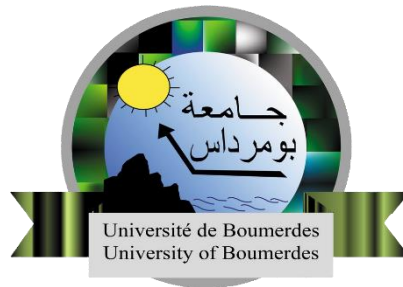


People's Democratic Republic of Algeria
Ministry of Higher Education and Scientific Research
University M'Hamed BOUGARA- Boumerdes



Institute of Electrical and Electronic Engineering

Department of Electronics

Final Year Project Report Presented in Partial Fulfillment of The requirement
for the Degree of

Master

In Electrical and electronic engineering

Option: **Computer engineering**

Title:

EMG Signals Classification for Neuromuscular Diseases
Detection Using Deep Learning

Presented by:

Loubar Lidia

Toubal Maria

Supervisor:

Co-supervisor:

Dr. Elhocine BOUTELLAA

Dr. Rachid NAMANE

Registration Number: / 2023

Abstract

Neuromuscular diseases are particular impairments that affect the muscle tissue or nervous system part connected to muscles. Electromyography (EMG) signals are valuable biosignals for the diagnosis of neuromuscular diseases. However, the classification of EMG signals is a challenging task due to the complexity of the signals and the variability of the diseases. In this project, we address the problem of EMG signals classification for the detection of neuromuscular diseases using deep learning techniques. The main goal of our project is to develop a robust deep-learning model that performs well on unseen data, thereby improving the reliability of diagnosis in real-life scenarios. To achieve this, we design a model which we train and evaluate on a dataset of EMG signals from patients with different neuromuscular diseases.

We assess the performance of our designed model using two different methods : the train-test split approach, commonly employed in the existing literature, and the subject-independent evaluation method, which ensures that the model is tested on completely unseen data.

The results show that the model achieves excellent performance on the train-test split approach. However, the second method produces varied and uneven scores for different patients, suggesting that EMG data of certain individuals may be more challenging to classify accurately. Nonetheless, some patients exhibit highly accurate classifications, demonstrating the potential performance of our designed model. The obtained results indicate the potential of the developed tool for the diagnosis of neuromuscular diseases.

Dedication

Toubal Maria

I would like to dedicate this project to the people who have played a key role in my path. Their help, guidance, and support have been essential to my success. I am truly thankful for their presence in my life.

To my dearest parents, I dedicate this work to you. You have always been my biggest supporter, and I could not have done this without your love and guidance. Thank you for always believing in me, even when I didn't believe in myself.

To my sister Kamelia, you are my best friend and my biggest inspiration. Thank you for always being there, no matter what.

To my friends, I am so grateful for your friendship and support throughout this project. You have made this journey so much more enjoyable. Thank you for always being there to lend a helping hand and for always being willing to share your knowledge.

To my colleagues at Microtech Lab, I am especially grateful for the way you welcomed me into the warm Microtech family. You have all been so generous with your time and your knowledge. You have enlightened me with your experience and helped me to grow as a professional.

To my partner Loubar Lidia, I am so proud of what we have accomplished together. This project is a testament to our hard work and dedication. Thank you for making this possible.

To the people whom I love, and to the people who love me, thank you for being in my life. Your love, support, and friendship mean the world to me. I am so grateful for all that you do. I love you all very much.

Loubar Lidia

I would like to dedicate this project to the special individuals who have played an invaluable role in my journey.

To my parents, whose unwavering support and encouragement have been the guiding light throughout my journey. Your belief in me and constant encouragement have been instrumental in my pursuit of knowledge and personal growth.

To my siblings, you have been my cheerleaders and confidants. Your presence and encouragement have been a constant reminder that I am never alone on this path. Thank you for always being there for me.

To my teachers, your guidance and wisdom have shaped my mind and ignited my curiosity. Your dedication to imparting knowledge has made a profound impact on my educational journey.

To my friends, you have been my pillars of strength, offering support and understanding throughout this project. Your words of encouragement and shared experiences have made the challenges more bearable and the victories more joyful.

And finally, to my project partner Toubal Maria and my best friend, thank you for being by my side. Your unwavering support and expertise have always been valuable. This project would not have been the same without you.

Thank you from the bottom of my heart.

Acknowledgment

We would like to express our sincere gratitude to our supervisors, Dr. Elhocine BOUTELLAA and Dr. Rachid NAMANE, for their guidance, support, and encouragement throughout this project. Their expertise and insights were invaluable to us, and we are grateful for their willingness to share their knowledge with us.

We would also like to thank Mr. K. M. Naimul Hassan, the author of the paper "ALSNet," for answering our questions about his paper. His insights were helpful in understanding the state-of-the-art in EMG signal classification, and we are grateful for his willingness to help us.

Finally, we would like to thank Mr. Ould-Oulhadj Nacer, a patient of myopathy, for sharing with us the process followed by his doctors to diagnose him with the illness. His story was both inspiring and heartbreaking, and it helped us to understand the importance of our work.

We are grateful to all of the people who helped us in the accomplishment of this project. Their support and guidance made this project possible.

Table of Content

Abstract.....	2
Dedication	3
Acknowledgment.....	5
Table of Content.....	6
List of Figures.....	9
List of Tables	11
List of Abbreviations	12
General Introduction.....	13
Chapter 1: Electromyography Concepts.....	15
1.1 Introduction.....	16
1.2 Generalities on the Muscles	16
1.2.1 Definition and Types of Muscles.....	16
a. Skeletal muscle.....	17
b. Smooth muscle.....	17
c. Cardiac muscle	17
1.2.2 Skeletal muscles.....	18
1.2.3 Motor Unit	18
1.2.4 Neuromuscular Junction	19
1.3 Electromyogram.....	19
1.3.1 EMG Generation in the Muscle	19
1.3.2 EMG Signal Acquisition.....	19
1.2.3 Factors Affecting EMG Signals.....	20
1.2.4 Clinical Applications of EMG	21
1.4 Neuromuscular Disorders	21
a. Myopathy	21
b. Neuropathy.....	22
1.5 Conclusion	22
Chapter 2: Deep Learning and Related Works.....	23
2.1 Introduction.....	24
2.2 Deep Learning.....	24
2.3 Artificial Neural Networks	26
2.4 Layers in a Neural Network.....	26
2.4.1 Convolutional layers	26
2.4.2 Pooling layers.....	28
2.4.3 Fully connected layers	28
2.4.4 Recurrent layers	29
2.4.5 Batch normalization layer	29
2.4.6 Dropout layer	30

2.5 Activation functions.....	30
2.6 Training a neural network.....	33
2.6.1 Backpropagation	34
2.6.2 Optimiser/gradient descent	34
2.6.3 Loss function.....	35
2.6.4 Epoch, batch, and iteration.....	36
2.6.5 Train, validation, and test subsets	36
2.7 Performance Metrics	36
2.7.1 Confusion Metrics.....	37
2.7.2 Accuracy	37
2.7.3 Precision.....	37
2.7.4 Recall	37
2.7.5 F1 score	38
2.8 Related works.....	38
2.8.1 DeepEMGNet: An Application for Efficient Discrimination of ALS and Normal EMG Signals	38
2.8.2 Electromyography (EMG) based Classification of Neuromuscular Disorders using Multi-Layer Perceptron	39
2.8.3 EMG Signal Classification for Detecting Neuromuscular Disorders	40
2.8.4 Deep learning-based diagnosis of myopathy and neuropathy	40
2.8.5 Analysis and Classification of Muscular Paralysis Disease using Electromyography Signal with Machine Learning	41
2.8.6 ALSNet: A Dilated 1-D CNN For Identifying ALS Fom Raw EMG Signal	41
2.9 Conclusion	43
Chapter 3: Implementation and Evaluation of the Classification System	45
3.1 Introduction.....	46
3.2 Implementation	46
3.2.1 Tools and Technologies	46
Python	46
Jupyter Notebook	46
Kaggle	47
Google Collaboratory.....	47
3.2.2 Libraries	47
Keras	47
NumPy	47
Pandas	48
Waveform-database (WFDB)	48
Matplotlib.....	48
Scikit-learn.....	48
3.3 Data and Preprocessing.....	48
3.3.1 Description of the data	49

3.3.2 Data Preparation.....	50
a. Muscle selection:.....	50
b. Windowing and removing missing values:	51
c. Balancing the data:	51
3.4 Proposed Network Architecture.....	53
3.4.1 CNN Model.....	54
3.4.2 Training Methodology	56
3.5 Results and Discussion	57
3.5.1 Train-Validation-Test Split Approach.	57
3.5.1.1 Multiclass Classification.....	57
3.5.1.2 Binary Classification.....	60
3.5.2 Subject-Independent Evaluation Approach	65
3.5.2.1 Multiclass Classification.....	65
3.5.2.2 Binary Classification-Myopathy vs. Normal	69
3.5.2.3 Binary Classification-ALS vs. Normal	73
3.6 Comparison	77
3.7 Conclusion	79
General Conclusion.....	79
References.....	81
Appendix A.....	86
A.1 LSTM Model.....	86
A.2 Results and Discussion.....	87

List of Figures

Figure 1.1:	Types of muscles.	3
Figure 1.2:	Structure of The Motor Unit	4
Figure 1.3:	(a) Surface EMG. (b) Intramuscular EMG.	6
Figure 2.1:	Relationship between deep learning, machine learning, and artificial intelligence.	11
Figure 2.2:	Effect of training data size on the performance of DL vs. conventional ML.	11
Figure 2.3:	Example of the convolution process in a convolutional layer.	14
Figure 3.1:	EMG signals of (a) Normal (b) Myopathy and (c) ALS subjects.	36
Figure 3.2:	1D CNN model for multiclass classification.	41
Figure 3.3:	1D CNN model for binary classification.	42
Figure 3.4:	Multiclass Classification Train and Validation Plots. (a) Loss Plots.(b) Accuracy plots.	44
Figure 3.5:	Multiclass Classification Confusion Matrix.	44
Figure 3.6:	Train and Validation Plots for Binary Classification (Myopathy vs. Normal). (a) Loss Plots. (b)Accuracy Plots.	46
Figure 3.7:	Confusion Matrix of the CNN Model for Binary Classification (Myopathy vs. Normal) using Train-Test Split Approach.	47
Figure 3.8:	Binary Classification (ALS vs. Normal) Train and Validation Plots. (a) Loss Plots. (b)Accuracy Plots.	48
Figure 3.9:	Confusion Matrix for Binary Classification (ALS vs. Normal) using Train-Test Split Approach.	49
Figure 3.10:	Train and Validation Plots for Multiclass Classification Subject Independent Evaluation Method -worst result-. (a) Loss Plots. (b)Accuracy Plots.	52
Figure 3.11:	Confusion Matrix for Multiclass Classification Subject Independent Evaluation Approach -worst result-.	52
Figure 3.12:	Train and Validation Plots for Multiclass Classification Subject Independent Evaluation Method -best result-. (a) Loss Plots. (b)Accuracy Plots.	53
Figure 3.13:	Confusion Matrix for Multiclass Classification Subject Independent Evaluation Method -best result-.	54
Figure 3.14:	Train and Validation Plots for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Method -worst result-. (a) Loss Plots. (b)Accuracy Plots.	56

Figure 3.15:	Confusion Matrix for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -worst result-.	56
Figure 3.16:	Train and Validation Plots for Binary Classification (Myopathy vs. Normal)Subject Independent Evaluation Method -best result-. (a) Loss Plots. (b)Accuracy Plots.	57
Figure 3.17:	Confusion Matrix for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -best result-.	58
Figure 3.18:	Train and Validation Plots for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Method -worst result-.(a)Loss Plots. (b)Accuracy Plots.	60
Figure 3.19:	Confusion Matrix for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -worst result-.	60
Figure 3.20:	Train and Validation Plots for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Method -best result-.(a) Loss Plots. (b)Accuracy Plots.	62
Figure 3.21:	Confusion Matrix for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -best result-.	62

List of Tables

Table 2.1:	Formulas and Graphic representation of the activation functions.	18
Table 2.2:	Descriptive comparison between different approaches for neuromuscular disorders classification and their achieved accuracy.	29
Table 3.1:	Summary Statistics of the Dataset.	39
Table 3.2:	Multiclass Classification CNN Model Performance metrics Train-Test Split Approach.	45
Table 3.3:	Performance metrics for Binary Classification (Myopathy vs.. Normal) Train-Test Split Approach.	47
Table 3.4:	Performance metrics for Binary Classification (ALS vs. Normal) using Train-Test Split Approach.	49
Table 3.5:	Summary of Evaluation Metrics for the Train-Test Split Approach.	50
Table 3.6:	Performance Metrics for Multiclass Classification for Different Patients.	51
Table 3.7:	Multiclass Classification CNN Model Performance metrics Subject Independent Evaluation Approach -worst result-.	53
Table 3.8:	Multiclass Classification CNN Model Performance metrics Subject Independent Evaluation Approach -best result-.	54
Table 3.9:	Performance Metrics for Binary Classification (Myopathy vs. Normal) for Different Patients.	55
Table 3.10:	Performance metrics for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -worst result-.	57
Table 3.11:	Performance metrics for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -best result-.	58
Table 3.12:	Performance Metrics for Binary Classification (ALS vs. Normal) for Different Patients.	59
Table 3.13:	Performance metrics for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -worst result-.	61
Table 3.14:	Performance metrics for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -best result-.	63
Table 3.15:	Performance comparison with other methods.	64

List of Abbreviations

ALS	Amyotrophic Lateral Sclerosis
NMJ	Neuromuscular Junction
EMG	Electromyogram
iEMG	intramuscular Electromyogram
sEMG	surface Electromyogram
ANN	Artificial Neural Network
CNN	Convolutional Neural Network
NN	Neural Network
BCE	Binary Cross-Entropy
CCE	Categorical Cross-Entropy
STFT	Short Time Fourier Transform
MLP	multi-layer perceptron
AR	autoregressive method
RMS	root mean square
ZC	zero crossing
WL	waveform length
MAV	mean absolute value

General Introduction

Neuromuscular diseases are a group of disorders that affect the nervous system and muscles, causing muscle weakness, twitching, and wasting. Early diagnosis and detection of these diseases are essential for effective treatment. Electromyogram (EMG) is a medical test that measures the electrical activity of muscles. This test can provide valuable information about the health of muscles and nerves that control them [1].

Diagnosing conditions affecting muscles and nerves can be a complex process that requires a comprehensive medical evaluation. The evaluation usually involves a physical examination, a review of the patient's medical history, and laboratory tests. In some cases, additional tests such as a muscle biopsy, nerve conduction studies, electromyography (EMG) [2], or magnetic resonance imaging (MRI) [3] may be necessary to help make a diagnosis. The difficulty of diagnosis can vary, and in some cases, further testing or referrals to specialists may be required. In recent years, deep learning has become a powerful tool for analyzing medical data, such as EMG signals, for disease diagnosis and treatment. With its ability to learn complex patterns and relationships from large amounts of data, deep learning has the potential to greatly improve the accuracy of neuromuscular disease detection from EMG signals.

In this final year project, we aim to develop a deep learning-based model for detecting neuromuscular diseases from EMG signals [6]. Our system will be trained on a dataset of EMG signals, the N2001 EMGlab dataset [5], containing electromyograms from patients with two different neuromuscular diseases, myopathy and neuropathy, and healthy individuals [6]. Our goal is to improve the accuracy of neuromuscular diseases classification to aid in early diagnosis and intervention, by creating a robust system that can accurately distinguish between healthy individuals and those with different neuromuscular diseases based on their EMG signals.

This report presents an in-depth exploration of the classification model of electromyography (EMG) signals for the detection of neuromuscular disorders. It is organized into several chapters, each addressing different aspects of the project. Chapter 1 introduces fundamental EMG concepts, while Chapter 2 explores deep learning techniques and related works. Chapter 3 focuses on the implementation and evaluation of our classification system, presenting the tools, technologies, and data preprocessing techniques employed.

Finally, the concluding chapter summarizes key findings and offers recommendations for further research. This report provides a comprehensive examination of EMG signal classification, highlighting its significance in diagnosing neuromuscular disorders.

Chapter 1:

Electromyography Concepts

1.1 Introduction

This chapter provides an overview of the basic concepts and principles of electromyography which play a key role in understanding the electrical activity of muscles. The initial sections provide an introduction to the different types of muscles, including skeletal, smooth, and cardiac muscles, with a particular emphasis on skeletal muscles, which are the primary focus of EMG analysis, and their functional units, known as motor units.

In addition, we will discuss the neuromuscular junction, which plays a critical role in muscle activation and is a key target for many neuromuscular disorders. We will also examine the process of EMG signal generation and acquisition, including the different methods used to acquire EMG signals and the various factors that can affect their accuracy and reliability.

Finally, we will introduce some of the clinical applications of EMG signals, including their use in the detection and diagnosis of neuromuscular disorders such as myopathy, and neuropathy.

Overall, this chapter provides a foundational understanding of the concepts and principles of electromyography, which are essential for interpreting and analyzing EMG signals for neuromuscular disease detection and diagnosis.

1.2 Generalities on the Muscles

1.2.1 Definition and Types of Muscles

Muscles are specialized tissues that enable movement in the body by generating force through contraction. When a muscle contracts, it pulls on the bones and joints, causing movement in the body. As depicted in **Figure 1.1** there are three main types of muscles:

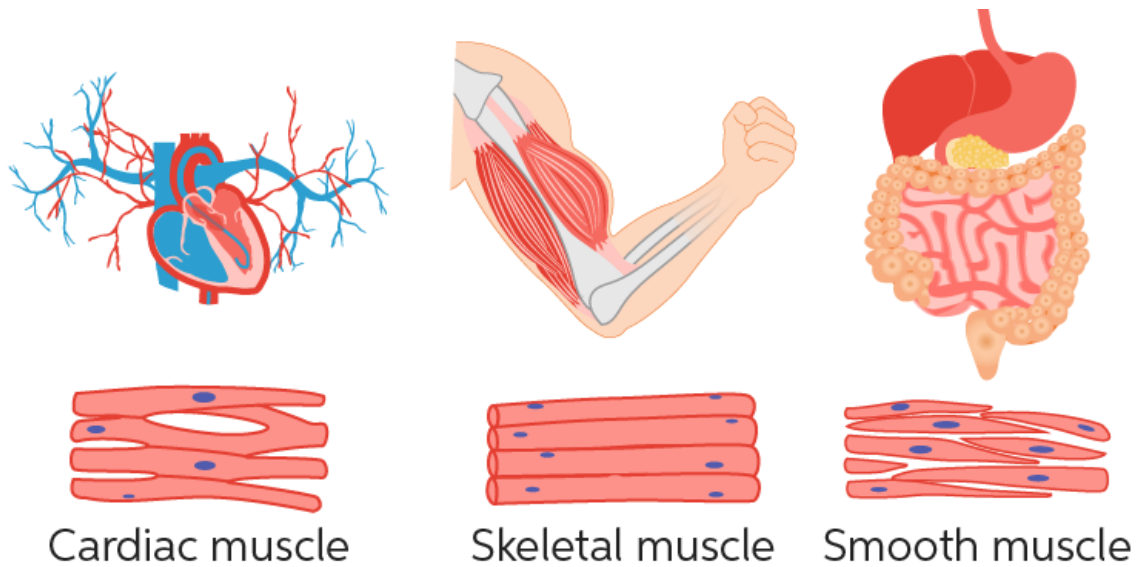


Figure 1.1: Types of muscles [7].

a. Skeletal muscle

Also known as striated or voluntary muscle, skeletal muscle is attached to bones and responsible for voluntary movements of the body. These muscles are under conscious control and can generate high force [8]. Regarding our project, we are particularly interested in this type of muscle.

b. Smooth muscle

Also known as involuntary muscle, smooth muscle is found in the walls of internal organs and blood vessels. These muscles are not under conscious control and generate low force.

c. Cardiac muscle

Found only in the heart, cardiac muscle is responsible for the involuntary contraction of the heart to pump blood. These muscles have properties of both skeletal and smooth muscle and generate moderate force [9].

