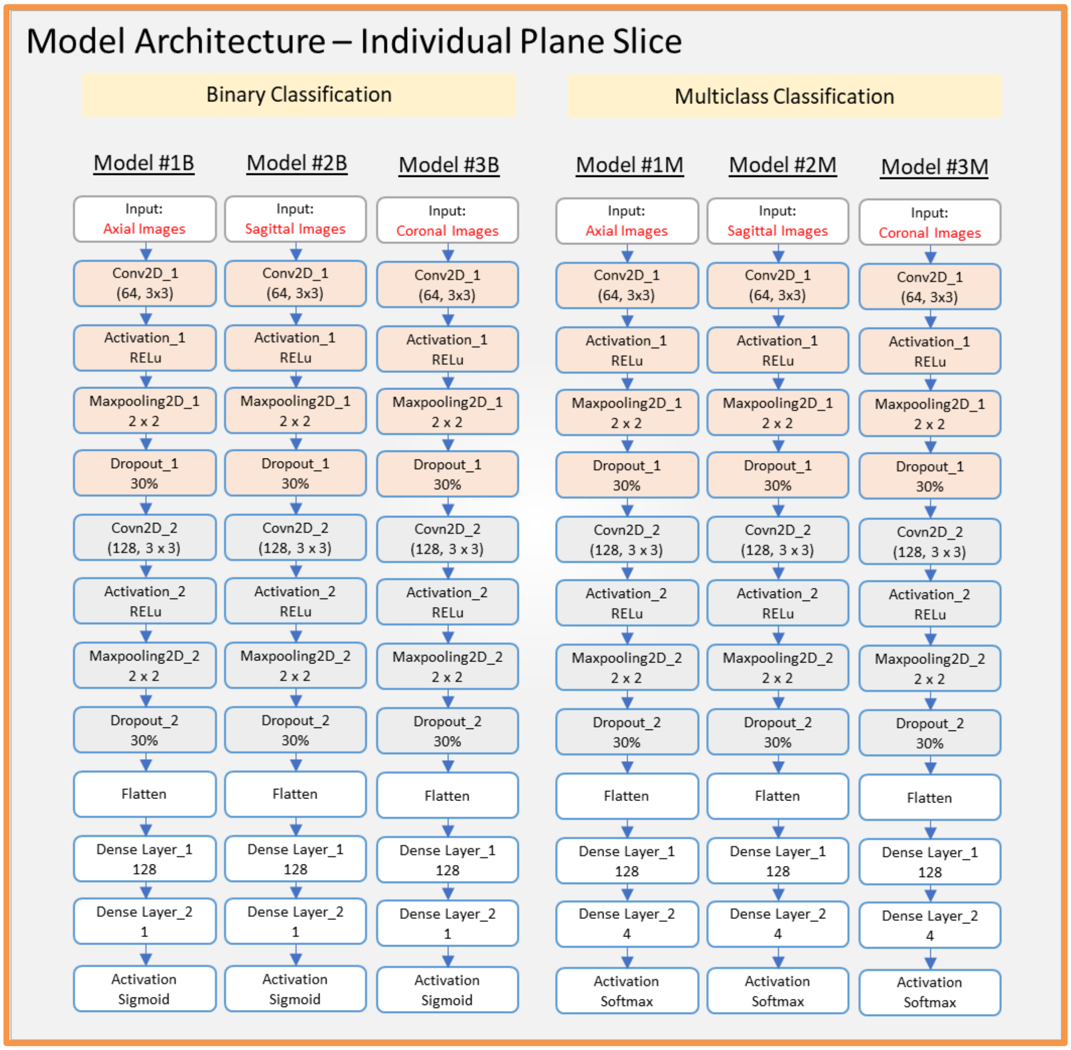


Figure 6.8 Model architectures of individual plane slice for binary and multiclass classifications

These three are individual models and their outputs are also individual scores of classifications based on single planar out of the three, coronal, axial, and sagittal. All these features are concatenated and goes through another two fully connected dense layer with a sigmoid classification for models 1B-3B and softmax classification for models 1M-3M. This will output the probability of input image belonging to one of the classes, i.e CDR score.

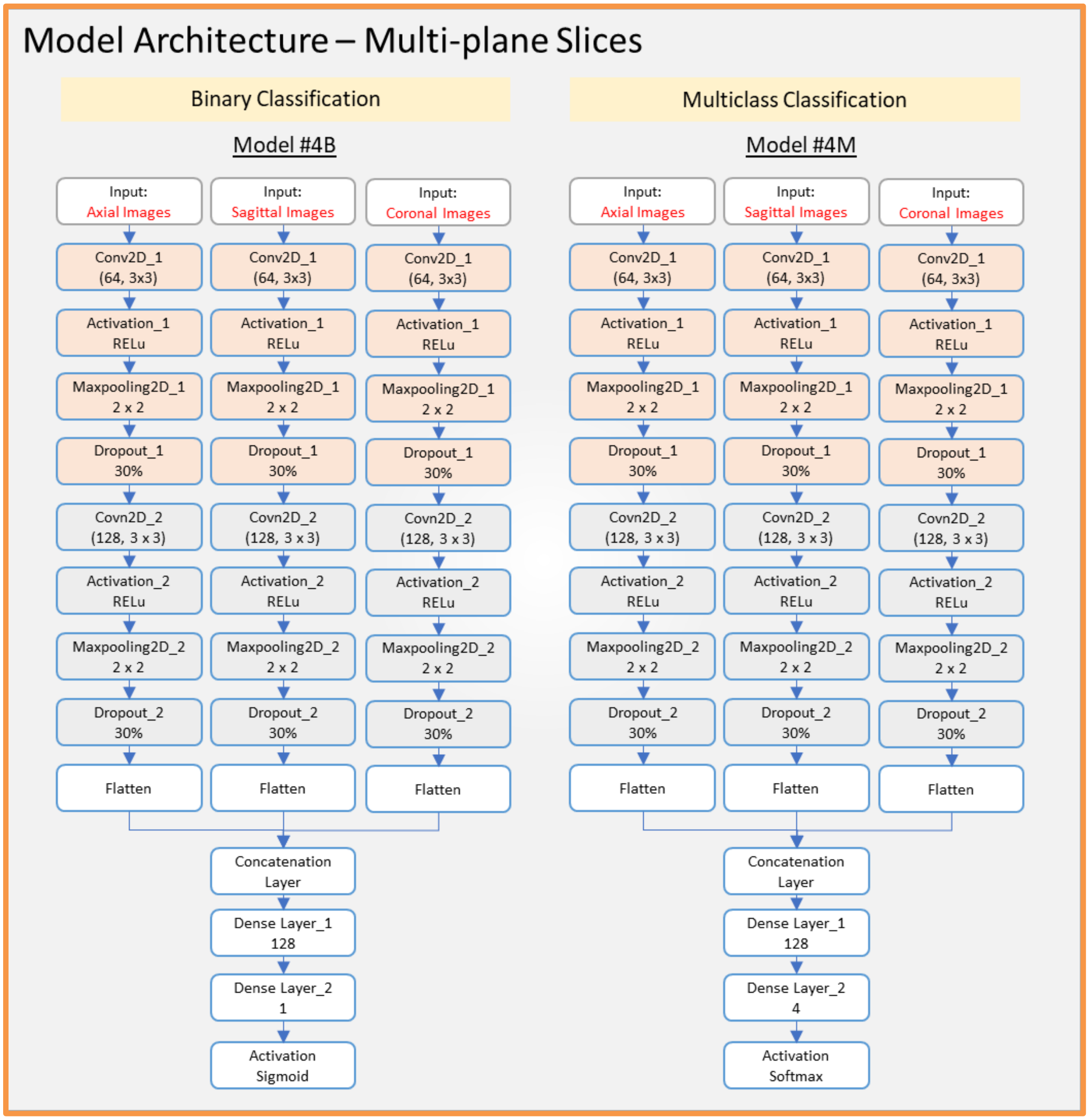


Figure 6.9 Model architecture of multi-plane slices for binary and multiclass classifications

Whereas, Model 4B-4M take weights from model 1B-1M, 2B-2M and 3B-3M flatten layers and concatenate these feature vectors before dense layer as shown in figure 6.9. For fair comparisons of models, in this chapter, the best individually working model was used for concatenated models 4B and 4M.