Each muscle type has unique characteristics in terms of structure, function, and control. Skeletal and cardiac muscles are striated, meaning that they have a banded appearance under a microscope, while smooth muscle does not have visible striations. Skeletal and cardiac muscles also have a highly organized structure of sarcomeres, the basic contractile unit of muscle fibers, while smooth muscle has a more disorganized structure. The types of muscle fibers and energy metabolism also differ between muscle types, reflecting their specialized functions in the body.

1.2.2 Skeletal muscles

The primary function of this type of muscles is to generate movement and maintain posture in the body, comprising 30 to 40% of the total body mass. Skeletal muscles are composed of long cylindrical cells called muscle fibers, each one of them is innervated by a single motor neuron [10] which is responsible for transmitting the signals from the central nervous system to skeletal muscle fibers [11]. When the motor neuron is activated, it generates an electrical impulse that travels down the length of the neuron and is transmitted to the muscle fiber at the neuromuscular junction, Electromyography (EMG) is an approach that is used to detect these electrical signals produced by motor neurons in the muscle.

1.2.3 Motor Unit

A motor unit as shown in **Figure 1.2**, consists of a single motor neuron and all the skeletal muscle fibers that it innervates [12]. It is responsible for controlling muscle contractions, when activated by an electrical signal, all the muscle fibers in the motor unit contract together [13]. The force of a muscle contraction is dependent on the number of activated motor units, and since the number of motor units varies by muscle type, the force of a contraction also varies accordingly. Muscles that require force have large motor units with more muscle fibers. In contrast, muscles that require fine control have smaller motor units with fewer muscle fibers.

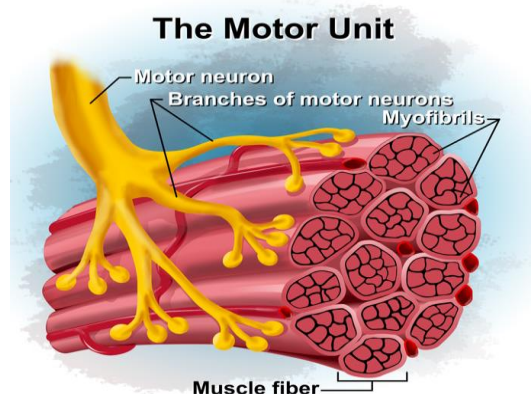


Figure 1.2: Structure of the Motor Unit [14].

1.2.4 Neuromuscular Junction

The neuromuscular junction (NMJ) establishes a connection between a motor neuron and a muscle fiber. It is responsible for initiating muscle contraction in response to the electrical signals generated by the motor neuron. The NMJ plays a crucial role in voluntary and

involuntary muscle movement, any disorder that affects it can cause neuromuscular problems such as paralysis and muscle weakness.

1.3 Electromyogram

Electromyogram (EMG) [15] is a common clinical test used to assess the function of muscles and the nerve cells that control them by measuring the electrical activity of a muscle. EMG studies are used to help in the diagnosis and management of disorders such as muscular myopathy, muscular dystrophy, and neuropathy.

1.3.1 EMG Generation in the Muscle

EMG signal is a bioelectric signal captured through the process of electromyography, it is a record of the electrical currents generated in skeletal muscles during their contraction. Specifically, it is generated by the depolarizing and repolarizing zones of the muscle fibers which generate action potentials that are spread along the muscle fibers [16]. By examining the features of EMG signals, such as the frequency, amplitude, and duration it is possible to obtain information about the neuromuscular activity. For instance, the amplitude and frequency of the EMG signal can be used to determine the level of muscle activity, the presence of any pathological changes, and the effectiveness of treatments [17].

1.3.2 EMG Signal Acquisition

An EMG (electromyography) signal is typically acquired using surface or needle electrodes that are placed on or inserted into the skin overlying or within a muscle, respectively [18]. There are two main types of EMG signal acquisition methods: surface EMG (sEMG) and intramuscular EMG (iEMG) [18].

For sEMG, the electrodes are placed on the skin surface overlying the muscle of interest [19]. This type of EMG is non-invasive and easier to apply compared to iEMG. However, the signal detected by sEMG electrodes is typically weaker and more susceptible to interference from other sources, such as electrical noise from nearby devices or movement artifacts.

For iEMG, a needle electrode is inserted directly into the muscle. This type of EMG provides a more detailed and accurate signal because it is able to capture the electrical activity of individual motor units within the muscle. However, it is also more invasive and requires specialized training to be applied safely and correctly.

The electromyogram (EMG) sensor detects electrical activity from a muscle using conductive pads placed on the skin [20]. Every time the muscle is activated, individual fibers within it receive electrical impulses, causing them to contract. The electrodes detect the electrical activity of the muscle fibers and generate a voltage signal that can be recorded and analyzed. The excitation and contraction of muscle fibers cause the electrical activity in a muscle that generates EMG signals. The EMG signal provides information about the timing and intensity of muscle activation, which can be used to assess muscle function and diagnose neuromuscular disorders [19].

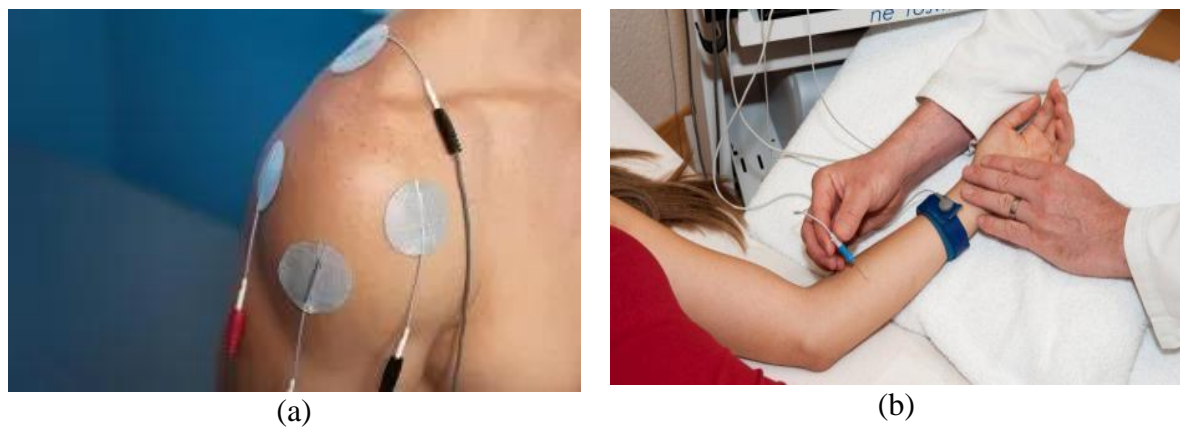


Figure 1.3: (a) Surface EMG [21]. (b) Intramuscular EMG [22].

1.2.3 Factors Affecting EMG Signals

Several factors can affect the acquisition of EMG signals [23], including:

- **Electrode Placement:** it is a critical factor in obtaining accurate signals as poor placement of the electrodes can result in a weak or noisy signal.
- **Skin Resistance:** skin should be cleaned and prepared for electrode placement to remove any oils or dirt that could affect the electrode-skin interface. Otherwise, skin resistance can create a barrier between the electrode and the skin, leading to a weak or no signal.
- **Electrical Interference:** It denotes the unwanted electrical signals that can interfere with the recording of EMG signals. They can be introduced by various sources such as electronic equipment.
- **Movement Artifacts:** It refers to the unwanted noise that can occur due to the movement of the subject being recorded or the recording electrodes.

1.2.4 Clinical Applications of EMG

EMG signals have a wide range of clinical applications, some of them are:

- Neuromuscular Disorders Diagnosis: EMG signals are commonly used in the diagnosis of disorders such as myopathy, and neuropathy. They help in distinguishing between disorders affecting muscles and those affecting nerves, while also providing insights into the severity of the condition.
- Physical Therapy [24]: EMG can be used to monitor real-time muscle activity and provide biofeedback during physical therapy sessions. This can assist patients to learn how to effectively control and enhance the strength of targeted muscles.
- Localization of Nerve Damage: when there is suspicion of a nerve injury or compression, EMG can be employed to help identify the specific location and extent of the damage.

Overall, EMG is a valuable tool in clinical applications that plays a crucial role in diagnosing and managing a wide range of neuromuscular conditions in addition to helping patients regain functional abilities.

1.4 Neuromuscular Disorders

Neuromuscular disorders are caused by pathologies affecting the nerves and muscles in the human body. As a consequence, these diseases interfere with the proper functioning of muscles and commonly manifest as muscle weakness, which constitutes the most prevalent clinical sign.

a. Myopathy

It is a general term that refers to a broad group of diseases that affect muscles, particularly, muscles connected to bones. It can have various causes, including genetic factors, infections, and metabolic disorders. The specific cause may vary depending on the individual and the type of myopathy being considered. Additionally, myopathies can be autoimmune, where the body's immune system attacks itself, leading to muscle function problems [25]. They can also be either inherited within families or acquired later in life [26]. Classification of myopathies is often based on their underlying cause.

In myopathy, there may be abnormal patterns of the electrical activity of the muscles being examined, such as increased or decreased muscle activity, which can be detected during EMG tests.

Typical symptoms of myopathy include muscle weakness, stiffness, and cramps. Some myopathies can affect muscles in the hands, feet, face, and eyes. In some cases, the problem can affect the heart and breathing muscles. As a consequence, people living with myopathy may have difficulty performing activities of daily living [27].

b. Neuropathy

It refers to the many conditions that involve damage to the peripheral nerves, which are the nerves outside the brain and spinal cord that are responsible for transmitting signals between the central nervous system and all other parts of the body [28]. It can occur due to various factors such as diabetes, nutritional or vitamin imbalances, alcoholism, exposure to toxins, and hereditary factors [29]. Depending on the type and location of the affected nerves, symptoms of the neuropathy can vary including pain or discomfort, numbness, muscle weakness, and loss of coordination or balance.

The diagnosis of neuropathy typically involves a comprehensive assessment of symptoms. Tests like electromyography (EMG) may be used to evaluate nerve function and determine the underlying cause.

1.5 Conclusion

In conclusion, this chapter has provided a general exploration of electromyography (EMG) concepts. We have discussed the critical role of the neuromuscular junction in muscle activation and highlighted factors that can affect the quality of EMG signals. Finally, we have examined some of the clinical applications of EMG signals in the detection and monitoring of neuromuscular disorders in addition to exploring different conditions such as myopathy and neuropathy. By exploiting the power of EMG, researchers, and practitioners can continue to improve the interpretation and treatment of these conditions, ultimately benefiting patients.

Chapter 2:

Deep Learning and Related Works

2.1 Introduction

This chapter aims to explore the fundamental concepts of deep learning which our project relies on to tackle the problem of neuromuscular disorder detection. We will discuss the structure and functioning of artificial neural networks, which form the basis of deep learning models. To gain a comprehensive understanding of deep learning models, we will explore the different types of layers that make up these networks. These layers include convolutional layers, recurrent layers, pooling layers, fully connected layers, batch normalization layers, and dropout layers. Each layer plays a critical role in processing the input data and learning relevant representations. Finally, we will review some related works in the field of using deep learning for neuromuscular disease detection. This review will cover the different approaches taken in these works, as well as their strengths and limitations. By examining the dataset and related works in detail, we aim to provide a comprehensive understanding of the research landscape in this area and identify potential avenues for our research.

2.2 Deep Learning

It is a subfield of machine learning, based on artificial neural networks, which is concerned with algorithms inspired by the structure and function of the human brain, allowing it to “learn” from large amounts of data. It enables brand-new products, businesses, and ways of helping people to be created. These include better healthcare, personalized education, self-driving cars, and many others [30]. **Figure 2.1** below represents the relationship between deep learning, machine learning, and artificial intelligence.

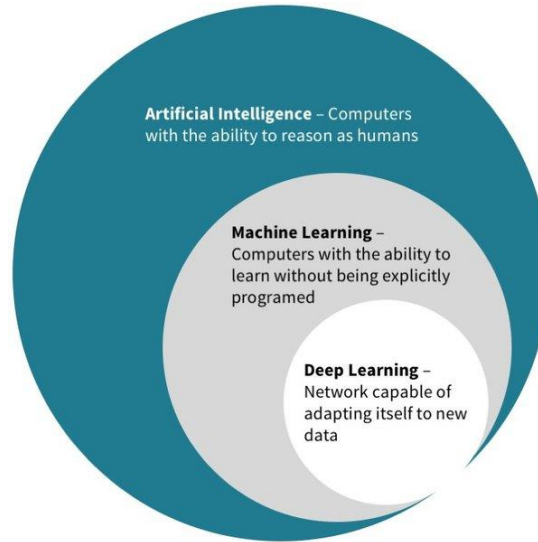


Figure 2.1: Relationship between deep learning, machine learning, and artificial intelligence[31].

In recent years, deep learning has started to gain popularity in medical image analysis and bioelectric signal processing. With the availability of large amounts of data, deep learning approaches outperform traditional feature extraction and machine learning methods in pattern detection and image recognition in terms of classification accuracy [32]. Early studies of deep learning applied to disease detection or classification have reported superior performance compared to conventional techniques or even better than medical experts in some tasks [33]. The idea of deep learning has been present for decades, and the key reasons that make it more powerful lately are, the availability of huge amounts of data, the ability to train very large neural networks either on a CPU, GPU, TPU, or any other acceleration hardware platform, and the algorithmic innovations, especially in the last several years where the main focus was about making the neural network run much faster. The effect of training data size on the performance of deep learning vs. conventional machine learning is graphed in **Figure 2.2**.

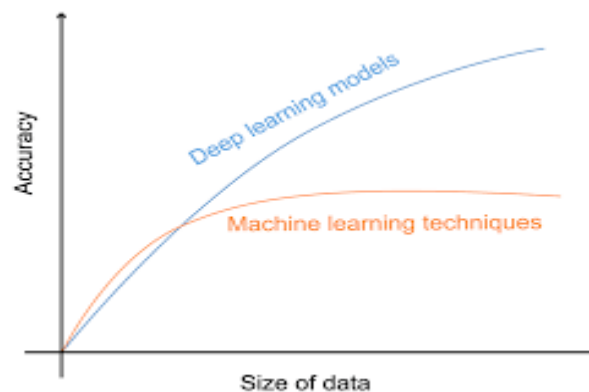


Figure 2.2: Effect of training data size on the performance of DL vs.. conventional ML [34].

2.3 Artificial Neural Networks

Artificial Neural Networks (ANNs), also called neural networks (NNs) or neural nets, are a type of machine learning algorithm that attempts to simulate the network of neurons that make up a human brain so that the computer will be able to learn things and make decisions in a human-like manner.

Cognitive neuroscientists have learned a tremendous amount about the intricacies of neural networks, brain functions, and cognitive processes in the human brain since computer scientists first attempted the original artificial neural network. One of the things they learned is that different parts of the brain are responsible for processing different aspects of information and these parts are arranged hierarchically [35]. So, input comes into the brain and each level of neurons provides insight, and then the information gets passed on to the next, more senior level. That's precisely the mechanism that ANNs are trying to replicate [36]. ANNs use different layers of mathematical processing to draw on the information it is fed. It consists of interconnected nodes -called artificial neurons- arranged in a series of layers that process information by weighing inputs and determining outputs based on set weights and biases. Neural networks need to be trained on a large amount of data called a training set, to recognize patterns and make predictions or decisions. Training a neural network involves using an optimization algorithm to find a set of parameters to best map inputs to outputs. Once the training is done, the neural net will try to classify future data based on what it thinks it is seeing (or hearing, depending on the task) throughout the different units.

Neural Networks have been successfully deployed in many fields such as computer vision, natural language processing, and robotics.

2.4 Layers in a Neural Network

For the sake of understanding the overall architecture of a NN, we define its basic building blocks which are different types of layers.

2.4.1 Convolutional layers

It is the key component of Convolutional Neural Networks (CNN). It defines a set of filters (or kernels), parameters that are to be learned throughout the training. Each filter convolves with the input and creates an activation map (feature map) [37], which in turn contributes to the

input of the next layer. Convolutional layers are followed by other layers such as pooling layers, fully connected layers, and normalization layers.

For the convolution operation, the filter is slid across the height and width of the input and the dot product between every element of the filter and the input is calculated at every spatial position. **Figure 2.3** shows an example of the convolution operation where the filter is convolved with an input matrix starting from the upper-left corner. It starts sliding according to the stride value of that layer, performing a dot product at each step.

One problem when applying a convolutional layer is that the input tends to shrink which causes a lot of information to be lost near the edges of the input. A solution to this issue is by padding the input. This is achieved by adding zeros to the input sequence to create a new padded sequence before applying the filters. This effectively increases the size of the input tensor and helps to maintain the spatial dimensions of the output volume.

Padding is often used in combination with a stride, which defines the step size of the convolutional filter as it moves through the input of the layer. When performing convolution on an input matrix of size (n_H, n_W, n_C) with a filter size of (f, f, n_C) , using a stride value 's' and a padding value 'p', the resulting feature map will have dimensions $(\frac{n_H + 2p - f}{s} + 1, \frac{n_W + 2p - f}{s} + 1)$. Where, 'n_H' represents the height of the input matrix, 'n_W' represents the width, 'n_C' represents the number of channels, and 'p' represents the padding value.

In a convolutional layer, a neuron is only connected to a local area of input neurons instead of a full connection. In other words, the receptive field size, which is the restricted area of the previous layer from which the neuron receives input of each neuron is small and is equal to the filter size.

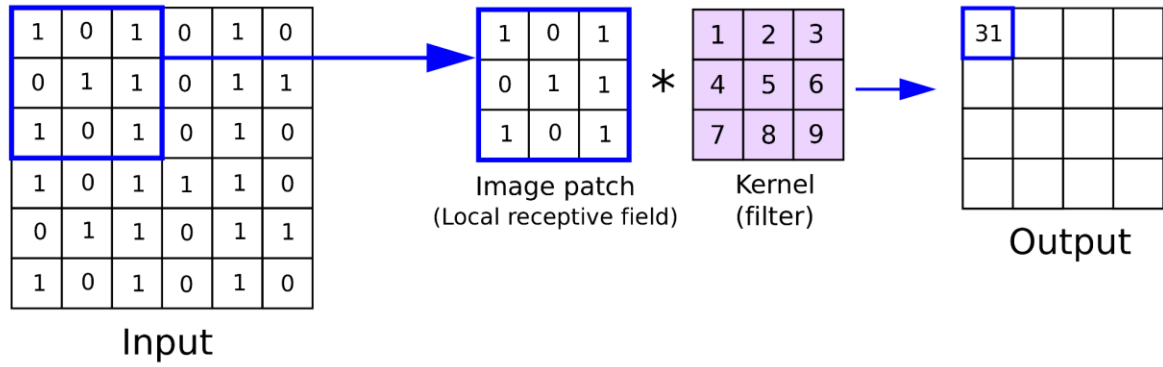


Figure 2.3: Example of the convolution process in a convolutional layer [38].

2.4.2 Pooling layers

Pooling layers are a critical component of deep learning neural networks. These layers are usually applied after one or more convolutional layers and before one or more fully connected layers. They work by reducing the spatial dimensions, length, or the number of elements in the input data while preserving important features, such as edges and color information [30].