For machine learning models, model 5B and 5M accepts, MMSE score, total gray matter volume, and total cortex volume as input parameter x and CDR scores of corresponding x as label for y. For machine learning models, six different classifiers were explored, 1) multilayer perceptron (MLP), 2) support vector machine (SVC), 3) random forest classifier, 4) decision tree classifier, 5) k-nearest neighbor classifier and lastly, 6) gaussian naïve Bayes.

6.4 Results

This section includes the initial results of all of the ten models discussed above in section 6.3. First, machine learning model outcomes are discussed and then deep learning model outcomes are reported.

6.4.1 Outcomes of Machine Learning Algorithms

Regression analysis was performed to find the highly correlated clinical parameters with CDR score which can serve as an input candidate for model 5B and 5M as described in section 6.2. Table 6.1 shows the breakdown of final dataset based on male, female and all subjects. It also shows the resulting average MMSE, gray matter volume and cortex volume for each corresponding class with CDR score. The normal MMSE score is calculated out 30, hence higher MMSE score is an indicator of normal cognition. The average standard MMSE score also have a criterion based on educational level, however, for simplification purposes, only CDR score is used as a discriminatory parameter for classification

.

Table 6.1 Subject demographics based on CDR score for number of MRI scans, MMSE score, Age at scan and entry, gray matter volume and education level scores. Except for MRI scan total, all other demographics averages were calculated alone with standard deviations. OASIS-3 dataset only provides with age at entry, hence firstly age at scan for each individual subject was calculated based on number of days between corresponding subsequence scans, following with averaged out age based on CDR scores.

	CDR Score	No. MRI Scans	MMSE	Age at Scan	Age at Entry	Gray Matter Volume	Education
All	0	1549	29.07 ± 1.57	69.7 ± 9.3	65.6 ± 9.1	557360 ± 55799	15.9 ± 2.7
	0.5	302	26.87 ± 2.79	75.6 ± 7.4	72.4 ± 7.4	536946 ± 54819	15.0 ± 3.0
	1	94	22.32 ± 3.98	75.6 ± 8.6	73.5 ± 8.7	517352 ± 58994	14.4 ± 3.2
	2	7	15.00 ± 4.73	71.1 ± 10.7	68.9 ± 10.5	449228 ± 47783	16.0 ± 2.3
Female	0	938	29.14 ± 1.73	69.2 ± 9.3	65.0 ± 9.1	535473 ± 45259	15.4 ± 2.8
	0.5	135	26.78 ± 2.95	74.3 ± 7.8	70.8 ± 7.9	507102 ± 45679	14.0 ± 3.0
	1	42	21.88 ± 4.57	75.1 ± 9.2	73.7 ± 9.8	484665 ± 51409	14.3 ± 2.7
	2	3	12.00 ± 4.58	69.2 ± 16.3	66.7 ± 14.4	407205 ± 32724	15.3 ± 3.1
Male	0	611	28.96 ± 1.30	70.4 ± 9.3	66.6 ± 9.0	590960 ± 53734	16.5 ± 2.5
	0.5	167	26.94 ± 2.66	76.7 ± 6.8	73.7 ± 6.6	561072 ± 49547	15.8 ± 2.8
	1	52	22.67 ± 3.43	76.0 ± 8.1	73.4 ± 7.8	543753 ± 51294	14.4 ± 3.5
	2	4	17.25 ± 3.86	72.6 ± 6.5	70.6 ± 8.5	480746 ± 27607	16.5 ± 1.9

Also, OASIS dataset only provides age at entry. Average age at scan is calculated for each scan per subject based on number of days at scan added to age of entry at scan 0. From the age distribution, female subjects seem to be participating at least two years prior to male subject, however, there is no correlation amongst the subject age and CDR score. Number of subjects with moderate dementia and Alzheimer’s disease with CDR score of 2 are very low, however, one of the interesting finding for original dataset is that if a subject at entry already had a CDR score of 2 or 3, there was no further testing performed. This creates a bias in the original dataset as the patient is already classified with a positive Alzheimer's disease and would not help in longitudinal study.

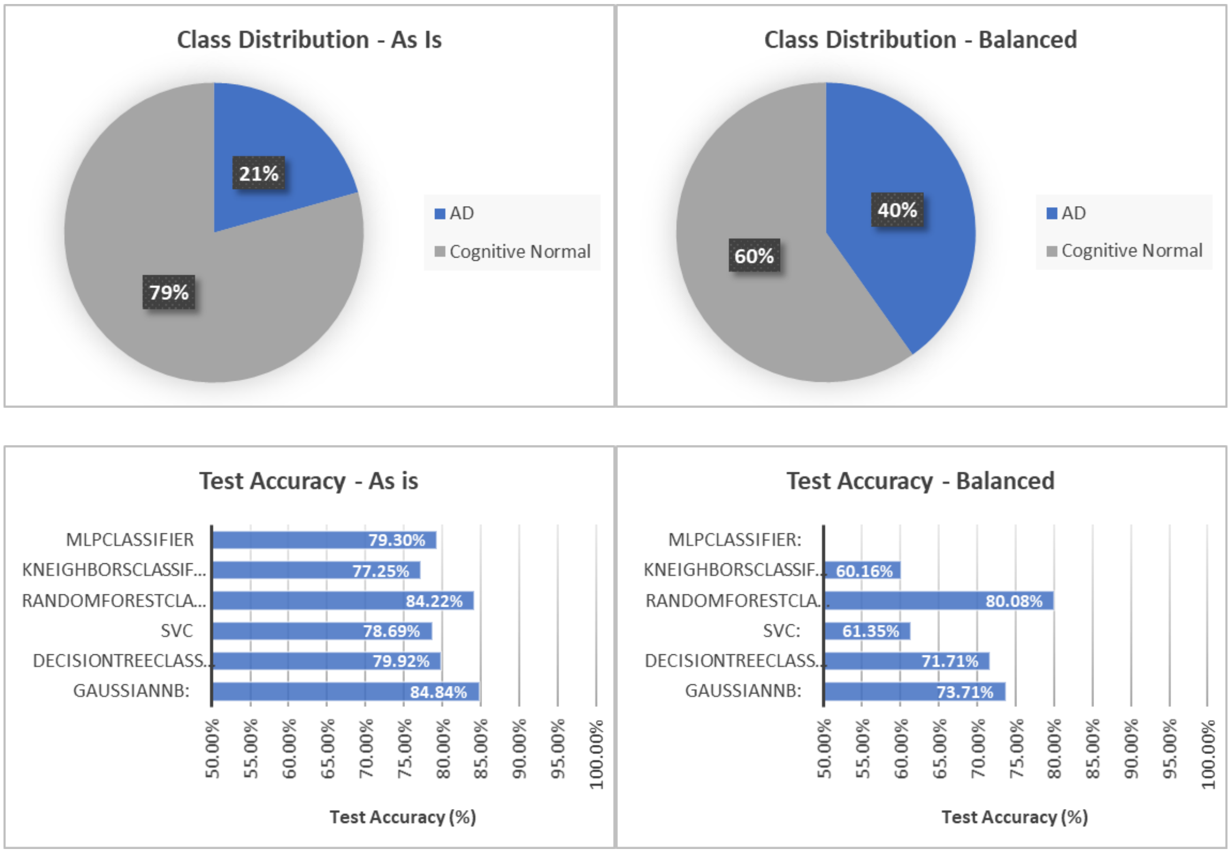


Figure 6.10 Population Distribution and Tess Accuracy data for binary classification using machine learning algorithm

In Figure 6.10, various machine learning algorithm has been performed for binary classification . Also, accuracy performance of "As is" dataset is compared against "Balanced". Based on data, the accuracy data follows population distribution data,. For example when the population distribution was balanced from 79% to 60% for cognitively normal to ensure ratio of conginively normal to Alzhiemer diesease patient is around 1, the accuracies dropped. With "as is" and "balance data", randomforest classification technique performs the best for binary classification.