Depending on its type, the pooling operation is achieved by taking the maximum or average value within a defined window or filter size across each channel of the input tensor. The resulting output from the pooling layer is then fed into the next layer of the neural network for further processing.

Pooling layers are important for improving the computational efficiency of neural networks and reducing overfitting. However, it is essential to choose the right type and size of pooling layer based on the specific task at hand [39].

2.4.3 Fully connected layers

Also known as Dense Layers, are a type of layer commonly used in ANNs, where each neuron is connected to every neuron in the previous layer [40] with each connection having an associated weight. The output of a fully connected layer is obtained by applying a linear transformation, followed by an activation function, which can be mathematically expressed as

$$Y = \text{activation}(Wx + b) \quad (1)$$

where ‘activation ’ is an activation function that controls how well the network model learns the training dataset, it is applied element-wise to the linear transformation ‘ $Wx + b$ ’. ‘ W ’ is the weight matrix, ‘ x ’ is the input features vector, and ‘ b ’ is the bias vector.

The aim of using fully connected layers is to learn non-linear mappings from the input features to output labels by iteratively adjusting the weights and biases using backpropagation to minimize the difference between the predicted output and the actual output.

These layers are typically placed at the end of a neural network architecture, where features have been extracted earlier using convolutional or pooling operations after that flattened into a 1D vector and passed through one or more fully connected layers. The number of neurons in the final fully connected layer is equal to the number of output classes of the network [30].

2.4.4 Recurrent layers

Recurrent Layers are a type of layer in a neural network that processes sequential data, such as time series or text, by maintaining a memory of previous inputs. They allow the network to capture long-term dependencies in the input sequence [41]. Recurrent Layers can be Simple Recurrent Layers (SRL), Long Short-Term Memory (LSTM) Layers, or Gated Recurrent Unit (GRU) Layers. They are commonly used in language modeling, machine translation, and speech recognition.

2.4.5 Batch normalization layer

Batch normalization is a method used to improve the training stability and speed of a NN by normalizing (re-centering and re-scaling) the activations of a neural network's intermediate layers[42].

The batch normalization layer involves two primary operations: first, it centers the input activations to have zero mean (μ) by subtracting the batch mean. After that, it normalizes the activations by dividing by the batch standard deviation (σ), these operations can be expressed mathematically as follows:

- Given a mini-batch of activations denoted ‘ t ’, the batch mean ‘ μ ’ and the variance σ^2 are calculated as follows:

$$\mu_t = \frac{1}{m} * \sum_{i=1}^m x_i \quad (2)$$

$$\delta_t^2 = \frac{1}{m} * \sum_{i=1}^m (x_i - \mu_t)^2 \quad (3)$$

where ‘m’ is the mini-batch size, and ‘x_i’ is the activations of the ith neuron in the mini-batch.

- The normalized activations ‘Z’ can then be found by applying the following transformation:

$$Z_i = \frac{x_i - \mu_t}{\sqrt{\delta_t^2 + \varepsilon}} \quad (4)$$

where ‘ε’ is a small constant added for numerical stability.

This technique efficiently facilitates learning and can also serve as a regularization method to prevent overfitting of the model. It is added to the sequential model to standardize the input or output and can be applied at multiple points between the model layers [43].

2.4.6 Dropout layer

Dropouts are the regularization technique that is used to prevent overfitting in the model. By randomly deactivating a percentage of neurons in the network, dropouts interrupt the connections between incoming and outgoing from those neurons. Typically, dropouts are recommended after dense layers rather than convolutional layers [44]. To ensure optimal results, it is advisable to limit the dropout rate to 50% or lower [45]. Higher rates may lead to poor learning and compromised predictions.

2.5 Activation functions

Activation functions are an essential component of neural networks. They introduce non-linearity to the output of a neuron, which is essential for modeling complex patterns and relationships in data.

Activation functions are applied to the weighted sum of inputs to a neuron and transform the output into a non-linear form. The choice of the activation function is crucial since it affects the network's ability to learn and its performance [30].

There are several activation functions commonly used in neural networks, including:

- **Sigmoid Function:** The sigmoid function takes any real-valued number and maps it to a value between 0 and 1. It is often used in binary classification problems [46].
- **ReLU Function:** The Rectified Linear Unit (ReLU) function takes any real-valued input and returns the maximum between the input and 0. It is a simple and effective activation function that is often used in deep learning models [47].
- **Tanh Function:** The hyperbolic tangent (Tanh) function maps any real-valued number to a value between -1 and 1. It is often used in classification tasks and is similar to the sigmoid function [48].
- **Softmax Function:** The softmax function is often used as the output activation function in a neural network. It takes a vector of real numbers and normalizes them to yield a probability distribution [49].

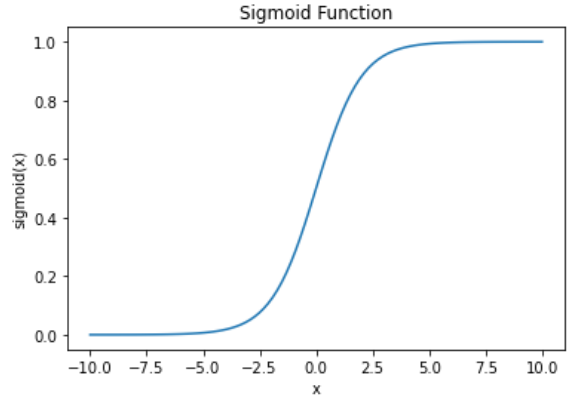
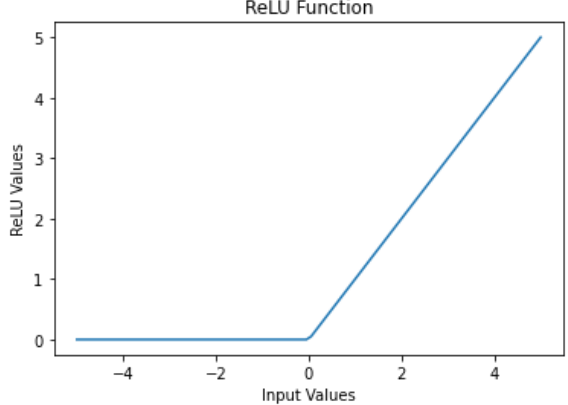
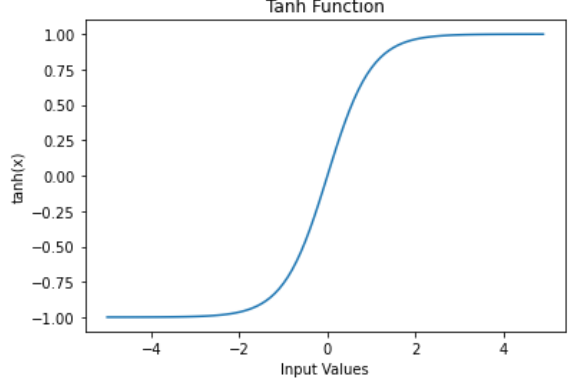
Softmax function doesn't have a single graph that can be plotted as it depends on the input values. Assuming we have an input vector $Z = [z_1, z_2, \dots, z_n]$, then applying the softmax function to each element of Z will be as follows

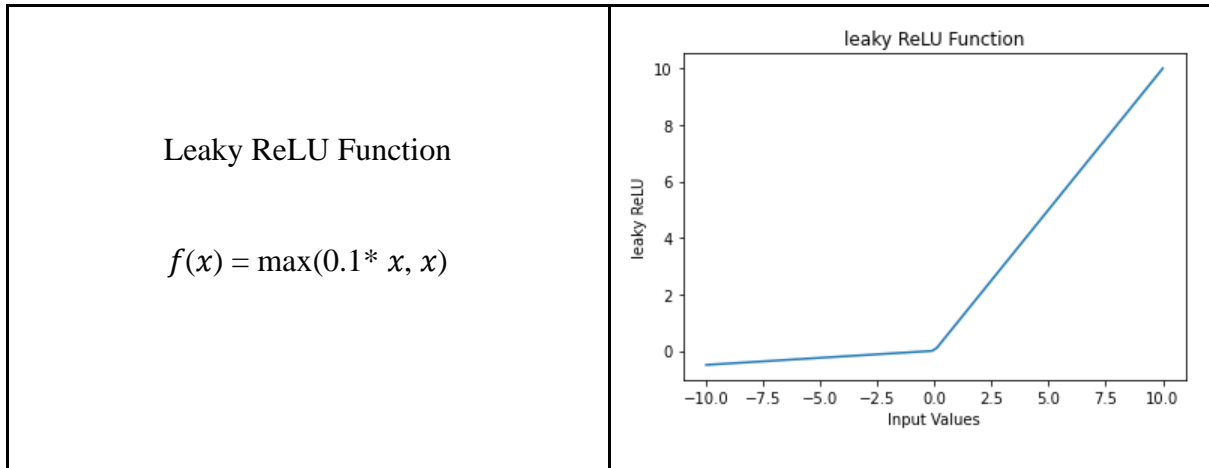
$$\text{softmax}(z_j) = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \text{ for } j = 1, \dots, K \quad (5)$$

- **Leaky ReLU Function:** The Leaky ReLU function is a variant of the ReLU function that allows for a small gradient when the input is negative. It is often used in deep learning models to prevent the "dying ReLU" problem [50].

The choice of activation function is an important consideration in designing and training neural networks for various applications.

Table 2.1: Formulas and graphic representation of the activation functions.

Function	Graph
<p>Sigmoid Function</p> $f(x) = \frac{1}{1 + e^{-x}} \in (0,1)$	 <p>The graph shows the Sigmoid Function, which is an S-shaped curve. The x-axis ranges from -10.0 to 10.0 with major ticks every 2.5 units. The y-axis, labeled 'sigmoid(x)', ranges from 0.0 to 1.0 with major ticks every 0.2 units. The curve starts near 0 for negative x, passes through (0, 0.5), and approaches 1 for positive x.</p>
<p>ReLU Function</p> $f(x) = \max(0, x)$	 <p>The graph shows the Rectified Linear Unit (ReLU) Function. The x-axis is labeled 'Input Values' and ranges from -4 to 4 with major ticks every 2 units. The y-axis is labeled 'ReLU Values' and ranges from 0 to 5 with major ticks every 1 unit. The function is zero for all negative input values and increases linearly with a slope of 1 for all positive input values.</p>
<p>Tanh Function</p> $f(x) = \tanh(x)$	 <p>The graph shows the Hyperbolic Tangent (Tanh) Function. The x-axis is labeled 'Input Values' and ranges from -4 to 4 with major ticks every 2 units. The y-axis, labeled 'tanh(x)', ranges from -1.00 to 1.00 with major ticks every 0.25 units. The curve is an S-shape centered at the origin (0, 0), approaching -1 as x goes to negative infinity and 1 as x goes to positive infinity.</p>



2.6 Training a neural network

As a neural network is being trained, its parameters are iteratively adjusted through a sequence of forward and backward propagations, while the aim of these parameters updates is to reduce the discrepancy between the expected output and the actual output.

The training of a neural network involves multiple stages, starting with the initialization of the network's parameters. Throughout the training process, several steps are undertaken to optimize the network's performance. The output is then determined using the current parameters once the input data has been sent through the network. A loss function, such as mean squared error or cross-entropy, is then used to determine the discrepancy between the predicted and actual output [30].

The error is then spread back through the network in the backward propagation step. With the chain rule of calculus, the gradient of the loss function with respect to each parameter is determined. The parameters are then updated using an optimization algorithm. This process is repeated for a specified number of epochs, or until the error minimization stabilizes.

The success of training a neural network depends on various factors, such as the choice of the loss function, the optimization algorithm, the learning rate, the number of learnable parameters, and the activation functions. To avoid underfitting or overfitting the data, careful tuning of these training parameters is necessary.

The main concepts and techniques required for training a neural network are presented in the following:

2.6.1 Backpropagation

Backpropagation is a fundamental algorithm used in training neural networks. The algorithm calculates the gradient of the loss function with respect to the network's parameters, enabling the network to learn from the data by adjusting its parameters in the direction of the steepest descent of the loss function.

The backpropagation algorithm involves two main steps: a forward pass and a backward pass. During the forward pass, the output is computed using the current parameters for a given input. Then, during the backward pass, the error is propagated back through the network, allowing for the update of the parameters. This process of forward and backward passes is repeated until the network reaches convergence.

Backpropagation has proven to be an effective algorithm for training deep neural networks, enabling them to learn complex and non-linear representations of data. The algorithm has been widely used in many applications, including image classification, speech recognition, and natural language processing [47].

2.6.2 Optimiser/gradient descent

In deep learning, optimizing the neural network's parameters is a critical task, and gradient descent is the most commonly used optimization algorithm.

Gradient descent works by iteratively adjusting the parameters in the direction of the negative gradient of the loss function. However, there are various challenges associated with using gradient descent, such as choosing an appropriate learning rate, dealing with vanishing or exploding gradients, and getting stuck in local minima. To address these challenges, researchers have developed various optimization algorithms, such as Momentum, Adagrad, Adam, and RMSprop, that aim to improve convergence speed and robustness. Additionally, regularization techniques such as L1 and L2 regularization, dropout, and early stopping can improve generalization performance. Recent advancements in optimizers, such as second-order optimization methods and meta-learning-based optimizers, have also shown promising results in improving optimization performance [51].

Choosing the appropriate optimizer is an essential aspect of deep learning model training, and it should be chosen based on the specific task, dataset, and model architecture.

2.6.3 Loss function

An algorithm's loss function measures the distance between its current output and its expected output. It evaluates how the algorithm models data. It can be categorized into two groups. One for classification (discrete values, 0,1,2...) and the other for regression (continuous values) [52]. More detail on some common loss functions is provided in the following:

- Mean Squared Error (MSE): is a commonly used loss function in regression problems. It measures the average squared difference between the predicted and actual values of the target variable.

$$\text{MSE} = 1/n * \sum (y_{\text{pred}} - y_{\text{actual}})^2 \quad (6)$$

- L1 Loss: is a commonly used loss function in regression problems. It measures the absolute difference between the predicted and actual values of the target variable.

$$\text{L1_Loss} = 1/n * \sum |y_{\text{pred}} - y_{\text{actual}}| \quad (7)$$

- Binary Cross-Entropy: is a commonly used loss function for binary classification problems.

$$\text{BCE} = -[y_{\text{actual}} * \log(y_{\text{pred}}) + (1 - y_{\text{actual}}) * \log(1 - y_{\text{pred}})] \quad (8)$$

- Categorical Cross-Entropy: is a commonly used loss function for multi-class classification problems.

$$\text{CCE} = - \sum y_{\text{actual_i}} * \log(y_{\text{pred_i}}) \quad (9)$$

Both categorical and binary cross-entropy measure the difference between the predicted and actual class probabilities.

2.6.4 Epoch, batch, and iteration

When training deep learning models, epochs, batches, and iterations play an important role. Epochs refer to the number of times the entire training dataset is passed through the model during training, while batches are subsets of the training data that are used to perform one update of the model's parameters. Iterations refer to the number of batches processed to complete one epoch during the training of a neural network. The choice of batch size, number of epochs, and iterations can have a significant impact on the performance of deep learning models. In order to optimize these hyperparameters, it is important to monitor validation loss and use techniques like early stopping [30]. By carefully selecting and tuning these hyperparameters, it is possible to achieve the best possible performance for a given deep-learning task.

2.6.5 Train, validation, and test subsets

The training subset is used to optimize the model's parameters, while the validation subset is used to estimate the performances of each epoch, which allows to tune hyperparameters and to prevent overfitting by checking how the model acts on unseen data during training. The test subset is used to evaluate the performance of the model on unseen data. It is important to ensure that the data is split randomly and that the subsets are representative of the overall dataset. Typically, a split of 60-20-20 (train-validation-test) is used, however, this can vary depending on the size of the dataset and the specific needs of the task [53].

Another important consideration is to avoid leaking information from the validation or test subsets into the training subset, as this can lead to over-optimistic results. To address this, it is important to preprocess the data and split it before performing any feature selection or model training.

2.7 Performance Metrics

Performance Metrics are a part of every machine learning pipeline. They tell you if you're making progress, and put a number on it. All machine learning models need a metric to judge performance. These metrics are chosen depending on the specific problem and dataset, and it is important to interpret them within the context of the problem being solved.

2.7.1 Confusion Metrics

A confusion matrix is a table used to evaluate the performance of a machine learning model for classification tasks. It has four possible outcomes for each class:

- True Positive (TP): The model correctly predicted the positive class.
- True Negative (TN): The model correctly predicted the negative class.
- False Positive (FP): The model incorrectly predicted the positive class.
- False Negative (FN): The model incorrectly predicted the negative class.

The confusion matrix can be used to calculate a variety of metrics, such as accuracy, precision, recall, and F1 score.

2.7.2 Accuracy

Accuracy is a performance metric that measures the proportion of correct predictions made by a machine learning model on a given dataset. It is calculated by dividing the number of correct predictions by the total number of predictions made. The formula for calculating accuracy is given in Eq. 10.

$$\text{Accuracy} = \frac{(TP + TN)}{(TP + TN + FP + FN)} \quad (10)$$

2.7.3 Precision

Precision is a performance metric that measures the proportion of true positives (correctly predicted positive instances) among all positive predictions made by a machine learning model. It is calculated by dividing the number of true positives by the sum of true positives and false positives.

$$\text{Precision} = \frac{TP}{(TP + FP)} \quad (11)$$

2.7.4 Recall

Recall is a performance metric that measures the proportion of true positives (correctly predicted positive instances) among all actual positive instances in a dataset. It is calculated by dividing the number of true positives by the sum of true positives and false negatives.

$$\text{Recall} = \frac{TP}{(TP + FN)} \quad (12)$$

2.7.5 F1 score

The F1 score is a single metric that combines precision and recall, providing a balanced measure of a model's performance by considering both the ability to correctly identify positive instances (precision) and the ability to capture all positive instances (recall). It is calculated as follows

$$F1\ Score = \frac{2 * Precision * Recall}{Precision + Recall} \quad (13)$$

2.8 Related works

In recent years, deep learning techniques have shown great promise in classifying EMG signals for disease detection [54] [55]. This section presents a review of related works in EMG signal classification for neuromuscular disease detection using deep learning methods including Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), and Multi-Layer Perceptron (MLP). We explore the various deep-learning techniques employed in EMG signal analysis and discuss their performance in disease detection. The paper selection process was based on a defined set of criteria that aimed to identify recent publications that provided significant insights into the utilization of deep learning techniques and achieved optimal outcomes. Besides, the primary focus was on selecting papers that used similar data to the one employed in this study. The set of selected papers is summarized in the following.

2.8.1 DeepEMGNet: An Application for Efficient Discrimination of ALS and Normal EMG Signals

DeepEMGNet [56] proposes a deep learning application for the classification of amyotrophic lateral sclerosis (ALS) and normal electromyogram (EMG) signals. The paper shows the importance of EMG signals in analyzing neuromuscular diseases.

The authors of the paper highlight that current methods for discriminating between ALS and normal EMG signals can be time-consuming and require expert interpretation. They argue that DeepEMGNet provides a more efficient and accurate approach to this task.

This study introduces an approach that combines the time-frequency representation of EMG signals and convolutional neural networks (CNN). The Short Time Fourier Transform (STFT) is employed to achieve the time-frequency representation of the EMG signals. The CNN architecture consists of two convolution layers, two pooling layers, and a fully connected layer.

The softmax function is used to calculate the probability of each class label. The dataset used for evaluation was released from the University of Copenhagen in 2001 [5] and it contains 89 ALS signals and 133 Normal signals, for data augmentation, a random sampling mechanism was adopted. The paper also compares the proposed method with other state-of-the-art methods and shows that the proposed method outperforms them in terms of accuracy, sensitivity, and specificity. The accuracy of the proposed method was 96.69%, the sensitivity was 94.24%, and the specificity was 97.59%.