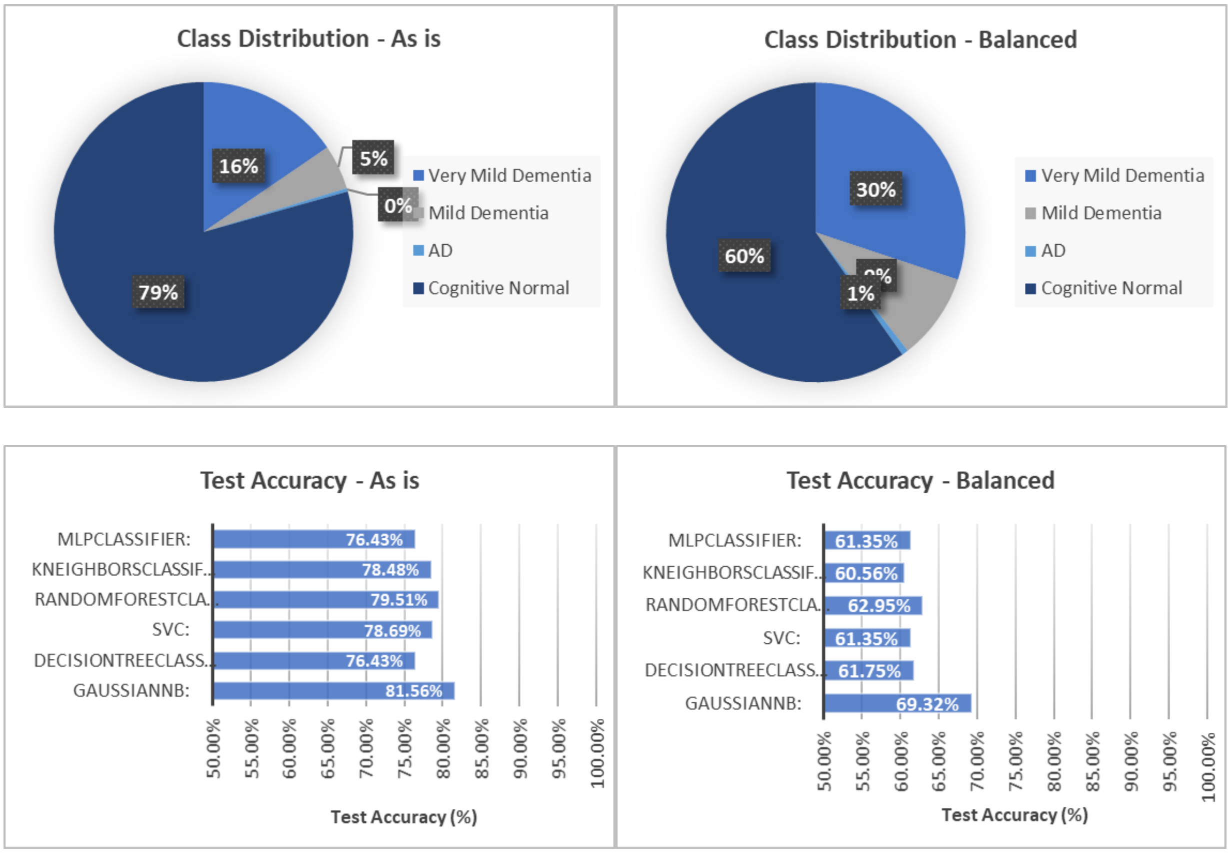


Figure 6.11 Population distribution and test accuracy of various ML algorithms for multiclass classification

In multiclass classification, figure 6.11, similar trend is seen as binary classification – accuracy result is dependent on population distribution. Gaussian naïve bayes technique performs best for multiclass classification.

6.5.2 *Outcomes of Deep Neural Network*

Figure 6.12 clearly represents that concatenation layer (Model 4B and 4M) greatly improves training and validation accuracy and loss for both Binary and Multiclass classification. All of the models are neither overfitting nor underfitting based on accuracy and loss graphs.