The authors suggest that the model could be used as a diagnostic tool to assist healthcare professionals in the early detection of these illnesses.

2.8.2 Electromyography (EMG) based Classification of Neuromuscular Disorders using Multi-Layer Perceptron

The aim of this study [57] is to develop an accurate automatic diagnostic system to classify intramuscular EMG signals into healthy, myopathy, or neuropathy categories to aid the diagnosis of neuromuscular diseases.

The data were obtained from an EMG lab database [58]. The EMG signals were obtained from healthy subjects and subjects suffering from neuropathy or myopathy with a different age mean. There are 5 patients for each group divided into 100 samples.

The proposed method uses a multi-layer perceptron (MLP) for classifying the EMG signals. The MLP is trained using features extracted from the raw EMG signals. However, the paper does not mention any details about the architecture of the MLP model. This makes it difficult to replicate the results of the study or to compare the performance of the MLP model to other models. The feature extraction process involves transforming the EMG signals into a set of features such as autoregressive method (AR), root mean square (RMS), zero crossing (ZC), waveform length (WL), and mean absolute value (MAV). The proposed method is evaluated using EMG signals corresponding to three different classes, one for healthy subjects and two for subjects with neuromuscular disorders.

The methods such as Root Mean Square, Zero Crossing, Wavelength, and Mean Absolute Value are suitable methods for feature extraction. However, most of the highest results of classification come from Autoregressive (AR) methods, achieving 86.3% accuracy in distinguishing healthy, myopathy, and neuropathy groups.

2.8.3 EMG Signal Classification for Detecting Neuromuscular Disorders

The approach proposed in this paper [59] aims to differentiate neuromuscular disorder patients from healthy people based on EMG signals that were recorded from the biceps. Eleven features that are: Root mean square, Waveform length, Variance of EMG, Maximum fractal length, Modified mean absolute Value, Enhanced Wavelength, Difference Absolute Standard Deviation Value, Average Amplitude Change, Variance of FFT, FFT maximum density, Variance of neo, were extracted from the EMG signals and then given as input for an Artificial Neural Network (ANN). This ANN is composed of a two-layered feed-forward network with specific characteristics was employed. The network consisted of an input layer, a hidden layer with sigmoid activation neurons, and an output layer with softmax activation neurons. This type of network is commonly referred to as a "patternnet." The reported classification accuracy is 85%, which is acceptable but not to be trusted for medical purposes.

2.8.4 Deep learning-based diagnosis of myopathy and neuropathy

This paper [55] proposes a deep learning-based approach for diagnosing myopathy and neuropathy. The authors collected a dataset of EMG signals from patients who visited Seoul National University Hospital and underwent EMG between June 2015 and July 2020 with myopathy and neuropathy and used it to train and evaluate a convolutional neural network (CNN) model. The dataset used in this paper is not publicly available due to privacy considerations. The authors of the paper state that the dataset contains medical information about patients who have been diagnosed with myopathy or neuropathy. However, the paper did not provide any specific statistics or further detail about the dataset.¹

The deep learning-based algorithm was based on a convolutional neural network (CNN) architecture, which was compared to traditional machine learning (ML) algorithms such as support vector machines (SVMs) and random forests (RFs). The results showed that the CNN-based algorithm outperformed the traditional ML algorithms in terms of accuracy, sensitivity, and specificity. The CNN-based algorithm achieved an accuracy of 92.5%, a sensitivity of 91.7%, and a specificity of 93.3%, while the SVM and RF algorithms achieved accuracies of

¹ We attempted to reach the authors of the paper and requested access to the dataset, but we did not receive a response. We believe that the dataset would be valuable for our research and the development of our project.

85.0% and 87.5%, respectively. However, further research is needed to validate the proposed approach on larger and more diverse datasets.

The paper concludes that the proposed approach shows promise in improving the accuracy and efficiency of the diagnosis of myopathy and neuropathy, which can lead to better patient outcomes.

2.8.5 Analysis and Classification of Muscular Paralysis Disease using Electromyography Signal with Machine Learning

The objective of this study [60] is to utilize features extracted from EMG signals in both time and frequency domains to distinguish between normal and paralyzed conditions. The study focuses on twelve statistical features extracted from the EMG signals, including Mean Value, Variance, Mean Absolute Value, Root Mean Square, Waveform Length, Zero crossing, Log Detector, Difference Absolute Standard Deviation Value, Average Amplitude Change, Variance Absolute Value, Kurtosis of signal, and Skewness of the signal. These features are used to classify Paralysis and Normal conditions. To perform the classification, various Machine Learning techniques are employed, including Multi-Layer Perceptron (MLP), Support Vector Machine (SVM), Random Forest (RF), Gradient Boosting (GB), and Nearest Neighbor (NN) classifier models.

The EMGLab dataset [5] is used for evaluation. For feature extraction, 300.37 msec rectangular windows with an overlap of 99.84 msec are used. The classification is performed with test sample sizes of 40, 30, 20, and 10%, and training sample sizes of 60, 70, 80, and 90%, respectively.

The obtained accuracies per classifier are as follows, 72% for MLP, 73% for SVM, 72% for RF, 71% for GB, and 69% for NN. The best-performing ML model is SVM. Most of the recent works have better accuracy than these classifiers.

2.8.6 ALSNet: A Dilated 1-D CNN For Identifying ALS From Raw EMG Signal

In this study [61], a novel approach called ALSNet is proposed for the identification of Amyotrophic Lateral Sclerosis (ALS) using raw EMG signals. Unlike traditional methods that rely on manual feature extraction, ALSNet eliminates the need for such preprocessing and can

directly detect ALS subjects. This makes the method more feasible for practical implementation by reducing the computational cost required for extracting features.

In the proposed architecture of ALSNet, there are three 1D convolution layers. Each convolution layer is followed by a ReLU activation function and a batch normalization layer. The dilation rate of the convolution layers progressively increases, with rates of 1, 2, and 3, respectively. After the final convolution layer, a global max pooling layer is employed to extract the most important features. The output from the pooling layer is then passed through two fully connected layers. The first dense layer consists of 64 nodes and is followed by a ReLU activation function and a batch normalization layer. The final dense layer serves as the output layer, containing a single node with a Sigmoid activation function.

The clinical EMG signals of N2001 EMGLAB open access Dataset [5] were used in our experiment, each signal was segmented into 11 segments and each of the segments had a time duration of 1s. The dataset was split into train, validation, and test sets by a ratio of 80:20:25. The training, validation, and test sets were made up of data from different subjects so that the proposed model can be trained and evaluated properly.

The performance of the ALSNet in terms of overall accuracy, sensitivity, specificity, and balanced accuracy showed good promise.

Table 2.2 provides a comprehensive overview of the various approaches employed for classifying neuromuscular disorders, along with the corresponding accuracy levels achieved by each method. This table presents a descriptive comparison among the different approaches

Table 2.2: Descriptive comparison between different approaches for neuromuscular disorders classification and their achieved accuracy.

Reference		Methods	Subjects			Dataset	Muscle	Evaluation
Ref	Year		Group	N°Sub.	N° Signals			ACC (%)
[56]	2018	STFT and CNN	AH	(8,10)	(89,133)	[5]	BB	96.69%
[57]	2015	AR, RMS, ZC, WL, MAV, and MLP	AMH	(5,5,5)	(100,100,100)	[58]	BB	86.3%
			HS	(5,5)	(100,100)			82.5%
			MH	(5,5)	(100,100)			81%
			AH	(5,5)	(100,100)			80.5%
			MA	(5,5)	(100,100)			77%
[59]	2021	ANN, and 11 feature extracting methods	HS	(7,10)	(80,80)	[5]	BB	85%
[55]	2023	CNN	AMH	(20,20,20)	NS	NS	NS	92.5%
[60]	2022	12 feature extracting methods, MLP, RF, SVM, GB, NN	AMH	(8,7,10)	(332,315,300)	[5]	BB,VM	73%
[61]	2022	CNN	AH	(8,10)	(151,151)	[5]	BB,VM	97.74%

Notes: A = ALS, H = Healthy, M = Myopathy, S = Sick, BB = Biceps Brachii, VM = Vastus Medialis, NS = Not Specified.

2.9 Conclusion

In conclusion, the review of the papers on EMG signal classification for neuromuscular disease detection using deep learning has highlighted several important features and architectures. The statistical features have been widely used and have shown promising results in terms of accuracy. time-frequency representation of EMG signals-based methods have been explored in some papers, while raw signal processing has been used in others, with mixed results. Among the network architectures, ANN, MLP, and CNN models have been proposed, with varying degrees of success. The evaluation data used in the papers varied greatly, making direct

comparisons between the results impractical. EMGLab was the only public dataset used in some papers, and achieved results vary greatly from one paper to another.

To the best of our knowledge, it is noteworthy that the majority of the reviewed papers did not specifically address the issue of subject independence. This indicates that data from the same subject might be present in both the training and test sets, compromising the validity of the results. To address this limitation, our study focuses on proposing a model that overcomes this challenge by employing subject-independent data for training, validation, and testing. We will elaborate on this approach in the subsequent chapter.

Furthermore, it is worth noting that the last paper entitled "ALSNet" adopted the subject-independent evaluation approach but within the narrower context of binary classification for ALS vs. Normal cases. Drawing inspiration from their work, we have incorporated and expanded upon this approach in our research, encompassing both multiclass and binary classifications.

Chapter 3:

Implementation and Evaluation of the Classification System

3.1 Introduction

Chapter 3 focuses on the design and implementation of the classification model for EMG signals in the context of neuromuscular disease detection. The chapter provides an overview of the tools, technologies, and libraries employed. Furthermore, it provides a description of the data utilized for the classification system and explains the data preparation techniques used to ensure reliable and meaningful analysis. Moreover, it presents the proposed network architecture based on a CNN model and discusses the training methodology. The chapter also highlights the results and discussion of two evaluation approaches: the train-test-validation split and subject-independent evaluation. Additionally, it includes a concise comparison of our work's performance with related studies in the field.

This chapter serves as a foundational guide for understanding the design and implementation of the classification system, setting the stage for subsequent analysis and conclusions in the project.

3.2 Implementation

3.2.1 Tools and Technologies

a. Python

It is a widely used high-level, interpreted programming language. It is favored for its ease to use, simplicity, and readability, making it an ideal programming language for both beginners and experts [62].

In deep learning, Python is frequently used for data processing, model building, and training. It offers a rich set of libraries for deep learning such as Keras, Tensorflow, and PyTorch, providing high-level abstractions for building and training networks.

b. Jupyter Notebook

It is an open-source web-based tool that provides an interactive environment where users can write and execute code cells, visualize data and results, and experiment with different algorithms [63].

For deep learning, it is commonly used for model development, data exploration, and visualization. It is frequently used along with deep learning libraries allowing users to train and evaluate deep neural networks.

c. Kaggle

It is an online platform that is dedicated to data science and machine learning. It provides a cloud-based environment for running code, which can be used for training machine learning models and running experiments. The platform comes with pre-installed libraries and frameworks, such as Keras, Tensorflow, and PyTorch, and can be customized to use accelerators such as GPUs and TPUs to speed up the training of deep learning models.

d. Google Collaboratory

Collaboratory, often shortened to “Colab”, is a free cloud-based Jupyter Notebook environment by Google that allows the creation and execution of Python code using Jupyter Notebook format. It integrates with other Google services, such as Google Drive, allowing users to import and export data easily. Google Colab is an excellent tool for deep learning tasks and offers additional features over vanilla Jupyter Notebook, such as providing free access to GPU (Graphics Processing Unit) and TPU (Tensor Processing Unit) resources, which are powerful computing accelerators.

3.2.2 Libraries

a. Keras

It is an open-source library that operates on top of deep learning frameworks such as Tensorflow, and Theano. It also takes advantage of their performance optimizations and hardware support, such as GPU accelerations. Keras offers a high-level API for creating and training deep neural networks, allowing developers and researchers to efficiently experiment with various models and architectures.

b. NumPy

It is a powerful Python library for scientific computing and data analysis. It is used for working with arrays by offering numerous functions for performing numerical operations on these arrays. With NumPy, users can perform linear algebra, Fourier transform, and matrix manipulations among other mathematical operations. NumPy's versatile array object enables users to work with vast datasets with ease and speed.

c. Pandas

It is an open-source Python library used for working with datasets. The name “Pandas” is derived from the term “panel data” which refers to a type of dataset commonly used in economics, mathematics, and statistical analysis. Pandas offers various data manipulation capabilities including data cleaning, filtering, transformation, and dataset merging. It supports multiple data formats such as CSV, Excel, SQL, and more.

d. Waveform-database (WFDB)

It is a library that provides a range of tools for reading, writing, processing, and visualization of various types of physiological signals such as EMG and electrocardiogram(ECG) and associated annotations. WFDB is widely used in both research and clinical contexts and is supported on many different platforms and programming languages.

e. Matplotlib

It is a plotting library for Python that is built on NumPy arrays. Matplotlib allows creating static, interactive, and dynamic representations of data in Python. It is a powerful tool for data visualization. In this work, it was utilized to visualize the input data. Specifically, it was employed to generate visual representations of the EMG signals from each class, allowing for a comprehensive examination of the data. Furthermore, Matplotlib was utilized to plot the accuracy and loss graphs. In addition, we used it to generate a confusion matrix, facilitating a visual representation of the classification performance.

f. Scikit-learn

Also known as Sklearn, is a widely used open-source machine learning library for Python. It is built on top of other scientific computing libraries such as NumPy, SciPy, and Matplotlib. This library provides tools and algorithms for various machine-learning tasks, including classification, regression, model selection, and preprocessing of data. In addition, it provides methods for data splitting, cross-validation, hyperparameter tuning, and performance metrics.

3.3 Data and Preprocessing

To tackle the problem of neuromuscular disease classification, we utilize the EMGLab dataset [5]. As stated before, this dataset is the only publicly available dataset for addressing this problem. Therefore, the choice of the data is straightforward. In the following, we describe the

dataset and the preprocessing steps we performed to improve data quality and prepare it for model evaluation.

3.3.1 Description of the data

EMGlab or N2001 database contains clinical EMG signals gathered from three groups of subjects: one normal control group, one group of patients with myopathy, and one group of Amyotrophic Lateral Sclerosis (ALS) [64] patients. The latter is a kind of neuropathy. It is a progressive neurodegenerative disease that affects nerve cells responsible for controlling voluntary muscle movement, causing a loss of muscle control. The exact cause of ALS is largely unknown, although a combination of genetic and environmental factors is thought to play a role. The symptoms of ALS vary from person to person but commonly include muscle atrophy, muscle twitching, and cramps, individuals may experience difficulty with activities such as walking, speaking, swallowing, and breathing.

The EMG collected recordings are distributed as follows:

- 10 healthy subjects that have an age interval of 21–37, 4 females and 6 males. None of them had signs or a history of neuromuscular disorders
- 7 myopathy patients with an age ranging from 19 to 63, 2 females and 5 males. All 7 had clinical and electrophysiological signs of myopathy.
- 8 ALS patients that cover the age interval 35–67, 4 females and 4 males. Besides clinical and electrophysiological signs compatible with ALS, 5 of them died within a few years after the onset of the disorder, supporting the diagnosis of ALS.

Signals were acquired from the abductor pollicis brevis (a muscle in the hand), the biceps brachii (upper arm muscle), the vastus medialis and vastus lateralis (muscles in the thigh), the tibialis anterior (a muscle in the lower leg), the deltoideus (shoulder muscle), the tensor fasciae latae (muscle in the thigh and hip), and the triceps brachii (muscle in the upper arm), by inserting a concentric needle electrode at five different places at three insertion depths mentioned as low, medium and deep.

The sampling rate of EMG signals was at 23.438 kHz with each signal recorded for 11.2 seconds. The recordings were digitized using high and low pass filters with cut-off frequencies of 2Hz and 10 kHz, respectively.

For each subject, the recordings are taken using 1 to 3 muscles, from the ones mentioned before, which may differ from one subject to another.

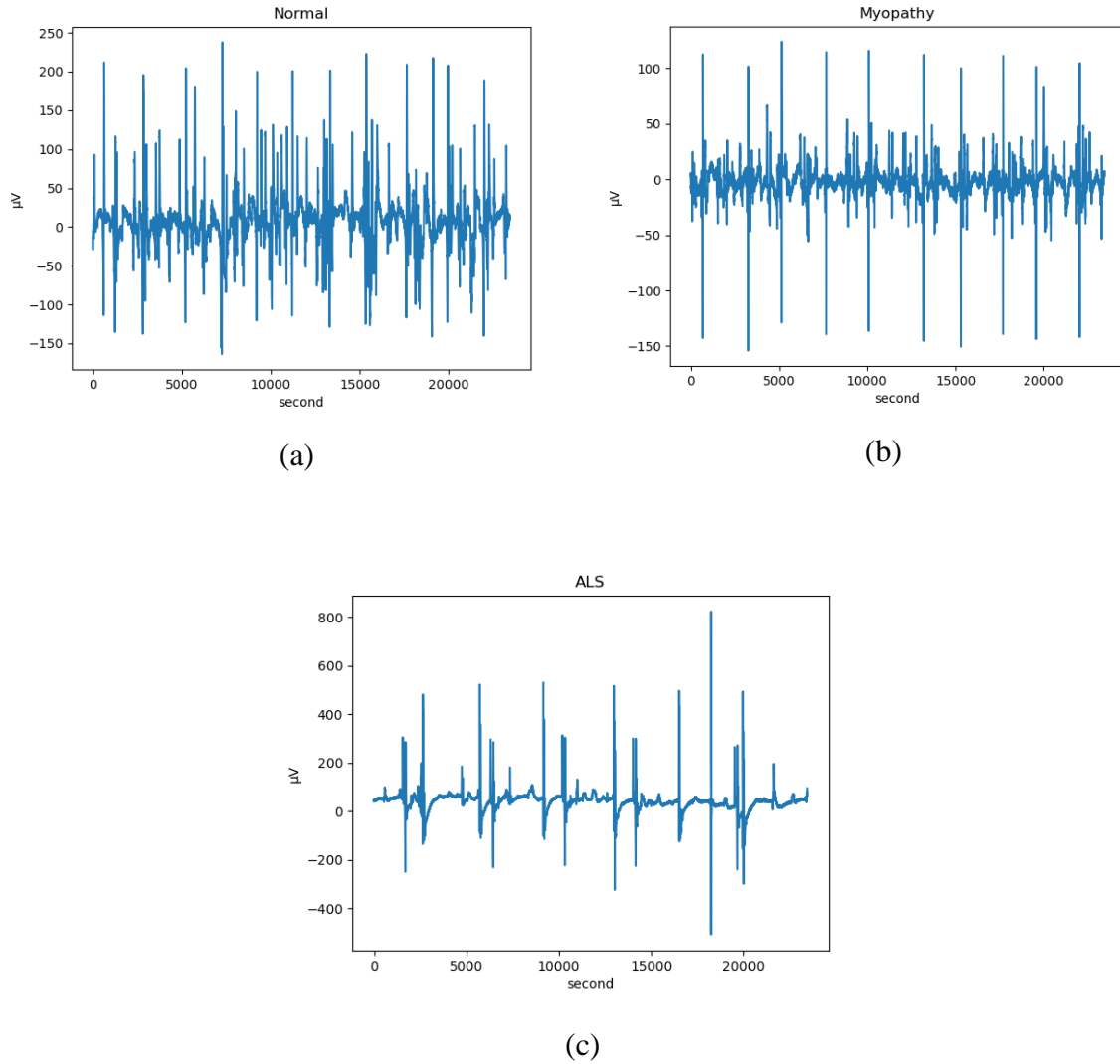


Figure 3.1: EMG signals of (a) Normal (b) Myopathy and (c) ALS subjects.

3.3.2 Data Preparation

a. Muscle selection:

For the data preparation process, we first identified three groups of participants in the study: a normal control group, a group of patients with myopathy, and a group of patients with ALS. The normal control group had EMG signals recorded only from the biceps brachii muscle. Therefore, we selected only the EMG signals recorded from the biceps brachii muscle for the myopathy and ALS groups as well. The decision to focus on the biceps brachii muscle for all

classes was based on practical considerations and the necessity to ensure a fair comparison between ALS, myopathy, and normal signals within the available dataset. While it would have been advantageous to include normal signals from multiple muscle types, the limitation of available data constrained our choice. This selection process resulted in 98 EMG signals from patients with ALS, 107 EMG signals from patients with myopathy, and 270 EMG signals from normal individuals. These statistics provide the basis for subsequent data analysis, allowing for further investigation of differences in muscle function between the normal, myopathy, and ALS groups.

After selecting the EMG signals recorded from the biceps brachii muscle for each group, we observed that the distribution of EMG signals across the three groups was imbalanced, with a higher number of EMG signals from the normal control group compared to the myopathy and ALS groups.

b. Windowing and removing missing values:

Before addressing the class imbalance issue, we preprocessed the dataset by applying a sliding window approach which is a common windowing technique used in signal processing. This method involves shifting a window of a fixed size along the signal with a specified overlap between neighboring windows. The overlap ensures that each data point in the signal is included in multiple windows, reducing the potential impact of distortions or abnormalities in the data that are caused by the abrupt changes in the signal at the start and the end of each window. In our case, we used a window size of 23437 data points and an overlap of 14062 (60%). However, due to the nature of the data, we encountered many windows containing missing or NaN values. To address this, we removed all windows that contained NaN values. After applying the windowing technique and removing the windows with NaN values.

c. Balancing the data:

Moving to the imbalanced dataset issue. We first tried using the undersampling technique to remove some signals from the normal control group. However, this approach did not yield satisfactory results as it significantly reduced the size of the dataset and potentially removed useful information. Subsequently, we explored the use of data augmentation techniques to improve the classification system's performance. One of the techniques we employed was the introduction of noise to the EMG signals. The aim of this approach was to enhance the model's ability to handle real-world scenarios and improve its generalization capabilities. However,

upon evaluating the results, we found that the noise addition did not lead to significant improvements in accuracy.

As a result, we explored an alternative technique known as SMOTE (Synthetic Minority Over-sampling Technique), to create synthetic EMG signals for the myopathy and ALS groups. The main idea behind SMOTE is to generate synthetic examples of the minority class to achieve a balanced class distribution. For each minority class sample, SMOTE identifies its k-nearest neighbors in feature space. It then creates new samples by interpolating between the minority class sample and its k-nearest neighbors. This approach allowed us to balance the dataset by increasing the number of EMG signals for the minority classes while maintaining the original distribution of the data.

SMOTE is applied only to the training data and not the validation and test data since they should represent samples of the data that the model will encounter in the real world, by keeping it untouched, we ensure that it remains a reliable measure of the model's ability to generalize to new, unseen samples.

In contrast, for the train-validation-test split approach, it was unnecessary to generate synthetic data points using SMOTE. The data undersampling technique employed for balancing the data yielded satisfactory results without the need for synthetic data generation. However, in the case of the subject independent method, incorporating SMOTE into the process actually improved the results significantly compared to the approach without synthetic data generation.

It's worth noting that while this technique can be effective for data balancing if the number of samples in the majority class is much larger than the number of samples in the minority class, SMOTE will create many synthetic examples that are very similar to each other, which can lead to overfitting and poor generalization. The general rule of thumb to address this problem is to set the maximum number of synthetic examples to double the number of examples in the minority class. This helps to ensure that the synthetic examples generated by SMOTE are diverse and representative of the minority class, without overfitting the model to the training data.

To mitigate this issue in our particular situation, an additional step was taken to effectively handle the class imbalance. To prevent the overgeneration of synthetic examples, it was necessary to remove some signals or instances from the majority class (Healthy) before applying SMOTE reducing the number of healthy signals from 270 to 192.

Table 3.1: Summary Statistics of the Dataset.

Type of signals	N° Signals			
	Initially	Windowing	Removing NaNs	SMOTE
ALS	98	2548	2511	4665
Healthy	192	4992	4990	4990
Myopathy	107	2782	2772	4886

3.4 Proposed Network Architecture

During the course of this project, we conducted a series of experiments to explore different approaches for improving the classification system. Initially, we trained a 1D CNN model on an undersampled dataset, followed by evaluating the model's performance on an augmented dataset with added noise. Additionally, we experimented with 2D CNN models trained on spectrograms and time-domain plots of the EMG signals. the results of these experiments fell short of our expectations and did not meet the desired level of accuracy. However, when we utilized the SMOTE technique to balance the data and trained an improved 1D CNN model, it demonstrated promising performance. In the upcoming section, we will provide a detailed explanation of this successful 1D CNN model.

To evaluate the effectiveness of our proposed model, we employed two evaluation techniques. First, we utilized the standard train-validation-test split approach to assess the performance of our models. We then implemented a subject-independent evaluation approach where both the test and validation sets are subject-independent, which means that the data used for evaluating and validating the classification system consists of EMG signals from individuals who were not part of the training process. We believe that this last evaluation approach is more realistic, as in the real world we do not dispose of prior patient data. However, to the best of our understanding, most of the previously presented related research papers do not apply such an approach.

3.4.1 CNN Model

Convolutional neural networks with one-dimensional architecture are frequently used to analyze time series data. This enables us to use them on raw EMG signals that vary over time. Our proposed model for classifying EMG signals in neuromuscular diseases was built upon the ALSNet model, which was originally developed by K. M. Naimul Hassan et al. for binary classification of ALS and normal cases [61]. ALSNet demonstrated promising results in its original context, showcasing its effectiveness in distinguishing between ALS and normal cases. However, when we extended the model to our multiclassification task involving ALS, myopathy, and normal cases, we encountered challenges due to the increased complexity and class imbalance. To overcome this limitation, we made several modifications to the model architecture, including adding additional layers, adjusting parameters, and implementing regularization techniques. These modifications were crucial in improving the model's performance and enhancing its ability to accurately classify EMG signals across multiple classes.

In addition to these modifications, we also made a deliberate choice as discussed previously regarding the selection of muscles for each class. While ALSNet utilized signals from both the biceps brachii and vastus medialis muscles for ALS signals, we chose to use signals solely from the biceps brachii muscle for all classes. This decision was made to ensure consistency in our approach and simplify the comparison between ALS, myopathy, and normal signals. While ALSNet served as a valuable starting point, the modifications made in our proposed model were necessary to meet the requirements of our specific task.

For multiclass classification, the model starts with four 1D convolutional layers to extract features from the input data, each having a different filter size and a number of filters. Activation functions like the Rectified Linear Unit (ReLU) are used to introduce non-linearity. Additionally, a dilation rate is specified for each convolutional layer. The dilation rate controls the spacing between the values in the filters, allowing the model to capture information at different scales or receptive fields, while simultaneously keeping the number of parameters relatively low [65]. Moreover, the model's architecture incorporates L2 regularization (weight decay) in the convolutional layers to control overfitting. These layers are followed by batch normalization and dropout layers for regularization and improved generalization. The architecture also includes a Global Max Pooling 1D layer, which effectively reduces the dimensionality of the feature maps obtained from the convolutional layers. Convolutional layers are initialized using the He normal initialization method, which addresses the

vanishing/exploding gradients problem by setting the initial weights of the neural network in a way that helps to mitigate this issue, leading to a more stable and effective training process, enabling the network to acquire valuable insights and representations from the input data.

After the convolutional layers, the model incorporates two fully connected layers. The first fully connected layer consists of 128 neurons and uses the ReLU activation function. This layer is followed by a batch normalization and a dropout layer, which further enhance the model's generalization capabilities. The final layer of the model is a dense layer with 3 neurons, employing the softmax activation function, which outputs a probability for each class. This architecture is represented in **Figure 3.2**.

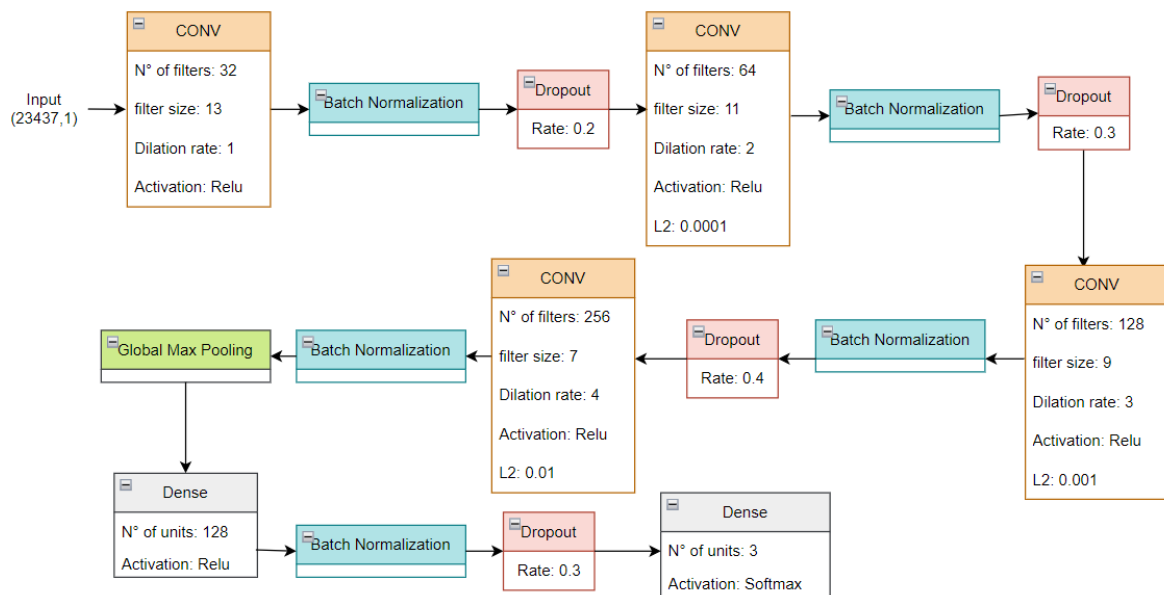


Figure 3.2: 1D CNN model for multiclass classification.

For the task of binary classification, concerning Normal vs. ALS and Normal vs. Myopathy, we employed the same convolutional neural network (CNN) architecture as before with minor modifications. The modification involved changing the number of neurons in the last dense layer to 1, and the activation function was set to 'sigmoid'. The model architecture is shown in **Figure 3.3** below.

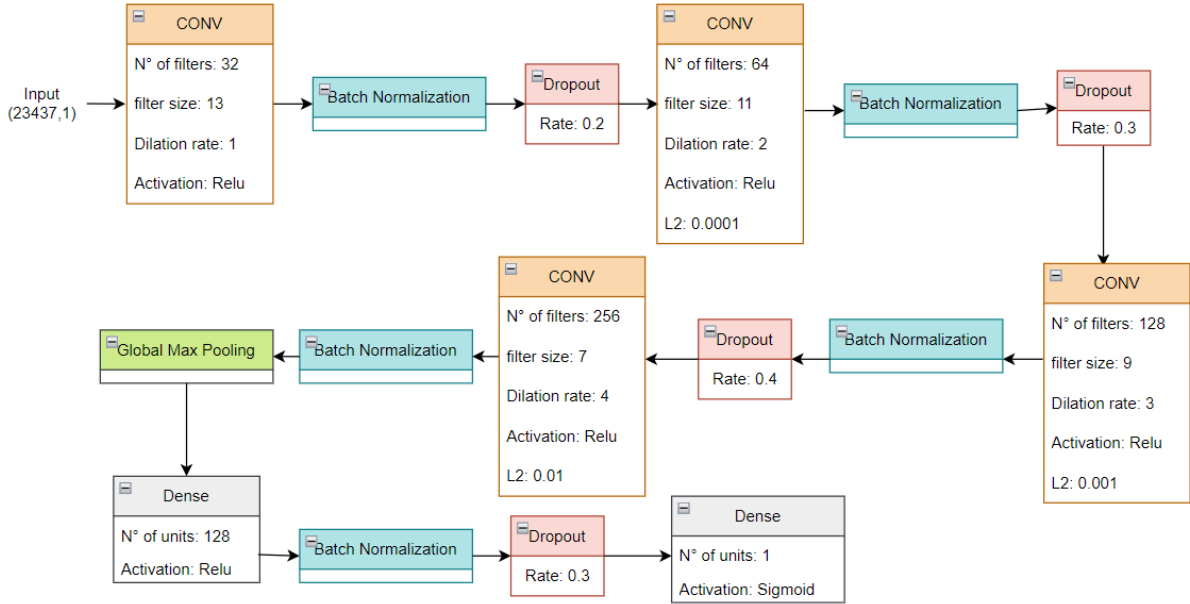


Figure 3.3: 1D CNN model for binary classification.

3.4.2 Training Methodology

The binary classifications followed similar training approaches. We utilized a binary cross-entropy loss function and accuracy metric. We used the Adam optimizer with an initial learning rate of 0.001. The training is performed for a maximum of 100 epochs. We employed early stopping, monitoring the validation loss to stop training if no improvement occurred for 20 epochs for the myopathy vs. normal classification and 25 epochs for the ALS vs. normal classification. Additionally, the learning rate was reduced by a factor of 0.1 if no improvement in the validation loss was observed for a specific number of epochs. We used a batch size of 16 for the ALS vs. normal classification and 32 for the myopathy vs. normal classification.

For the multiclassification task, which involved distinguishing between ALS, myopathy, and normal signals, we utilized a sparse categorical cross-entropy loss function and sparse categorical accuracy metric. The model was trained similarly to the binary classifications, early stopping was implemented with a patience of 25 epochs, and the learning rate was adjusted when the validation loss did not improve for 6 consecutive epochs and the batch size for the multiclassification task was set to 16. Furthermore, to prevent any bias due to the order of the data, the training data is shuffled before each epoch for all the classification tasks. This ensures that the model sees the data in different orders during training, promoting unbiased learning.

We monitored the validation loss since it quantifies how well the model generalizes to new, unseen data. While the model's parameters are updated based on the training data to minimize the training loss, the true measure of its true performance is evaluated using the validation set, which contains data that the model hasn't seen before.

3.5 Results and Discussion

In this section, we present the results of our analysis of the EMG signals dataset using two different validation approaches. We begin by presenting the results of the train-test-validation split approach, followed by the results of the subject-independent evaluation approach. By employing these two approaches, we have taken into account both the overall performance of the model on unseen data through the train-validation-test split, as well as its generalization capabilities across different subjects through the subject-independent validation and testing. This allows for a more comprehensive evaluation of the model's performance and robustness. Finally, we compare the performance of the two approaches and provide insights into the suitability of each approach for analyzing EMG signal data.

3.5.1 Train-Validation-Test Split Approach.

In the train-validation-test split approach, the splitting of the dataset into training, validation, and testing subsets is typically done randomly. Randomization helps ensure that the data in the training, validation, and testing sets are representative of the overall dataset and are not biased toward specific patterns or characteristics.

During the training process, the data was randomly split using the `train_test_split` function, with a test size of 0.25. This means that 25% of the data was reserved for testing, while the remaining 75% was used for training. Additionally, a validation split of 20% was applied to the training data, further dividing it into a training subset and a validation subset.

3.5.1.1 Multiclass Classification

Figure 3.4 illustrates the loss and accuracy for the multiclass classification task on both the training and validation sets. The initial loss is 5.1553, and the initial sparse categorical accuracy is 59.49% on the training set. The initial validation loss is 4.0339, and the initial validation sparse categorical accuracy is 0.4694.

As the training progresses, both the loss and accuracy improve. By the end of the training, the model achieves a training loss of 0.2257 and a training sparse categorical accuracy of 99.88%.

The validation loss decreases to 0.2455, and the validation sparse categorical accuracy is 99.75%. The test accuracy is 99,85%.

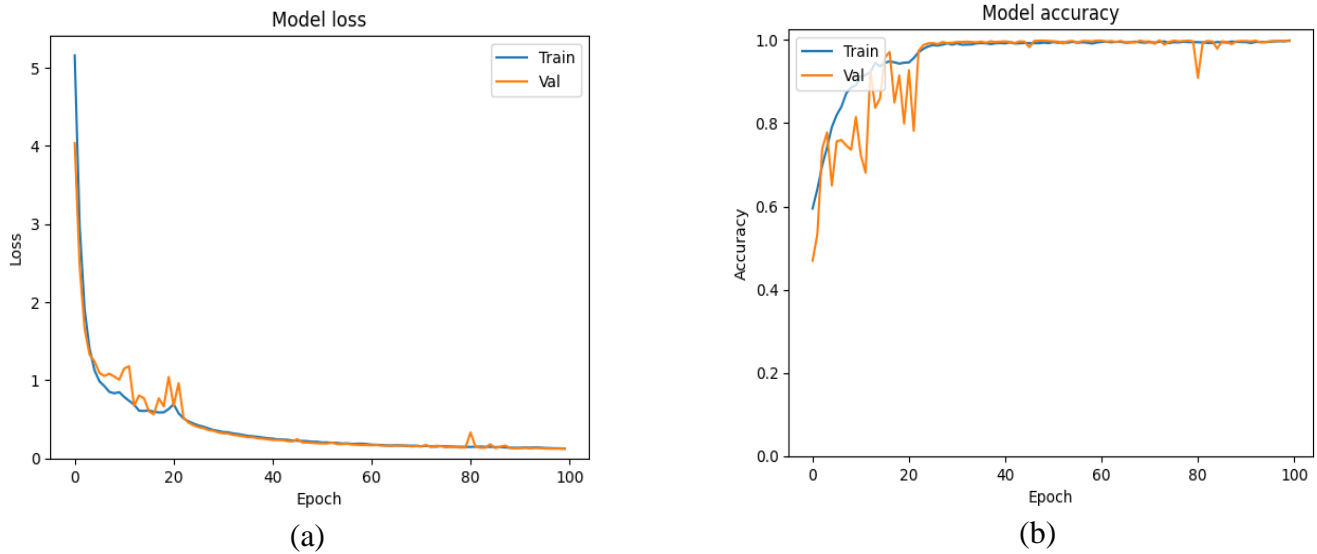


Figure 3.4: Multiclass Classification Train and Validation Plots.

(a) Loss Plots. (b) Accuracy plots.

The classification model's performance was evaluated using a confusion matrix, which provides detailed insights into its predictive capabilities for each class. **Figure 3.5** depicts the confusion matrix of multiclass classification for the train-validation-test split approach.

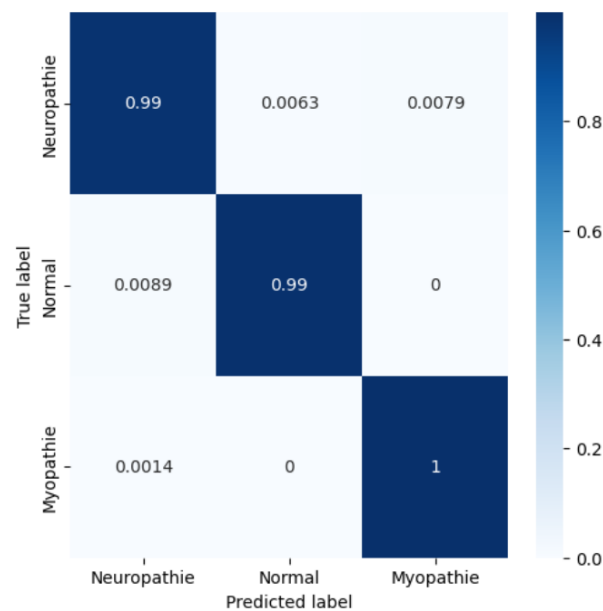


Figure 3.5: Multiclass Classification Confusion Matrix.