Binary Classification Multiclass Classification

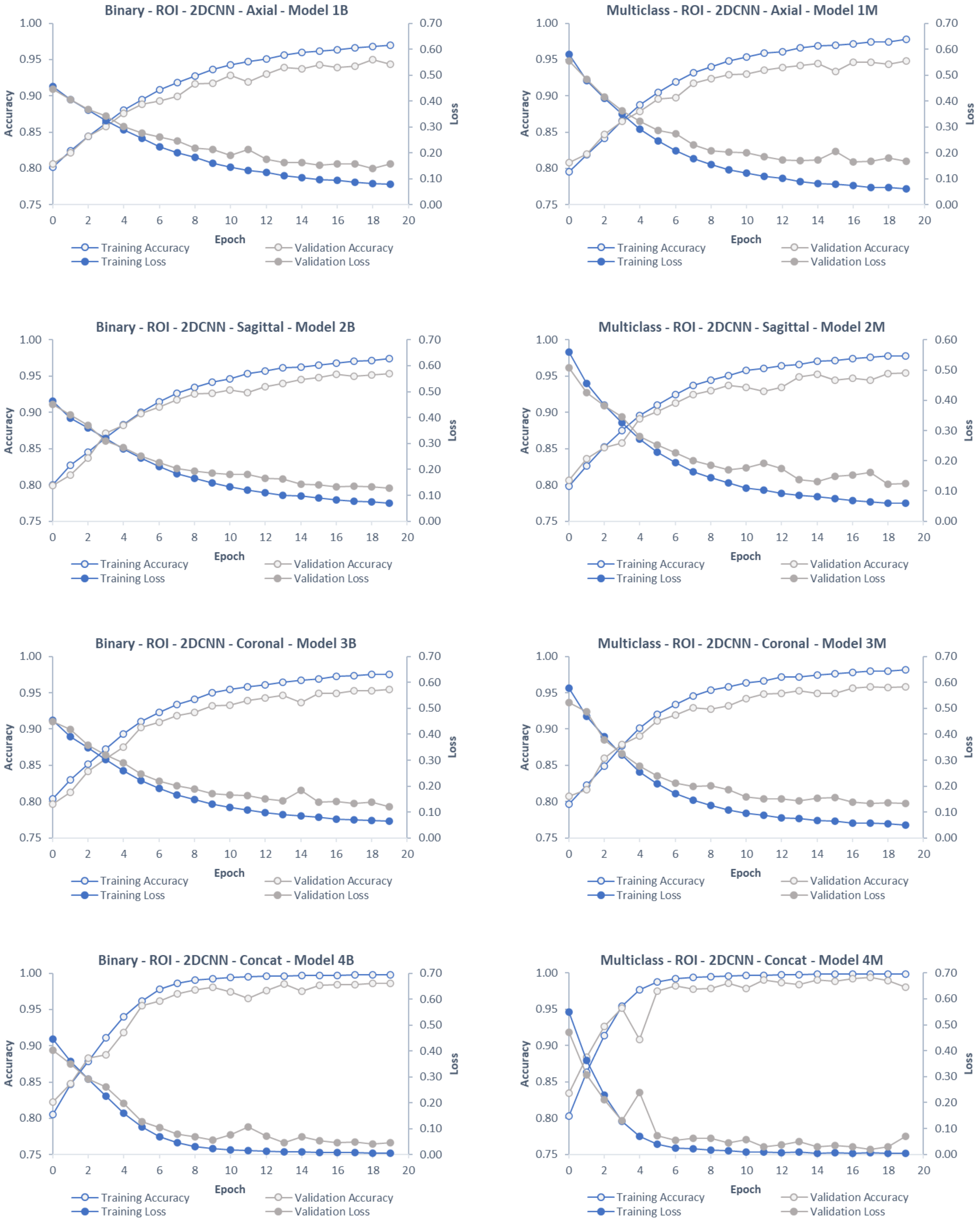


Figure 6.12 Training vs validation accuracy and loss graphs for each model

For test data evaluation, we have used following equations in order to determine the performance of each model:

$$Recall\ or\ Sensitivity = \frac{TP}{TP + FN}$$

$$Specificity = \frac{TN}{TN + FP}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

$$F1\ Score = \frac{2 * TP}{2 * TP + FP + FN}$$

As shown in Table 6.2, binary classification models show that other than sensitivity, precision, F1 score, specificity, and accuracy of Model 4B (which is a concatenation model) perform the best. Whereas, Model 3B performs best for sensitivity, an ability to measure patient with disease correctly. Also, out of three individual plan models (Model 1B, Model 2B, and Model 3B), Model 3B performs the best indicating that for Binary classification, indicating Coronal images are extremely important in identifying a patient with disease correctly.

Table 6.2 Precision, sensitivity, F1 score, specificity, and accuracy data of binary classification deep learning models using test data

	Model 1B	Model 2B	Model 3B	Model 4B
Precision	0.89	0.87	0.86	0.93
Sensitivity	0.84	0.90	0.92	0.86
F1 Score	0.86	0.89	0.89	0.90
Specificity	0.97	0.96	0.96	0.98
Accuracy	0.94	0.95	0.95	0.96

In table 6.3, the data show that Model 2M performs best in terms of sensitivity in order to predict the patient with disease correctly. However, precision, F1_score, accuracy, and specificity of the model 4M is highest compared to other multiclass classification model architectures.

Table 6.3 Precision, sensitivity, F1 score, specificity, and accuracy data of multiclass classification deep learning models using test data

Model	Performance Criteria	AD	MCI	vMCI	CN
Model 1M	Sensitivity	0.91	0.95	0.85	
	Sensitivity (AD + MCI)	0.94			
	Sensitivity (AD + MCI + vMCI)	0.88			
	Precision	1.00	0.93	0.84	
	Precision (AD + MCI)	0.94			
	Precision (AD + MCI + vMCI)	0.87			
	F1_Score	0.95	0.94	0.85	
	F1_Score (AD + MCI)	0.94			
	F1_Score (AD + MCI + vMCI)	0.87			
	Specificity				0.89
	Accuracy	0.95			
Model 2M	Sensitivity	1.00	0.95	0.86	
	Sensitivity (AD + MCI)	0.95			
	Sensitivity (AD + MCI + vMCI)	0.89			
	Precision	1.00	0.91	0.88	
	Precision (AD + MCI)	0.92			
	Precision (AD + MCI + vMCI)	0.89			
	F1_Score	1.00	0.93	0.87	
	F1_Score (AD + MCI)	0.93			
	F1_Score (AD + MCI + vMCI)	0.89			
	Specificity				0.90
	Accuracy	0.96			
Model 3M	Sensitivity	1.00	0.94	0.84	
	Sensitivity (AD + MCI)	0.95			
	Sensitivity (AD + MCI + vMCI)	0.86			
	Precision	1.00	0.87	0.88	
	Precision (AD + MCI)	0.87			
	Precision (AD + MCI + vMCI)	0.88			
	F1_Score	1.00	0.90	0.86	
	F1_Score (AD + MCI)	0.91			
	F1_Score (AD + MCI + vMCI)	0.87			
	Specificity				0.88
	Accuracy	0.95			
Model 4M	Sensitivity	0.96	0.92	0.84	
	Sensitivity (AD + MCI)	0.93			
	Sensitivity (AD + MCI + vMCI)	0.87			
	Precision	1.00	0.95	0.98	
	Precision (AD + MCI)	0.95			
	Precision (AD + MCI + vMCI)	0.97			
	F1_Score	0.98	0.94	0.91	
	F1_Score (AD + MCI)	0.94			

Model	Performance Criteria	AD	MCI	vMCI	CN
	F1_Score (AD + MCI + vMCI)	0.91			
	Specificity				0.88
	Accuracy	0.97			

6.5 Summary

Despite of many attempts to classify Alzheimer’s disease in past, there have not been many studies that attempt multi-class classification based on multi-modal data. Here, in this chapter, a comprehensive comparisons of binary classification vs multi-class classification have been investigated. In real-world data, there are very limited number of true positive data-points, which causes a huge imbalance in classes and requires either class-specific data augmentations or balancing all of the other classes. However, these tasks affect the bias and variance in input data for neural networks. In this chapter though, none of the augmentations were performed, rather other efforts in model architecture were made for generalization. Concatenated model 4M achieved lower sensitivity (true positive rate), compared to individual sagittal model 3M, achieved the highest sensitivity, this could be due to the fact that sagittal plane consists of more sequential hippocampi region. Other clinical data information was utilized in a separate machine learning models, and next step in this study would be to bring all multi-modal data into same embedding space to collectively utilize those to classify the disease.

7 CONCLUSIONS AND FUTURE DIRECTIONS

Multi-modality based deep learning applications are the next big wave in the analysis where utilizing all patient data available to correctly identify the disease detection and prognosis. With the rise in usage of electronic records, at some point all of the patient's data will be easily available to a healthcare professional and it is important to understand how studying patient profile collectively can be very crucial in making a correct judgement call. However, in future more generalized deep neural networks will become an important factor to identify a rare case with real-world imbalanced data.

In this dissertation, several small studies of high impact have been performed in medical image analysis and genomics sequence classification that can serve as a brief template for deep learning applications in this domain. These independent studies can be utilized to create a combinational model if the source of initial datapoints share the same origination, meaning, come from same subject or is from same class with multi-data problems. It is hard to find such real-world dataset publicly, hence models developed in chapter 4, 5 and 6 aids as an example of specific classification tasks corresponding fields of medical image analysis and bioinformatics to be combined in overall pipeline where genomics, imaging and clinical all subject level data is available.

Chapter 4 and 5, developed models can correctly identify 16S rRNA sequences to its classes at family and genus taxa with very high accuracies. At species level, one of the models, BiLSTM, achieved 70.78% for over 2000 species classes. This task remains highly challenged as the dataset does not contain too many sequences to represent individual species. Moreover, species that fall under same genus, tend to share > 97% sequence identity, that causes a lot of room for ambiguity in classification. There are only two other studies for 16S rRNA sequence classification tasks that utilizes deep learning architectures [35, 34], but in this dissertation, the comprehensive comparison of simple recurrent neural networks, convolutional neural networks, hybrid of both recurrent and convolutional networks as well as ensemble networks are investigated instead of just one or two models included in previous studies. This comprehensive study shows recurrent neural networks stabilizes and improves sequence classification accuracies compared to convolutional neural networks. This is the only investigation where recurrent neural networks are utilized for 16S rRNA sequence classification task.

In chapter 6, eight multi-data deep architecture models are developed to classify Alzheimer's disease from MRI slices of all three planes individually and combined to either in normal versus patient with disease, or multiclass classification. Two other models are developed to study the impact of highly correlated subject clinical data that in future can enhance the performance of the deep neural networks. Often times, the classification task is achieved on only single plane or single model, in this chapter, we study the impact of all planes, axial, sagittal and coronal separately and by concatenation. There is a huge potential for an improvement in accuracy with such models, and all of those improvements are further discussed in-depth later in this chapter.

From many experiments performed in this dissertation, one of the biggest learning lessons was to understand which deep learning architectures apply for specific tasks at hand. It is known that convolutional neural networks work best with visual contents such as images and videos, whereas, recurrent neural networks work best with textual data as well as time-series data due to their ability to unroll with respect to time component. Even though 1D CNN can be utilized for string, words or character level embeddings, the performance of this architecture fails to achieve the accuracy compared recurrent neural networks. However, for hybrid models involving convolutional neural networks and at least one layer of recurrent neural network, such as LSTM or BiLSTM, tend to stabilize the overall architecture to almost meet accuracy of recurrent neural networks alone. For image data classification, type of CNN such as 1D, 2D or 3D, depends on the input data as well as available computational power. Even the initial layer of convolutions, the filter size and final number of parameters that can be utilized, heavily depends on available computational power. For example, chapter 6, 3D CNN implementation was not successful with filter numbers of 128 or 64. Other hyperparameter optimization requires some tricks like lowering the learning rate starting from 0.0001 to 0.00001 or higher it to 0.001.

In an overly optimistic long-term goal, in future, there are many other applications that can be viewed separately at first but can be combined to localize, detect, classify and segment a tumor/anomaly/disease in any modality imaging. The goal is to collectively understand all these different applications and increase data utilization to improve architecture's performance and generalization across platforms or diseases.

For, Alzheimer's disease classification chapter 6, there are several direct improvements as below that can:

1) Improve machine learning algorithm by comparing the ratio of cortex total volume to white/gray matter. Use this ratio to compare against CDR scores.
2) Compare the outcomes of different pre-processing techniques, especially ROI vs non-ROI.
3) None of the scans matched with clinical diagnosis of severe dementia CDR score of The final classification included a very small number of sessions that belonged to CDR score of 2, hence, we used a loss function that can serve to normalize the softmax-cross entropy according to the authors that introduced class balanced loss [88]. For class y that has n_y training samples, the class-balanced cross entropy loss is as below:

$$\mathrm{CBsoftmax}(z, y) = \frac{-(1 - \beta)}{(1 - \beta)n_y} \log\left(\frac{\exp(z_y)}{\sum_{j=1}^{C} \exp(z_j)}\right)$$

4) Model architecture to combine clinical data, MRI data, and any other modality data as shown in figure 7.1 below.

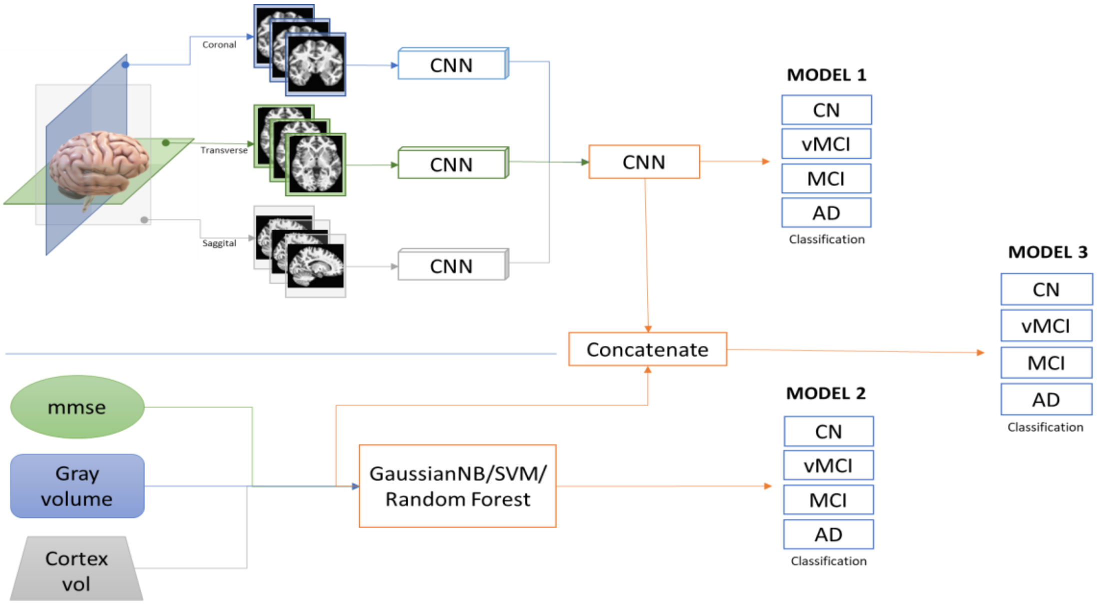


Figure 7.1 Combinational model to improve classification accuracies of Alzheimer's disease for multi-class, multi-data input data

5) Utilize U-net and 3DCNN architectures instead of 2DCNN, although our initial experiments with 3DCNN showed lower accuracies than 2DCNN model architectures. This might be due to limited computational power resources as neural network with only smaller number of filters could be utilized in order for model to even run.

For 16S rRNA gene sequence classification, below are some of the future paths that can be studied further. After studying various different deep learning architectures, it is determined that higher accuracies at species taxa level requires further refinement of

1) Cleaning and pre-processing of 2456 classes in species taxa to ensure at least thirty to forty sequences per class is maintained in species.

2) Using larger than 100 bp length sequences (this is applicable to improve accuracies of other two taxa family and genus as well). Try variable length sequences as an input data as well as dynamic input sequence inputs in recurrent neural networks, that is the input length sequence does not need to be fixed length.

3) Developing a probabilistic model embedded with a deep learning model.

4) An ensemble model for incorporating hierarchical embedded information that currently is being analyzed separately.

ACKNOWLEDGEMENT

In chapter 6 of this dissertation, OASIS-3 dataset was utilized to report our findings, we are thankful for making this dataset available for use. OASIS-3 dataset is supported by following grants. OASIS-3: Principal Investigators: T. Benzinger, D. Marcus, J. Morris; NIH P50AG00561, P30NS09857781, P01AG026276, P01AG003991, R01AG043434, UL1TR000448, R01EB009352. AV-45 doses were provided by Avid Radiopharmaceuticals, a wholly owned subsidiary of Eli Lilly.

8 REFERENCES

[1] A. Esteva, A. Robicquet, B. Ramsundar, V. Kuleshov, M. DePristo, K. Chou, C. Cui, G. Corrado, S. Thrun and J. Dean, "A guide to deep learning in healthcare.," *Nature Medicine,* vol. 25, no. 1, pp. 24-29, 2019.

[2] Y. LeCun, "1.1 Deep Learning Hardware: Past, Present, and Future," in *2019 IEEE International Solid- State Circuits Conference - (ISSCC)*, 2019.