Additionally, the performance of the classification model was evaluated using several metrics, including precision, recall, and F1 score. These metrics provide valuable insights into the model's ability to correctly classify instances across different classes.

The precision metric measures the proportion of correctly predicted positive instances out of all instances predicted as positive. In our evaluation, the model achieved a precision score of 0.99 for all classes. This indicates that all instances classified as positive were indeed true positives.

The recall metric, also known as sensitivity or true positive rate, measures the proportion of correctly predicted positive instances out of all actual positive instances. Our model achieved a recall score of 1.00 for all classes, indicating that it successfully captured all positive instances.

The F1 score, which combines both precision and recall, provides a balanced measure of the model's performance. With an F1 score of 1.00 for all classes, our model demonstrates excellent accuracy and robustness in classifying instances across the board.

Table 3.2: Multiclass Classification CNN Model Performance metrics
Train-Test Split Approach.

Class	Precision	Recall	F1 score
ALS	0.99	0.99	0.99
Normal	0.99	0.99	0.99
Myopathy	0.99	1.00	1.00

3.5.1.2 Binary Classification

Turning our attention to the domain of binary classification, our model was designed to differentiate between two distinct classes.

First, we present the results of the Myopathy vs. Normal classification. The training process spanned a total of 82 epochs. Throughout these epochs, the model exhibited a progressive

improvement in performance, as evidenced by the reduction in the loss value and the increase in accuracy.

In **Figure 3.6**, we present the train and validation plots. **Figure 3.6 (a)** displays the loss plots, while **Figure 3.6 (b)** showcases the accuracy plots.

During the initial stage of training, the model achieved an accuracy of 74.84% on the training set and 80.60% on the validation set. The corresponding loss values were 3.4419 for the training set and 1.8283 for the validation set. Throughout the training, the graph in **Figure 3.6 (b)** shows that the training accuracy is increasing steadily, while the validation accuracy shows significant fluctuations. This is a classic sign of overfitting. Overfitting occurs when the model learns the training data too well, and as a result, it does not generalize well to new data. One of the reasons why we might not have been able to completely prevent overfitting, even with regularization is that the data for class myopathy is not large enough. If the dataset is too small, the model will not have enough information to learn from, and it will be more likely to overfit the training data. At the end of the training process, the accuracy of the training set reached a score of 98.91%, indicating that the model successfully classified 98.91% of the training examples correctly. Similarly, the accuracy of the validation set reached 91.72%. Furthermore, the loss values decreased significantly. The loss on the training set reached a value of 0.1561, indicating that the model's predictions were very close to the actual labels in the training data. Similarly, the loss on the validation set reached 0.2931. The test accuracy was 92.47%.

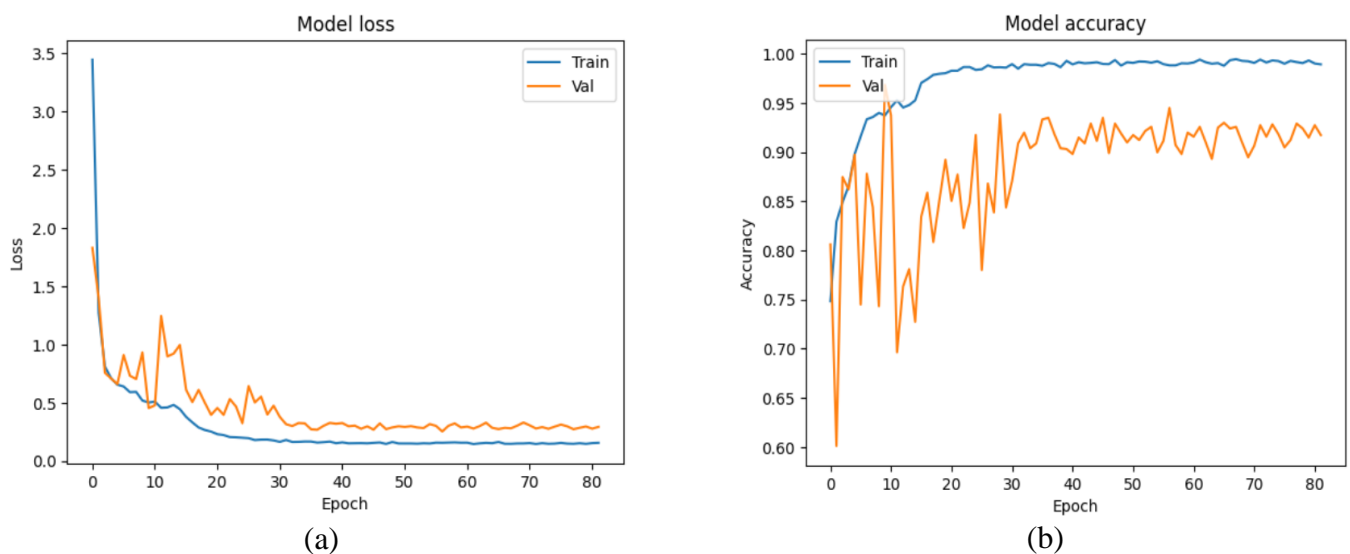


Figure 3.6: Train and Validation Plots for Binary Classification (Myopathy vs. Normal).

(a) Loss Plots. (b) Accuracy Plots.

The performance of the classification model was assessed using a confusion matrix, shown below in **Figure 3.7**. The latter offers detailed insights into its ability to predict accurately for each class.

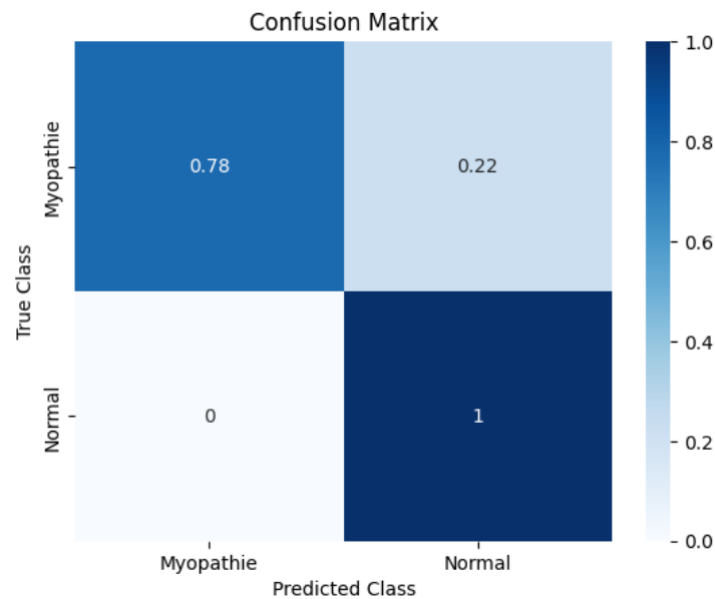


Figure 3.7: Confusion Matrix of the CNN Model for Binary Classification (Myopathy vs. Normal) using Train-Test Split Approach.

The Performance Matrix below, presented in **Table 3.3**, indicates that the classification model demonstrates strong performance in correctly identifying instances from class Normal, as evidenced by the high precision and recall values which are 0.9 and 1 respectively. Nonetheless, there is room for improvement in accurately capturing instances from class Myopathy, as indicated by the lower recall of 0.78.

Table 3.3: Performance metrics for Binary Classification (Myopathy vs. Normal) Train-Test Split Approach.

Class	Precision	Recall	F1 score
Myopathy	1.00	0.78	0.88
Normal	0.90	1.00	0.95

Furthermore, we provide an overview of the ALS vs. Normal classification results. The training process encompassed a comprehensive series of 67 epochs. During this iterative training phase, the model consistently demonstrated incremental advancements in its performance, as denoted by the continuous decrease in the loss metric and the concurrent rise in accuracy.

Throughout these 67 epochs, the model's performance steadily improved as indicated by the decrease in the loss metric. The training loss decreased from an initial value of 3.9028 to a final value of 0.1542. Additionally, the training sparse categorical accuracy increased from 75.27% to 98.73% during training. The validation loss decreased from 2.8310 to 0.1878, and the validation sparse categorical accuracy improved from 68.53% to 96.80% as the training progressed. The test accuracy is 96.06%. The loss and accuracy plots for both the training and validation are graphically represented in **Figure 3.8**.

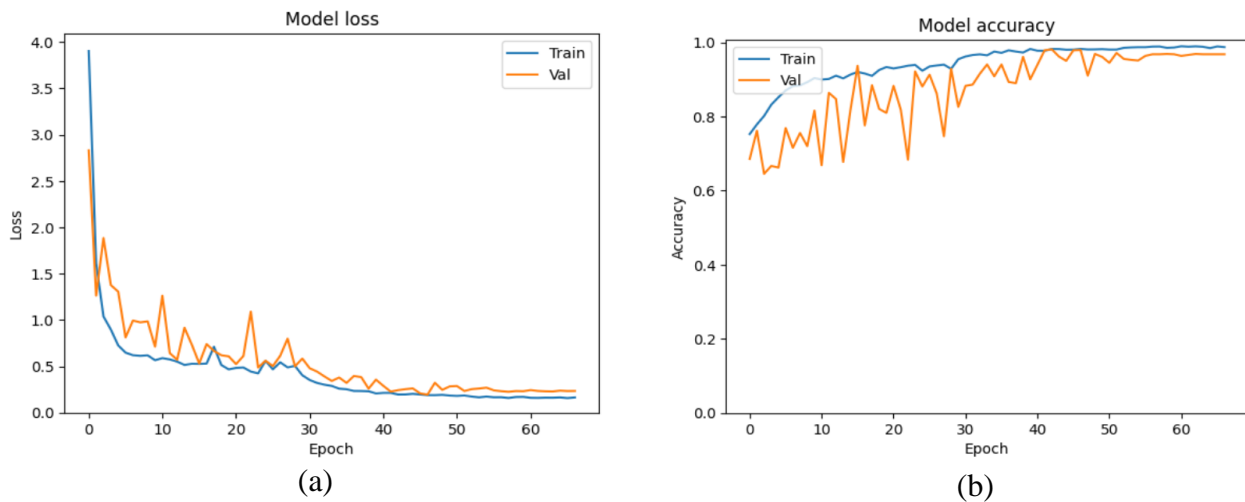


Figure 3.8: Binary Classification (ALS vs. Normal) Train and Validation Plots.

(a) Loss Plots. (b)Accuracy Plots.

Figure 3.9 presents the confusion matrix for the binary classification task of ALS versus Normal using the train-validation-test split Approach. The confusion matrix provides a visual representation of the model's performance by illustrating the distribution of predicted and true labels.

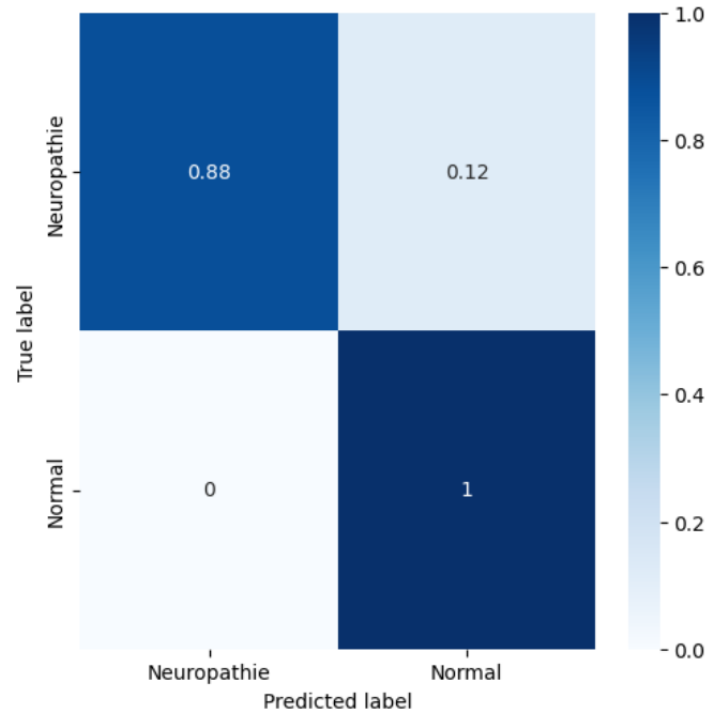


Figure 3.9: Confusion Matrix for Binary Classification (ALS vs. Normal) using Train-Test Split Approach.

The model's performance in classifying ALS and Normal cases can be evaluated by referring to **Table 3.4**, which presents a comprehensive set of performance metrics, including precision, recall, and F1-score for each class.

Table 3.4: Performance metrics for Binary Classification (ALS vs. Normal) using Train-Test Split Approach.

Class	Precision	Recall	F1 score
Normal	1.00	0.88	0.94
ALS	0.94	1.00	0.97

Table 3.5 presents a summary of the accuracies attained through the train-validation-test split approach for three distinct classifications: multiclass classification, Normal vs. ALS classification, Normal vs. Myopathy classification.

Table 3.5: Summary of Evaluation Metrics for the Train-Test Split Approach.

Classification Task	N° epochs	Train Time (min)	Train acc (%)	Val acc (%)	Test acc (%)	Avg F1 score
Normal vs. ALS vs. Myopathy	100	202	99,88%	99.75%	99,85%	0.993
Normal vs. Myopathy	82	162	98.91%	91.72%	92.47%	0.915
Normal vs. ALS	67	132	98.73%	96.80%	96.06%	0.955

The exceptionally high scores achieved through the implementation of the train-validation-test split evaluation method can be attributed to the unintended data leakage caused by the random splitting approach. This unintended data leakage led to the inclusion of records from the same subjects across different sets, ultimately inflating the performance metrics.

3.5.2 Subject-Independent Evaluation Approach

In this approach, we performed both multiclass (Normal vs. ALS vs. Myopathy) and binary classification (Normal vs. Myopathy and Normal vs. ALS). For each experiment, from each class, we excluded one patient for validation and another patient for testing, which is only used to evaluate the final performance of the trained model after training is completed. These patients were also removed from the training set so that the model could be validated and tested on completely unseen data.

We conducted a series of experiments for each classification task, where the subjects in the test and validation sets were changed for each experiment. By varying the subjects, we ensured a robust evaluation of our classification system's performance across different individuals.

The results of each experiment will be presented in dedicated tables (**Table 3.6**, **Table 3.9**, and **Table 3.12**), providing a comprehensive analysis and a clear overview of the performance metrics achieved for each classification task. Additionally, we will showcase the graphs and metrics for both the best and worst experiments conducted for each classification task.

In the tables: **Table 3.6**, **Table 3.9**, and **Table 3.12**, each subject ID corresponds to three subjects, representing one subject from each class (ALS, Normal, Myopathy).

We collected information regarding the number of epochs, the training time, the accuracies, and the average F1 score for classes of each experiment.

3.5.2.1 Multiclass Classification

Table 3.6: Performance Metrics for Multiclass Classification for Different Patients.

Subjects ID		N° epochs	Train Time (min)	Train acc (%)	Val acc (%)	Test acc (%)	Avg F1 score
Val	Test						
4	3	30	125	98.07 %	56.08 %	77.92 %	0.78
3	4	39	173	98.32 %	77.33 %	56.33 %	0.53
6	3	47	201	98.28 %	73.17 %	68.41 %	0.68
3	6	82	351	97.70 %	71.34 %	68.70 %	0.67
7	3	74	374	98.77 %	86.36 %	79.39 %	0.80
3	7	51	240	98.74 %	75.04 %	80.59 %	0.81
Average		54	244	98.31 %	73.22 %	71.89 %	0.71

From the table, we observe varying performance across different subject IDs. The training accuracies range from 97.70% to 98.77%, indicating that the models effectively learn the training data. However, there is more variation in the validation and test accuracies, ranging from 56.08% to 86.36% and 56.33% to 80.59%, respectively. The validation accuracy for the first experiment and test accuracy for the second experiment are the lowest and are both linked to subject 4 from each class. This indicates that our model faces challenges in accurately classifying those particular subjects.

The average F1 score ranges from 0.477 to 0.82, suggesting different levels of precision and recall across the subject IDs.

The number of epochs used for training varies between 30 and 82, and the training time spans from 125 to 374 minutes.

As can be observed from **Table 3.6**, the experiment with subject 3 for validation and subject 4 from each class for the test yielded the lowest test accuracy. Therefore, it is considered the least effective experiment for multiclass classification. The loss and accuracy plots for that experiment are illustrated in **Figure 3.10**.

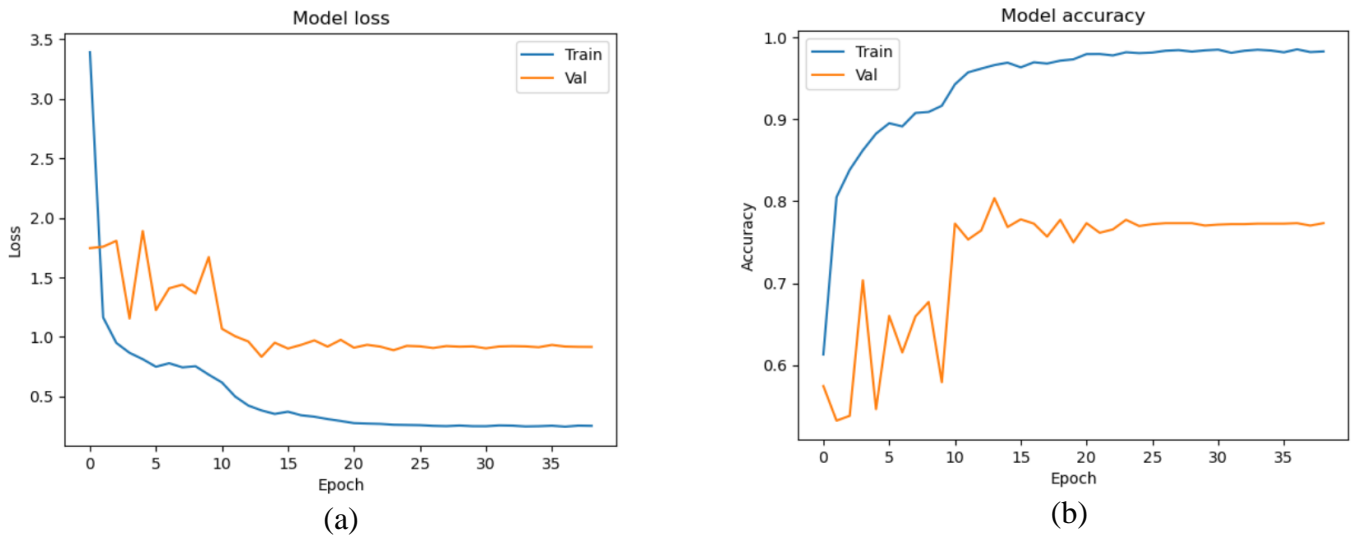


Figure 3.10: Train and Validation Plots for Multiclass Classification Subject Independent Evaluation Method -worst result-.

(a) Loss Plots. (b)Accuracy Plots.

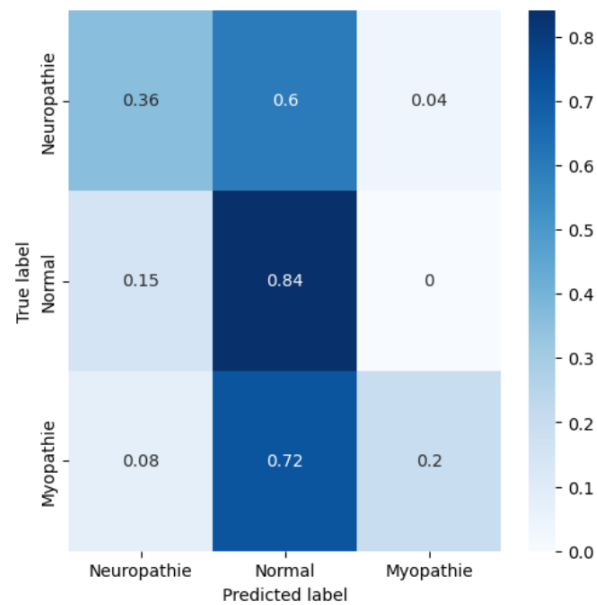


Figure 3.11: Confusion Matrix for Multiclass Classification Subject Independent Evaluation Approach -worst result-.

Figure 3.11 presents the confusion matrix for the multiclass classification using the subject independent evaluation approach. Proportions of correctly predicted labels for the ALS, Normal, and Myopathy classes were 0.36, 0.84, and 0.2, respectively. This suggests that the

model performed relatively better in classifying instances as Normal while facing challenges in accurately identifying cases of ALS and Myopathy.

Table 3.7: Multiclass Classification CNN Model Performance metrics
Subject Independent Evaluation Approach -worst result-.

Class	Precision	Recall	F1 score
ALS	0.57	0.36	0.44
Normal	0.55	0.84	0.67
Myopathy	0.73	0.20	0.32

Alternatively, based on the information provided in **Table 3.6**, it is noteworthy that the highest test accuracy of 80.59% was achieved in the experiment where patient 3 was utilized for validation, and patient 7 was employed for the test. Complementing these results, the accompanying loss and validation graphs for this specific experiment have been graphically illustrated in **Figure 3.12**.

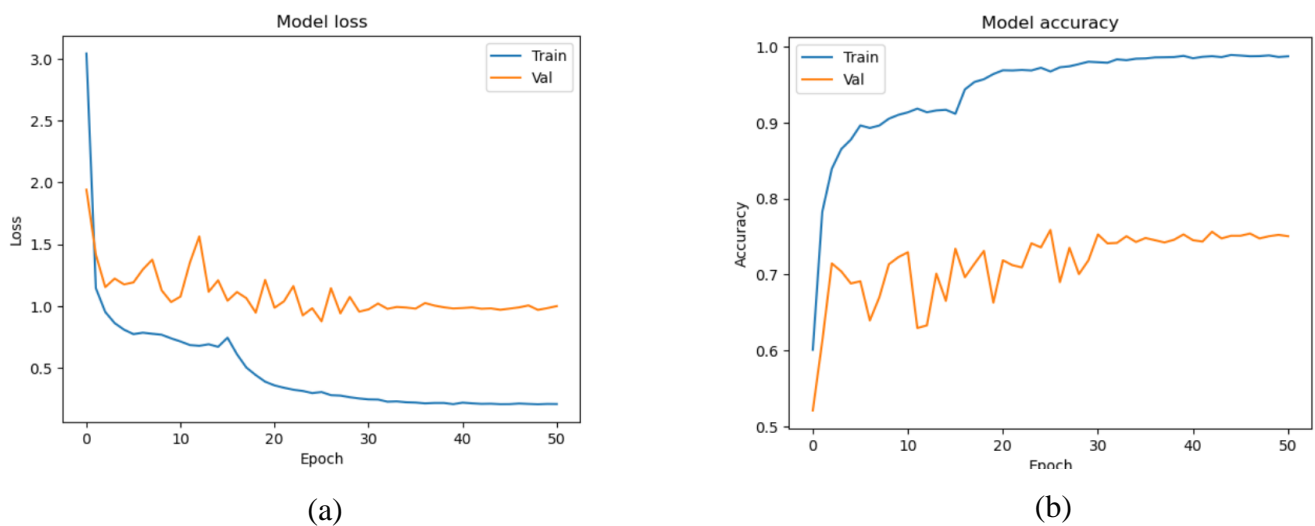


Figure 3.12: Train and Validation Plots for Multiclass Classification Subject Independent Evaluation Method -best result-.

(a) Loss Plots. (b)Accuracy Plots.

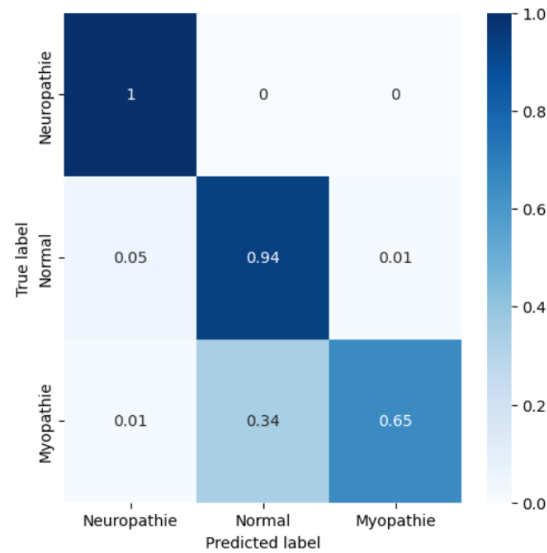


Figure 3.13: Confusion Matrix for Multiclass Classification Subject Independent Evaluation Method -best result-.

The results presented in **Figure 3.13** showcase proportions of correctly predicted labels for the ALS, Normal, and Myopathy classes were 1.00, 0.94, and 0.65, respectively. This suggests that the model performed exceptionally well in classifying ALS cases, with perfect precision. The model also performed well in classifying Normal cases, with high precision. The model achieved a moderate precision level for Myopathy cases. Overall, the obtained proportions in addition to the results illustrated in **Table 3.8** indicate that the model is effective in accurately identifying instances across multiple classes, with a notable emphasis on the precise classification of ALS cases.

Table 3.8: Multiclass Classification CNN Model Performance metrics Subject Independent Evaluation Approach -best result-.

Class	Precision	Recall	F1 score
ALS	0.92	1.00	0.95
Normal	0.60	0.94	0.73
Myopathy	0.99	0.65	0.78

3.5.2.2 Binary Classification-Myopathy vs. Normal

Table 3.9: Performance Metrics for Binary Classification (Myopathy vs. Normal) for Different Patients.

Subjects ID		N° epochs	Train Time (min)	Train acc (%)	Val acc (%)	Test acc (%)	Avg F1 score
Val	Test						
4	3	24	77	99.67 %	79.58 %	85.15 %	0.85
3	4	34	109	99.59 %	78.85 %	77.2 %	0.73
6	3	50	162	99.72 %	92.14 %	77.78 %	0.76
3	6	25	80	99.81 %	73.29 %	93.23 %	0.93
7	3	25	41	99.51 %	69.60 %	74.15 %	0.72
3	7	34	109	99.60 %	72.97 %	69.67 %	0.64
Average		32	96	99.65 %	77.74 %	79.53 %	0.77

Table 3.9 provides a summary of the scores obtained during the training of our model on myopathy vs. normal binary classification on various subjects. Notably, the training accuracy consistently demonstrates high values across all experiments. However, there exists considerable variability in the validation and test accuracies, encompassing a broad spectrum of scores. Conversely, the majority of experiments yield high F1 scores; nevertheless, some instances exhibit lower values, notably as low as 0.615. The obtained results suggest that the models exhibit satisfactory performance when applied to unseen data. However, it is worth noting that certain patients exhibit more challenging diagnostic scenarios than others.

The experiment with subject 3 for validation and subject 7 for the test achieved the lowest test accuracy. This experiment is therefore considered the least effective for binary classification (Myopathy vs. Normal). The loss and accuracy plots for this experiment are presented in **Figure 3.14**.

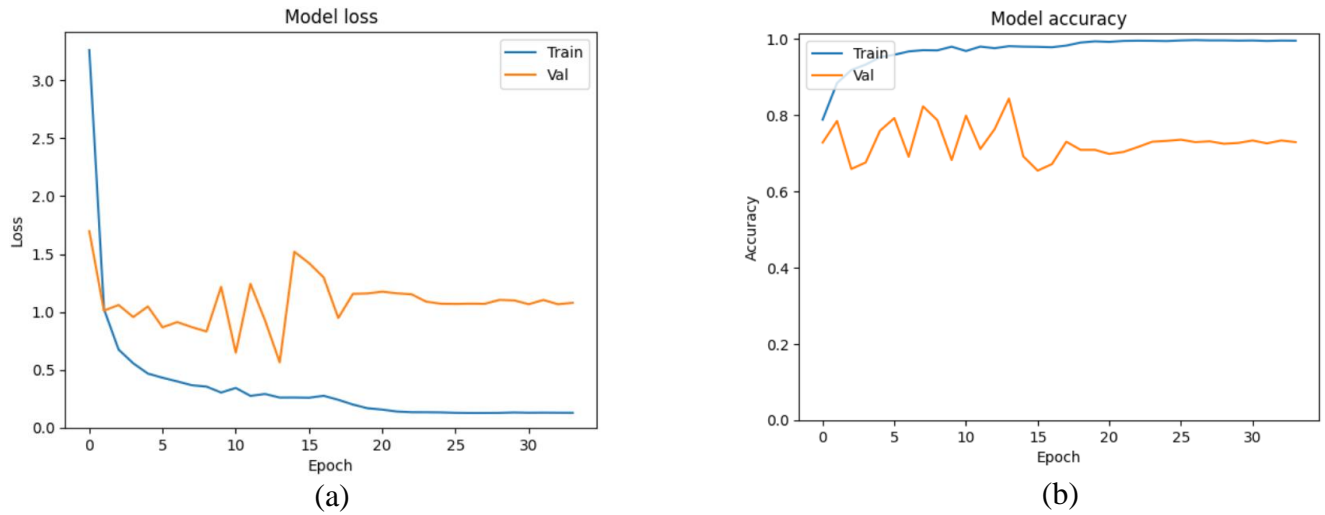


Figure 3.14: Train and Validation Plots for Binary Classification (Myopathy vs. Normal)

Subject Independent Evaluation Method -worst result-.

(a) Loss Plots. (b)Accuracy Plots.

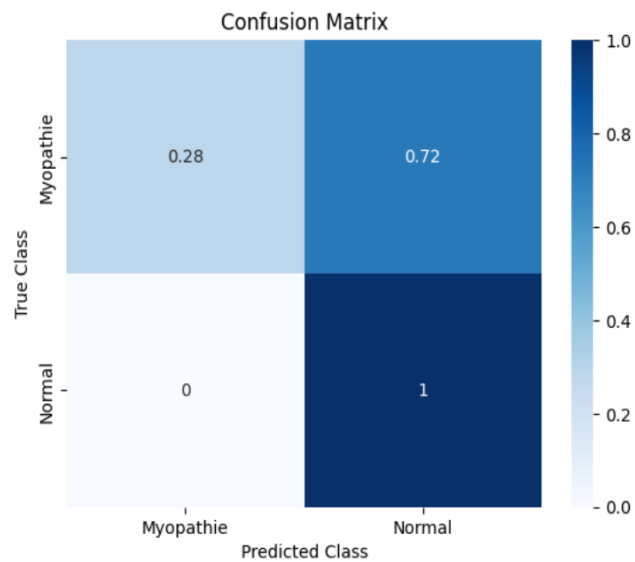


Figure 3.15: Confusion Matrix for Binary Classification (Myopathy vs. Normal) Subject

Independent Evaluation Approach -worst result-.

Table 3.10: Performance metrics for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -worst result-.

Class	Precision	Recall	F1 score
Normal	0.66	1.00	0.79
Myopathy	1.00	0.28	0.44

The Confusion matrix illustrated in **Figure 3.15** and **Table 3.10** above demonstrates that the model has high precision and recall for Normal cases, but it has minimal scores for Myopathy cases.

Although the model exhibited limited precision and recall for Myopathy cases, superior results were achieved in other experiments. Notably, the experiment utilizing subject 3 for validation and subject 6 for testing yielded an outstanding test accuracy of 93.23%. Detailed loss and accuracy plots depicting these results are presented in **Figure 3.16**.

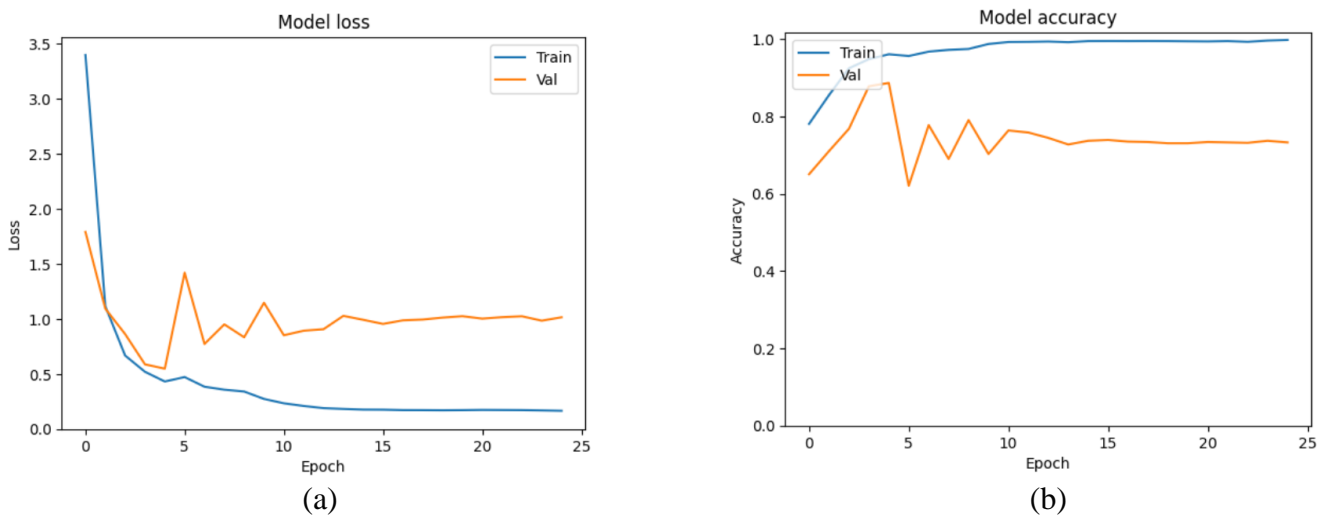


Figure 3.16: Train and Validation Plots for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Method -best result-.

(a) Loss Plots. (b)Accuracy Plots.

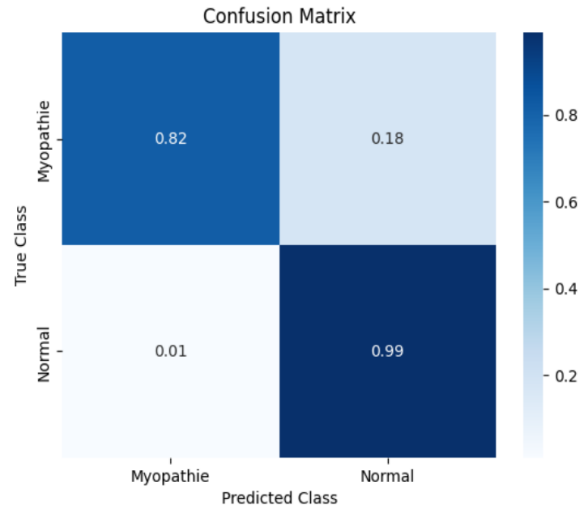


Figure 3.17: Confusion Matrix for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -best result-.

Table 3.11: Performance metrics for Binary Classification (Myopathy vs. Normal) Subject Independent Evaluation Approach -best result-.

Class	Precision	Recall	F1 score
Normal	0.91	0.99	0.95
Myopathy	0.99	0.82	0.89

Overall, the metrics in **Figure 3.17** and **Table 3.11** indicate a high precision and recall for the Normal class, showcasing the model's effectiveness in identifying instances accurately. The Myopathy class also exhibits a high precision but comparatively lower recall, indicating some challenges in correctly capturing all instances of Myopathy. Nevertheless, the F1 score for both classes demonstrates reasonably good performance, considering the balance between precision and recall.

The poor scores in classifying the Myopathy signals are likely due to the lack of myopathy signals in our dataset. The SMOTE technique was used to oversample the Myopathy signals, however, the use of synthetic data is not as effective as utilizing real signal data. To improve

the model's performance in classifying Myopathy signals, it is necessary to collect more Myopathy signals, as both Undersampling and oversampling techniques used to balance the dataset did not yield results as good as the ones obtained for Normal signals classification.

3.5.2.3 Binary Classification-ALS vs. Normal

Table 3.12: Performance Metrics for Binary Classification (ALS vs. Normal) for Different Patients.

Subjects ID		N° epochs	Train Time (min)	Train acc (%)	Val acc (%)	Test acc (%)	Avg F1 score
Val	Test						
4	3	43	126	98.97 %	68.15 %	91.84 %	0.92
3	4	55	155	98.85 %	86.4 %	69.69 %	0.68
6	3	34	97	98.97 %	68.77 %	87.26 %	0.87
3	6	32	91	98.50 %	83.37 %	77.89 %	0.76
7	3	58	187	99.15 %	94.93 %	86.4 %	0.86
3	7	47	159	98.99 %	88.03 %	97.03 %	0.97
Average		45	136	98.91 %	81.61 %	85.02 %	0.84

Concerning the binary classification ALS vs. Normal, we got suitable results. The training accuracy across all experiments is consistently high, averaging at 98.91%. This indicates that the models demonstrate proficiency in extracting meaningful patterns from the training data and successfully adapting to the training set. The validation and test accuracies show some variation across the experiments. The validation accuracy ranges from 68.15% to 94.93%, with an average of 81.61%. The test accuracy ranges from 69.69% to 97.03%, with an average of 85.02%. The average F1 scores range from 0.655 to 0.97. A score of 0.655 suggests that the model's precision and recall are moderately balanced, but there is room for improvement. On the other hand, a score of 0.97 indicates a high level of accuracy and precision in the model's predictions, implying that it performs well in both correctly identifying positive instances and avoiding false positives. These results indicate that the models perform reasonably well on unseen data, although there are some patients harder to diagnose than others.

We notice that for this classification (ALS vs. Normal) even the worst result has a better test accuracy than the previous binary classification. The lowest test accuracy of 69.69 % was recorded during the experiment with subject 3 for validation and subject 4 for test. The training and validation loss and accuracy plots are plotted below in **Figure 3.18**.

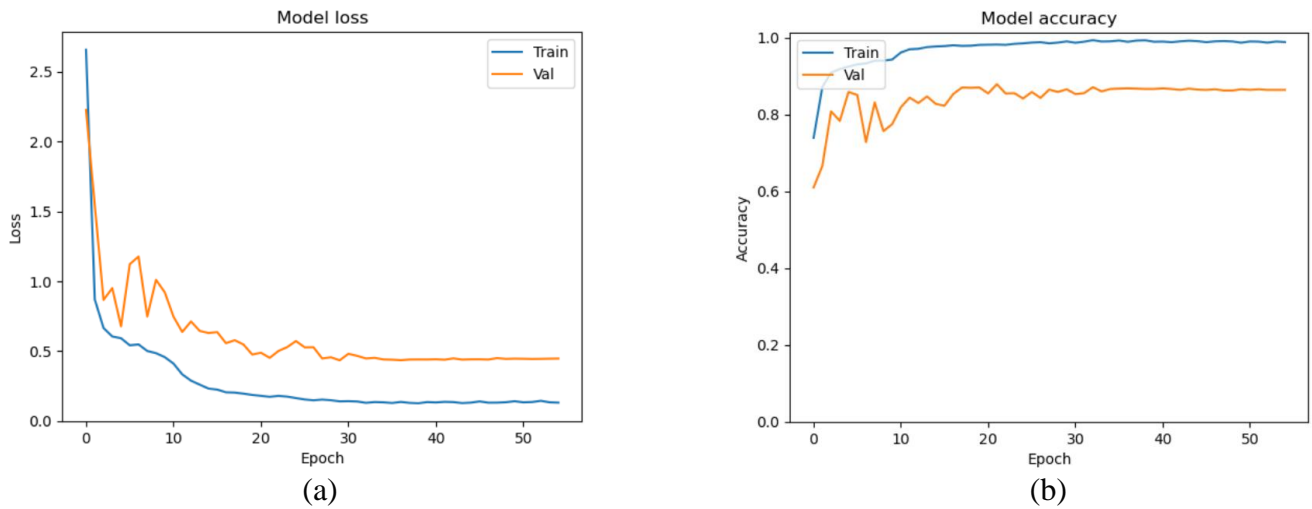


Figure 3.18: Train and Validation Plots for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Method -worst result-.
(a) Loss Plots. (b)Accuracy Plots.