[3] Y. LeCun, Y. Bengio and G. Hinton, "Deep learning," *Nature,* vol. 521, no. 7553, pp. 436-444, 2015.

[4] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein, A. C. Berg and L. Fei-Fei, "ImageNet Large Scale Visual Recognition Challenge," *International Journal of Computer Vision,* vol. 115, no. 3, pp. 211-252, 2015.

[5] J. Hirschberg and C. D. Manning, "Advances in natural language processing.," *Science,* vol. 349, no. 6245, pp. 261-266, 2015.

[6] G. J. S. Litjens, T. Kooi, B. E. Bejnordi, A. A. A. Setio, F. Ciompi, M. Ghafoorian, J. A. W. M. v. d. Laak, B. v. Ginneken and C. I. Sánchez, "A survey on deep learning in medical image analysis," *Medical Image Analysis,* vol. 42, pp. 60-88, 2017.

[7] L. Shen, L. R. Margolies, J. H. Rothstein, E. Fluder, R. McBride and W. Sieh, "Deep Learning to Improve Breast Cancer Detection on Screening Mammography.," *Scientific Reports,* vol. 9, no. 1, p. 12495, 2019.

[8] S. Jain, V. jagtap and N. Pise, "Computer Aided Melanoma Skin Cancer Detection Using Image Processing," *Procedia Computer Science,* vol. 48, pp. 735-740, 2015.

[9] A. Imran, J. Li, Y. Pei, J.-J. Yang and Q. Wang, "Comparative Analysis of Vessel Segmentation Techniques in Retinal Images," *IEEE Access,* vol. 7, pp. 114862-114887, 2019.

[10] D. A. Sippo, R. L. Birdwell, K. P. Andriole and S. Raza, "Quality Improvement of Breast MRI Reports With Standardized Templates for Structured Reporting," *Journal of The American College of Radiology,* vol. 14, no. 4, pp. 517-520, 2017.

[11] S. K. Goergen, F. J. Pool, T. J. Turner, J. E. Grimm, M. N. Appleyard, C. Crock, M. C. Fahey, M. F. Fay, N. J. Ferris, S. M. Liew, R. D. Perry, A. Revell, G. M. Russell, S.-c. S. Wang and C. Wriedt, "Evidence-based guideline for the written radiology report: Methods, recommendations and implementation challenges," *Journal of Medical Imaging and Radiation Oncology,* vol. 57, no. 1, pp. 1-7, 2013.

[12] H. P. Desai, P. P. Anuja, M. Weeks and S. Rajshekhar, "16S Ribosomal Gene Classification Using Recurrent Neural Network Models," in *15th International Symposium on Bioinformatics Research and Applications (ISBRA),* Barcelona, 2019.

[13] H. P. Desai, A. P. Parameshwaran, R. Sunderraman and M. Weeks, "Comparative Study Using Neural Networks for 16S Ribosomal Gene Classification," *Journal of Computational Biology,* vol. 27, no. 2, pp. 248-258, 2020.

[14] H. P. Desai, P. P. Anuja, S. Rajshekhar and M. Weeks, "Deep Ensemble Models for 16S Ribosomal Gene Classification," in *ISBRA 2020, Bioinformatics Research and Applications,* vol. 12304, Lecture Notes in Bioinformatics, pp. 1-9.

[15] J. Kubilius, "Ventral visual stream. figshare. Figure.," 2017. [Online]. Available: https://doi.org/10.6084/m9.figshare.106794.v3.

[16] D. H. Hubel and T. N. Wiesel, "Receptive fields, binocular interaction and functional architecture in the cat's visual cortex," *The Journal of Physiology,* vol. 160, no. 1, pp. 106-154, 1962.

[17] M. Eickenberg, A. Gramfort, G. Varoquaux and B. Thirion, "Seeing it all: Convolutional network layers map the function of the human visual system," *NeuroImage,* vol. 152, pp. 184-194, 2017.

[18] K. Fukushima, "Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position," *Biological Cybernetics,* vol. 36, no. 4, pp. 193-202, 1980.

[19] . Hinton, J. Blanks, H. Fong, P. Casey, E. Hildebrandt and M. Simons, "Novel localization of a G protein, Gz-alpha, in neurons of brain and retina," *The Journal of Neuroscience,* vol. 10, no. 8, pp. 2763-2770, 1990.

[20] Y. Lecun, "Gradient-based learning applied to document recognition," *Intelligent Signal Processing,* pp. 306-351, 2001.

[21] A. Krizhevsky, I. Sutskever and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *neural information processing systems,* vol. 141, no. 5, pp. 1097-1105, 2012.

[22] M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, M. Hasan, B. C. V. Esesn, A. A. S. Awwal and V. K. Asari, "The History Began from AlexNet: A Comprehensive Survey on Deep Learning Approaches.," *arXiv preprint arXiv:1803.01164,* 2018.

[23] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. E. Reed, D. Anguelov, D. Erhan, V. Vanhoucke and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2015.

[24] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," *international conference on learning representations,* 2015.

[25] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),* 2016.

[26] A. Gibson and J. Patterson, Deep Learning: A Practitioner's Approach, 2017.

[27] D. Britz, "Recurrent Neural Networks Tutorial, Part 1–Introduction to RNNs.," 23 12 2017. [Online].

[28] C. Olah, "Understanding lstm networks," 2015. [Online].

[29] J. Zou, M. Huss, A. Abid, P. Mohammadi, A. Torkamani and A. Telenti, "A primer on deep learning in genomics.," *Nature Genetics,* vol. 51, no. 1, pp. 12-18, 2019.

[30] H. M. Afify and M. A. Al-Masni, "Taxonomy metagenomic analysis for microbial sequences in three domains system via machine learning approaches," *Informatics in Medicine Unlocked,* vol. 13, pp. 151-157, 2018.

[31] N. Chaudhary, A. K. Sharma, P. Agarwal, A. Gupta and V. K. Sharma, "16S classifier: a tool for fast and accurate taxonomic classification of 16S rRNA hypervariable regions in metagenomic datasets.," *PLOS ONE,* vol. 10, no. 2, 2015.

[32] H. Vinje, K. H. Liland, T. Almøy and L.-G. Snipen, "Comparing K-mer based methods for improved classification of 16S sequences," *BMC Bioinformatics,* vol. 16, no. 1, pp. 205-205, 2015.

[33] D. Fioravanti, Y. Giarratano, V. Maggio, C. Agostinelli, M. Chierici, G. Jurman and C. Furlanello, "Phylogenetic convolutional neural networks in metagenomics," *BMC Bioinformatics,* vol. 19, no. 2, pp. 49-49, 2018.

[34] A. Fiannaca, L. L. Paglia, M. L. Rosa, G. L. Bosco, G. Renda, R. Rizzo, S. Gaglio and A. Urso, "Deep learning models for bacteria taxonomic classification of metagenomic data," *BMC Bioinformatics,* vol. 19, no. 7, p. 198, 2018.

[35] A. Busia, G. Dahl, C. Fannjiang, D. Alexander, L. Dorfman, R. Poplin, C. McLean, P.-C. Chang and M. DePristo, "A deep learning approach to pattern recognition for short DNA sequences," *bioRxiv,* p. 353474, 2018.

[36] C. Yuan, J. Lei, J. R. Cole and Y. Sun, "Reconstructing 16S rRNA genes in metagenomic data.," *Bioinformatics,* vol. 31, no. 12, pp. 35-43, 2015.

[37] M. Ramazzotti, L. Berná, C. Donati and D. Cavalieri, "riboFrame: An Improved Method for Microbial Taxonomy Profiling from Non-Targeted Metagenomics," *Frontiers in Genetics,* vol. 6, pp. 329-329, 2015.

[38] Q. Wang, G. M. Garrity, J. M. Tiedje and J. R. Cole, "Naïve Bayesian classifier for rapid assignment of rRNA sequences into the new bacterial taxonomy," *Applied and Environmental Microbiology,* vol. 73, no. 16, pp. 5261-5267, 2007.

[39] F. E. Angly, D. Willner, F. Rohwer, P. Hugenholtz and G. W. Tyson, "Grinder: a versatile amplicon and shotgun sequence simulator," *Nucleic Acids Research,* vol. 40, no. 12, 2012.

[40] R. D’Amore, U. Z. Ijaz, M. Schirmer, J. G. Kenny, R. Gregory, A. C. Darby, M. Shakya, M. Podar, C. Quince and N. Hall, "A comprehensive benchmarking study of protocols

and sequencing platforms for 16S rRNA community profiling," *BMC Genomics*, vol. 17, no. 1, pp. 55-55, 2016.

[41] D. A. W. Soergel, N. Dey, R. Knight and S. E. Brenner, "Selection of primers for optimal taxonomic classification of environmental 16S rRNA gene sequences," *The ISME Journal*, vol. 6, no. 7, pp. 1440-1444, 2012.

[42] W. Zheng, M. Tsompana, A. Ruscitto, A. Sharma, R. Genco, Y. Sun and M. J. Buck, "An accurate and efficient experimental approach for characterization of the complex oral microbiota," *Microbiome*, vol. 3, no. 1, pp. 48-48, 2015.

[43] S. Chakravorty, D. Helb, M. Burday, N. Connell and D. Alland, "A detailed analysis of 16S ribosomal RNA gene segments for the diagnosis of pathogenic bacteria.," *Journal of Microbiological Methods*, vol. 69, no. 2, pp. 330-339, 2007.

[44] S. Federhen, "The NCBI Taxonomy database.," *Nucleic Acids Research*, vol. 40, pp. 136-143, 2012.

[45] M. C. Nelson, H. G. Morrison, J. Benjamino, S. L. Grim and J. Graf, "Analysis, Optimization and Verification of Illumina-Generated 16S rRNA Gene Amplicon Surveys," *PLOS ONE*, vol. 9, no. 4, 2014.

[46] M. Schirmer, U. Z. Ijaz, R. D'Amore, N. Hall, W. Sloan and C. Quince, "Insight into biases and sequencing errors for amplicon sequencing with the Illumina MiSeq platform," *Nucleic Acids Research*, vol. 43, no. 6, 2015.

[47] L. Sifre and S. Mallat, "Rotation, Scaling and Deformation Invariant Scattering for Texture Discrimination," in *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[48] S. Basaia, F. Agosta, L. Wagner, E. Canu, G. Magnani, R. Santangelo and M. Filippi, "Automated classification of Alzheimer's disease and mild cognitive impairment using a single MRI and deep neural networks.," *NeuroImage: Clinical*, vol. 21, p. 101645, 2019.

[49] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu and X. Zheng, "TensorFlow: a system for large-scale machine learning," in *OSDI'16 Proceedings of the 12th USENIX conference on Operating Systems Design and Implementation*, 2016.

[50] F. Altaf, S. M. S. Islam, N. Akhtar and N. K. Janjua, "Going Deep in Medical Image Analysis: Concepts, Methods, Challenges, and Future Directions," *IEEE Access,* vol. 7, pp. 99540-99572, 2019.

[51] J. Fan, X. Cao, P. T. Yap and D. Shen, "BIRNet: Brain image registration using dual-supervised fully convolutional networks.," *Medical Image Analysis,* vol. 54, pp. 193-206, 2019.

[52] M. Achtman, "A Phylogenetic Perspective on Molecular Epidemiology," *Molecular Medical Microbiology,* pp. 485-509, 2002.

[53] P. Woo, S. Lau, J. Teng, H. Tse and K.-Y. Yuen, "Then and now: use of 16S rDNA gene sequencing for bacterial identification and discovery of novel bacteria in clinical microbiology laboratories," *Clinical Microbiology and Infection,* vol. 14, no. 10, pp. 908-934, 2008.

[54] P. D. Schloss, S. L. Westcott, T. Ryabin, J. R. Hall, M. Hartmann, E. B. Hollister, R. A. Lesniewski, B. B. Oakley, D. H. Parks, C. J. Robinson, J. W. Sahl, B. Stres, G. G.

Thallinger, D. J. V. Horn and C. F. Weber, "Introducing mothur: Open-Source, Platform-Independent, Community-Supported Software for Describing and Comparing Microbial Communities," *Applied and Environmental Microbiology,* vol. 75, no. 23, pp. 7537-7541, 2009.

[55] J. G. Caporaso, J. Kuczynski, J. Stombaugh, K. Bittinger, F. D. Bushman, E. K. Costello, N. Fierer, A. G. Peña, J. K. Goodrich, J. I. Gordon, G. A. Huttley, S. T. Kelley, D. Knights, J. E. Koenig, R. E. Ley, C. A. Lozupone, D. McDonald, B. D. Muegge, M. Pirrung, J. Reeder, J. R. Sevinsky, P. J. Turnbaugh, W. A. Walters, J. Widmann, T. Yatsunenko, J. Zaneveld and R. Knight, "QIIME allows analysis of high-throughput community sequencing data," *Nature Methods,* vol. 7, no. 5, pp. 335-336, 2010.

[56] M. L. Rosa, A. Fiannaca, R. Rizzo and A. Urso, "Probabilistic topic modeling for the analysis and classification of genomic sequences," *BMC Bioinformatics,* vol. 16, no. 6, pp. 1-9, 2015.

[57] A.-b. Zhang, D. S. Sikes, C. Muster and S. Q. Li, "Inferring Species Membership Using DNA Sequences with Back-Propagation Neural Networks," *Systematic Biology,* vol. 57, no. 2, pp. 202-215, 2008.

[58] Y. Park and M. Kellis, "Deep learning for regulatory genomics," *Nature Biotechnology,* vol. 33, no. 8, pp. 825-826, 2015.

[59] B. Alipanahi, A. Delong, M. T. Weirauch and B. J. Frey, "Predicting the sequence specificities of DNA- and RNA-binding proteins by deep learning," *Nature Biotechnology,* vol. 33, no. 8, pp. 831-838, 2015.

[60] Y. Kim, "Convolutional Neural Networks for Sentence Classification," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014.

[61] R. Johnson and T. Zhang, "Effective Use of Word Order for Text Categorization with Convolutional Neural Networks," in *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2015*, 2015.

[62] Y. Zhang and B. C. Wallace, "A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification," in *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2015.

[63] T. H. Nguyen and R. Grishman, "Relation Extraction: Perspective from Convolutional Neural Networks," in *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 2015.

[64] Y. Sun, L. Lin, D. Tang, N. Yang, Z. Ji and X. Wang, "Modeling mention, context and entity with neural networks for entity disambiguation," in *IJCAI'15 Proceedings of the 24th International Conference on Artificial Intelligence*, 2015.

[65] D. Zeng, K. Liu, S. Lai, G. Zhou and J. Zhao, "Relation Classification via Convolutional Deep Neural Network," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.

[66] J. Gao, P. Pantel, M. Gamon, X. He and L. Deng, *Modeling Interestingness with Deep Neural Networks*, 2014, pp. 2-13.

[67] Y. Shen, X. He, J. Gao, L. Deng and G. Mesnil, "A Latent Semantic Model with Convolutional-Pooling Structure for Information Retrieval," in *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, 2014.

[68] "GRD - Genomic-based 16S ribosomal RNA Database," RIKEN, Laboratory for Integrated Bioinformatics, Center for Integrative Medical Sciences, 2015.

[69] W. Yin, K. Kann, M. Yu and H. Schütze, "Comparative Study of CNN and RNN for Natural Language Processing," *arXiv preprint arXiv:1702.01923,* 2017.

[70] A. Ghosh, M. Aditya and K. Asif, "Metagenomic Analysis and its Applications," *Encylopedia Bioinformatics Computational Biology,* vol. 3, pp. 184-193, 2019.

[71] J. Qin, Y. Li, Z. Cai, S. Li, J. Zhu, F. Zhang, S. Liang, W. Zhang, Y. Guan, D. Shen, Y. Peng, D. Zhang, Z. Jie, W. Wu, Y. Qin, W. Xue, J. Li, L. Han, D. Lu, P. Wu, Y. Dai, X. Sun, Z. Li, A. Tang, S. Zhong, X. Li, W. Chen, R. Xu, M. Wang, Q. Feng, M. Gong, J. Yu, Y. Zhang, M. Zhang, T. Hansen, G. Sanchez, J. Raes, G. Falony, S. Okuda, M. Almeida, E. LeChatelier, P. Renault, N. Pons, J.-M. Batto, Z. Zhang, H. Chen, R. Yang, W. Zheng, S. Li, H. Yang, J. Wang, S. D. Ehrlich, R. Nielsen, O. Pedersen, K. Kristiansen and J. Wang, "A metagenome-wide association study of gut microbiota in type 2 diabetes," *Nature,* vol. 490, no. 7418, pp. 55-60, 2012.

[72] P. J. Turnbaugh, R. E. Ley, M. A. Mahowald, V. Magrini, E. R. Mardis and J. I. Gordon, "An obesity-associated gut microbiome with increased capacity for energy harvest," *Nature,* vol. 444, no. 7122, pp. 1027-1031, 2006.

[73] P. J. Turnbaugh, M. Hamady, T. Yatsunenko, B. L. Cantarel, A. Duncan, R. E. Ley, M. L. Sogin, W. J. Jones, B. A. Roe, J. P. Affourtit, M. Egholm, B. Henrissat, A. C. Heath, R. Knight and J. I. Gordon, "A core gut microbiome in obese and lean twins," *Nature*, vol. 457, no. 7228, pp. 480-484, 2009.

[74] F. H. Karlsson, F. Fåk, I. Nookaew, V. Tremaroli, B. Fagerberg, D. Petranovic, F. Bäckhed and J. B. Nielsen, "Symptomatic atherosclerosis is associated with an altered gut metagenome," *Nature Communications*, vol. 3, no. 1, pp. 1245-1245, 2012.

[75] J. M. Janda and S. L. Abbott, "16S rRNA gene sequencing for bacterial identification in the diagnostic laboratory: pluses, perils, and pitfalls.," *Journal of Clinical Microbiology*, vol. 45, no. 9, pp. 2761-2764, 2007.

[76] J. M. Berg, J. L. Tymoczko and L. Stryer, "Biochemistry 5th ed," , 2002.

[77] C. R. Woese, "Bacterial evolution," *Microbiological Reviews*, vol. 51, no. 2, p. 221, 1987.

[78] D. E. Wood and S. L. Salzberg, "Kraken: ultrafast metagenomic sequence classification using exact alignments," *Genome Biology*, vol. 15, no. 3, pp. 1-12, 2014.

[79] L. Rokach, "Ensemble-based classifiers," *Artificial Intelligence Review*, vol. 33, no. 1, pp. 1-39, 2010.

[80] R. Prabhavalkar, K. Rao, T. N. Sainath, B. Li, L. Johnson and N. Jaitly, "A Comparison of Sequence-to-Sequence Models for Speech Recognition," in *Interspeech 2017*, 2017.

[81] S. Venugopalan, M. Rohrbach, J. Donahue, R. Mooney, T. Darrell and K. Saenko, "Sequence to Sequence -- Video to Text," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[82] C. R. Jack, M. A. Bernstein, N. C. Fox, P. Thompson, G. Alexander, D. Harvey, B. Borowski, P. J. Britson, J. L. Whitwell, C. Ward, A. M. Dale, J. P. Felmlee, J. L. Gunter, D. L. G. Hill, R. Killiany, N. Schuff, S. Fox-Bosetti, C. Lin, C. Studholme, C. S. DeCarli, G. Krueger, H. A. Ward, G. J. Metzger, K. T. Scott, R. Mallozzi, D. Blezek, J. Levy, J. P. Debbins, A. S. Fleisher, M. Albert, R. Green, G. Bartzokis, G. Glover, J. Mugler and M. W. Weiner, "The Alzheimer's Disease Neuroimaging Initiative (ADNI): MRI methods.," *Journal of Magnetic Resonance Imaging,* vol. 27, no. 4, pp. 685-691, 2008.

[83] P. J. LaMontagne, T. L. Benzinger, J. C. Morris, S. Keefe, R. Hornbeck, C. Xiong, E. Grant, J. Hassenstab, K. Moulder, A. Vlassenko, M. E. Raichle, C. Cruchaga and D. Marcus, "OASIS-3: Longitudinal Neuroimaging, Clinical, and Cognitive Dataset for Normal Aging and Alzheimer Disease," *medRxiv,* 2019.

[84] K. A. Ellis, A. I. Bush, D. Darby, D. D. Fazio, J. Foster, P. Hudson, N. T. Lautenschlager, N. Lenzo, R. N. Martins, P. Maruff, C. Masters, A. Milner, K. Pike, C. Rowe, G. Savage, C. Szoeke, K. Taddei, V. Villemagne, M. Woodward and D. Ames, "The Australian Imaging, Biomarkers and Lifestyle (AIBL) study of aging: Methodology and baseline characteristics of 1112 individuals recruited for a longitudinal study of Alzheimer's disease," *International Psychogeriatrics,* vol. 21, no. 4, pp. 672-687, 2009.

[85] f. A. D. N. Initiative, "Predicting Alzheimer’s disease progression using multi-modal deep learning approach," *Scientific Reports,* vol. 9, no. 1, p. 1952, 2019.

[86] L. Lazli, M. Boukadoum and O. A. Mohamed, "A Survey on Computer-Aided Diagnosis of Brain Disorders through MRI Based on Machine Learning and Data Mining

Methodologies with an Emphasis on Alzheimer Disease Diagnosis and the Contribution of the Multimodal Fusion," *Applied Sciences,* vol. 10, no. 5, p. 1894, 2020.

[87] W. Henneman, J. Sluimer, J. Barnes, W. v. d. Flier, I. Sluimer, N. Fox, P. Scheltens, H. Vrenken and F. Barkhof, "Hippocampal atrophy rates in Alzheimer disease Added value over whole brain volume measures," *Neurology,* vol. 72, no. 11, pp. 999-1007, 2009.

[88] Y. Cui, M. Jia, T.-Y. Lin, Y. Song and S. Belongie, "Class-Balanced Loss Based on Effective Number of Samples," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[89] A. S. Lundervold and A. Lundervold, "An overview of deep learning in medical imaging focusing on MRI," *Zeitschrift Fur Medizinische Physik,* vol. 29, no. 2, pp. 102-127, 2019.

[90] X. Wang, I. K. Jordan and L. W. Mayer, "A Phylogenetic Perspective on Molecular Epidemiology," *Molecular Medical Microbiology (Second Edition)*, pp. 517-536, 2015.