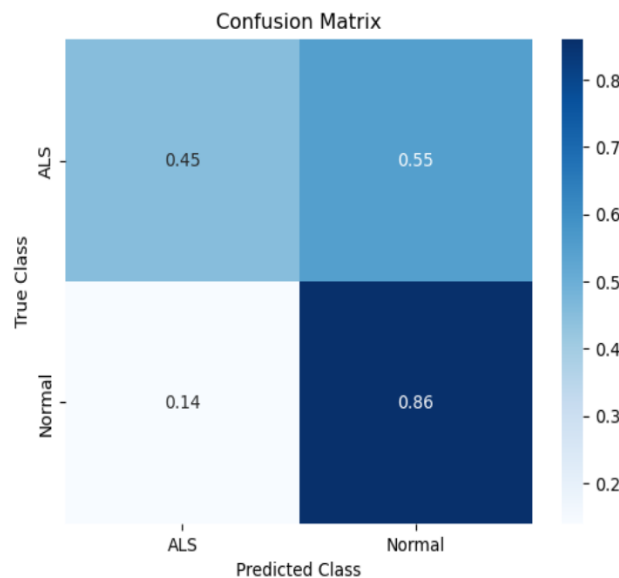


Figure 3.19: Confusion Matrix for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -worst result-.

Table 3.13: Performance metrics for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -worst result-.

Class	Precision	Recall	F1 score
Normal	0.70	0.86	0.77
ALS	0.68	0.45	0.54

Table 3.13 presents the performance metrics for the binary classification of ALS vs. Normal cases using the subject-independent evaluation approach. The precision for the Normal class is 0.70, indicating that 70% of the predicted Normal cases were correctly classified. The recall for the Normal class is 0.86, suggesting that 86% of the actual Normal cases were accurately identified. The F1 score, which combines precision and recall, is 0.77 for the Normal class. For the ALS class, the precision is 0.68, the recall is 0.45, and the F1 score is 0.54, indicating comparatively lower performance in predicting ALS cases.

In contrast, the experiment held on subject 3 for validation and subject 7 for test resulted in a test accuracy of 97.03%, which signifies a high level of performance. This high test accuracy indicates the model's effectiveness in making accurate predictions and demonstrates its capability to generalize well to unseen data. The loss and accuracy plots are represented below in **Figure 3.20**.

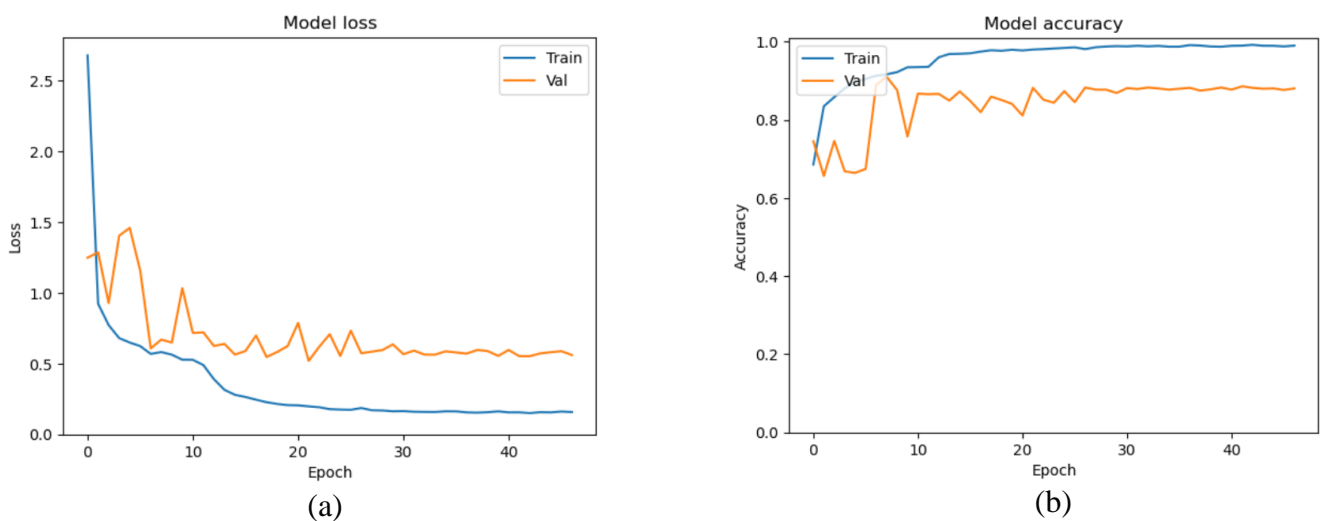


Figure 3.20: Train and Validation Plots for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Method -best result-.

(a) Loss Plots. (b) Accuracy Plots.

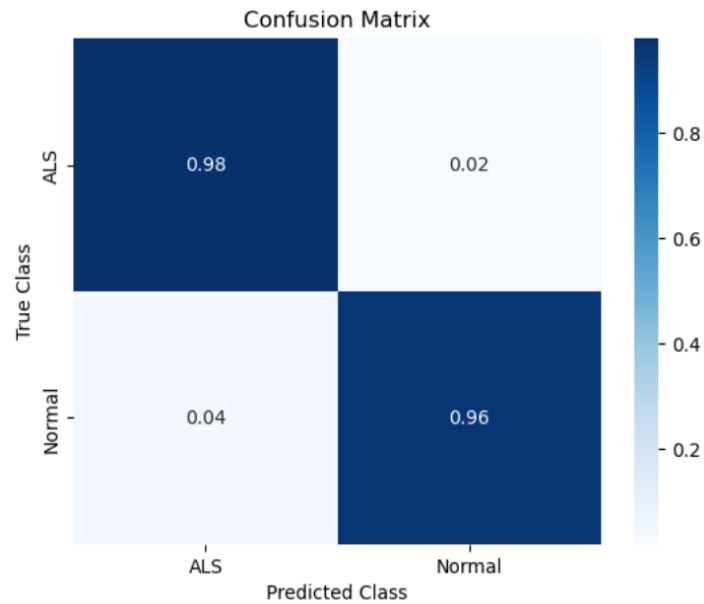


Figure 3.21: Confusion Matrix for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -best result-.

Table 3.14: Performance metrics for Binary Classification (ALS vs. Normal) Subject Independent Evaluation Approach -best result-.

Class	Precision	Recall	F1 score
Normal	0.98	0.96	0.97
ALS	0.96	0.98	0.97

The metrics represented in **Figure 3.22** and **Table 3.13** confirm that the proposed model works well in classifying the signals. The precision, recall, and F1 scores for both classes are excellent, indicating that the model is correctly identifying both Normal and ALS cases with a high degree of accuracy.

All in all, taking all accuracy graphs presented in this part, it is worth noting that all graphs of the subject-independent evaluation method show overfitting. This is likely due to the lack of data, as each time we remove a subject from each class for the validation set and another one for the test set, this leaves a small training set and a small validation set. This can lead to overfitting, as the model may learn the training set too well and not generalize well to new data. Even though we used regularization techniques (Dropout, L2, batch normalization), we were not able to completely prevent overfitting. This suggests that the data set is too small to train a model that can generalize well to new data. To address this issue, we could try to increase the size of the data set by collecting more data.

3.6 Comparison

Table 3.15 provides a comprehensive comparison of our work with four relevant studies in the field of EMG signal classification for neuromuscular diseases.

Using the subject-independent evaluation method, our model's best-achieved accuracy is 97.03% in classifying ALS vs. Normal, 93.23% in classifying Myopathy vs. Normal, and 80.59% in classifying ALS vs. Myopathy vs. Normal. These results demonstrated the effectiveness of our model in accurately classifying EMG signals when tested on completely unseen data, indicating its robustness in real-life scenarios.

Furthermore, we evaluated our model using the train-validation-test split approach. The results showed that our model achieved an accuracy of 96.06% in classifying ALS vs. Normal, and 92.47% in classifying Myopathy vs. Normal, and 99.85% in classifying ALS vs. Myopathy vs. Normal. These findings reinforced the reliability and effectiveness of our model in classifying neuromuscular diseases when trained and tested on specific data partitions.

While comparing our work with existing studies, it is important to acknowledge the complexities involved in comparing results across studies, given the variations in datasets, methodologies, and evaluation techniques. While direct comparisons may be challenging, our model demonstrated superior performance in the train-validation-test approach, surpassing the results reported in the referenced works. Additionally, it exhibited comparable or even higher performance in the subject-independent evaluation, further emphasizing the effectiveness of our approach when compared to existing studies.

Overall, our study contributes a robust deep-learning model for EMG signal classification and showcases its strong performance under both subject-independent and train-validation-test split evaluation methods. These findings reinforce the potential of our model to assist in the accurate

detection and classification of neuromuscular diseases, offering valuable insights for clinical applications.

Table 3.15: Performance comparison with other methods.

Ref.	Method	Evaluation Method	Classification Task	Test Acc (%)	Dataset
[56]	STFT and CNN	Train-Val-Test split	AH	96.69%	[5]
[57]	AR, RMS, ZC, WL, MAV, and MLP	Train-Val-Test split	AMH MH AH	86.3% 81% 80.5%	[58]
[55]	CNN	Train-Val-Test split	AMH	92.5%	NS
[61]	CNN	Subject Independent	AH	97.74 %	[5]
Ours	CNN	Train-Val-Test split	AMH AH MH	99.85% 96.06% 92.47%	[5]
Ours	CNN	Subject Independent	AMH AH MH	80.59% 97.03% 93.23%	[5]

Notes: A = ALS, H = Healthy, M = Myopathy, NS = Not Specified

3.7 Conclusion

In this chapter, we have presented the model we developed along with the results of our study. We found that the model was able to achieve high accuracy on the test data, suggesting that it has learned to extract meaningful features from the data. We also found that the model was able to generalize quite well to unseen data, as evidenced by its performance on the subject-independent evaluation method.

We believe that these results are promising and suggest that the model has the potential to be a valuable tool for a variety of tasks. However, we also recognize that there is still more work to be done. In particular, we would like to investigate ways to improve the model's performance on the minority classes especially for myopathy.

General Conclusion

In this report, we have outlined the main steps in designing and performing an EMG classification system using deep learning models for Neuromuscular Diseases Detection. Starting from the acquisition method of EMG signals to the process of training and evaluating the deep learning model for classification.

The proposed work aims to leverage deep learning techniques to ensure high accuracy in the classification of EMG signals for Neuromuscular disease detection. The publically available N2001 EMGLAB dataset was used to train and evaluate a 1D-CNN model we developed based on ALSNet. The model was evaluated using two methods: train-validation-test split, widely used in research papers, and subject-independent evaluation, which allows testing on new data unseen during the training process. The results showed that the model achieved high accuracy on both multiclass and binary classifications. The model was also able to generalize well to unseen data, although some patients were more difficult to diagnose than others. We are confident that with further research, this model will have the potential to make a real-world impact.

Future directions in this field include the development of multi-modal approaches that combine EMG signals with other types of data such as clinical, imaging, or genetic data; this could improve the accuracy and reliability of the classification of neuromuscular disorders using EMG signals. The development of accurate and reliable classification models could have significant clinical implications, such as early detection and diagnosis of neuromuscular disorders, personalized treatment, and monitoring of disease progression.

References

- [1] Diagnosis and treatment of myasthenia gravis - PubMed." PubMed, 1 Nov. 2014, <https://pubmed.ncbi.nlm.nih.gov/31385879/>. Accessed 12 June 2023.
- [2] D. J. Weidner, Ed., *Electromyography: Physiology, Engineering, and Noninvasive Applications*, 2nd ed. New York: Wiley-Liss, 2002.
- [3] J. A. Sorenson and W. H. Gold, Eds., "Magnetic Resonance Imaging: Physical Principles and Clinical Applications," 3rd ed. Philadelphia: Lippincott Williams & Wilkins, 2009.
- [4] M. S. Islam, M. S. Hossain and M. A. Hossain, "Deep learning for processing electromyographic signals: A taxonomy-based survey," *Journal of Electromyography and Kinesiology*, vol. 58, p. 102507, Sep. 2021, doi: 10.1016/j.jelekin.2021.102507.
- [5] M. Nikolic, "Detailed Analysis of Clinical Electromyography Signals EMG Decomposition, Findings and Firing Pattern Analysis in Controls and Patients with Myopathy and Amyotrophic Lateral Sclerosis," PhD Thesis, Faculty of Health Science, University of Copenhagen, 2001. [Online]. Available: <http://www.emglab.net>. Dataset: N2001.
- [6] S. K. Kim, J. H. Lee, J. H. Lee, J. H. Lee and K. H. Kim, "A Deep Learning Model for Automated Classification of Intraoperative Continuous EMG," *Journal of Clinical Medicine*, vol. 10, no. 10, p. 2115, May 2021, doi: 10.3390/jcm10102115.
- [7] Functions of Muscular System," Chegg Learn, accessed March 31, 2023, <https://www.chegg.com/learn/topic/functions-of-muscular-system>.
- [8] E. P. Widmaier, H. Raff, and K. T. Strang, *Vander's Human Physiology: The Mechanisms of Body Function*. McGraw-Hill Education, 2014.
- [9] R. J. Solaro and A. J. Moir, "Physiology and Pathology of Cardiac Muscle," Springer, 2018.
- [10] W. Scott, J. Stevens, and S. A. Binder–Macleod, "Human Skeletal Muscle Fiber Type Classifications," *Physical Therapy*, vol. 81, no. 11, pp. 1810–1816, Nov. 2001.
- [11] Department of Neurobiology and Anatomy, The University of Texas Medical School at Houston, "Motor Units and Muscle Receptors (Section 3, Chapter 1) Neuroscience Online: An Electronic Textbook for the Neurosciences," [Online]. Available: <https://nba.uth.tmc.edu/neuroscience/m/s3/chapter01.html>. [Accessed: Jan. 26, 2023].
- [12] University of Texas Health Science Center at Houston, "Chapter 1: Introduction to the nervous system," Neuroscience Online, [Online]. Available: <https://nba.uth.tmc.edu/neuroscience/m/s3/chapter01.html>. [Accessed: Jan. 26, 2023].
- [13] Wikipedia contributors, "Motor unit," Wikipedia, The Free Encyclopedia, May 8, 2023. [Online]. Available: https://en.wikipedia.org/wiki/Motor_unit. [Accessed: Jan. 26, 2023].
- [14] C. Waterbury, "The science of motor unit recruitment - Part 1," Chad Waterbury Fitness, Dec. 19, 2010. [Online]. Available: <https://chadwaterbury.com/the-science-of-motor-unit-recruitment-part-1/>. [Accessed: Jan. 26, 2023].

- [15] American Association of Neuromuscular & Electrodiagnostic Medicine, "What is EMG?" [Online]. Available: <https://www.aanem.org>. [Accessed: Jan. 28, 2023].
- [16] B. Jabbari and A. Machado, "Electromyography," in StatPearls [Internet], StatPearls Publishing, 2020. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK470400/>. [Accessed: Jan. 28, 2023].
- [17] R. Merletti and P. A. Parker, "Electromyography: physiology, engineering, and noninvasive applications," John Wiley & Sons, 2004.
- [18] G. Kamen and D. Gabriel, "Essentials of electromyography," Champaign, IL: Human Kinetics, 2014.
- [19] R. Merletti, A. Botter, A. Troiano, and E. Merlo, "Technology and instrumentation for detection and conditioning of the surface electromyographic signal: State of the art," *Clinical Biomechanics*, vol. 24, no. 2, pp. 122-134, 2009.
- [20] H. J. Hermens, B. Freriks, R. Merletti, D. Stegeman, J. Blok, G. Rau, and G. Hagg, "European recommendations for surface electromyography: Results of the SENIAM project," Roessingh Research and Development, 1999.
- [21] Sound Fitness and Spine Chiropractic, "Digital Range of Motion & Surface Electromyography," [Online]. Available: <https://www.soundfschiro.com/services/digital-range-of-motion-surface-electromyography/>. [Accessed: Mar. 11, 2023].
- [22] Maxi Sciences, "Electromyogramme (EMG) : définition, fonctionnement et risques de cet examen neurologique," Mar. 27, 2019. [Online]. Available: https://www.maxisciences.com/emg/electromyogramme-emg-definition-fonctionnement-et-risques-de-cet-examen-neurologique_art39336.html. [Accessed: Mar. 11, 2023].
- [23] H. J. Hermens, B. Freriks, C. Disselhorst-Klug, and G. Rau, "Development of recommendations for SEMG sensors and sensor placement procedures," *Journal of Electromyography and Kinesiology*, vol. 10, no. 5, pp. 361-374, 2000. doi: 10.1016/s1050-6411(00)00027-4.
- [24] R. Pilkar, K. Momeni, A. Ramanujam, M. Ravi, E. Garbarini, and G. F. Forrest, "Use of Surface EMG in Clinical Rehabilitation of Individuals With SCI: Barriers and Future Considerations," *Frontiers in Neurology*, vol. 11, 2020. doi: 10.3389/fneur.2020.578559.
- [25] Darras, "Definition and types of myopathy," [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK470225/>. [Accessed: Mar. 15, 2023].
- [26] B. Udd and R. C. Griggs, "Inherited myopathies," [Online]. Available: <https://pubmed.ncbi.nlm.nih.gov/20809097/>. [Accessed: Feb. 05, 2023].
- [27] National Center for Biotechnology Information, "Myasthenia Gravis," Aug. 22, 2022. [Online]. Available: <https://www.ncbi.nlm.nih.gov/books/NBK562290/#:~:text=The%20most%20common%20signs%20and,interfering%20in%20daily%20life%20activities>. [Accessed: Mar. 06, 2023].
- [28] National Institute of Neurological Disorders and Stroke (NINDS), "Peripheral Neuropathy Fact Sheet," [Online]. Available: <https://www.ninds.nih.gov/health-information/disorders/peripheral-neuropathy>. [Accessed: Apr. 01, 2023].

- [29] National Institute of Neurological Disorders and Stroke, "Peripheral neuropathy fact sheet," [Online]. Available: <https://www.ninds.nih.gov/health-information/disorders/peripheral-neuropathy>. [Accessed: Apr. 01, 2023].
- [30] I. Goodfellow, Y. Bengio, and A. Courville, "Deep Learning," MIT Press, 2016.
- [31] Quora, "What is the relationship between artificial intelligence, machine learning, deep learning, and artificial neural networks?," In Quora. Available: <https://www.quora.com/What-is-the-relationship-between-artificial-intelligence-machine-learning-deep-learning-and-artificial-neural-networks>. [Accessed: Feb. 16, 2023].
- [32] Y. Gao, B. Gao, Q. Chen, J. Liu, and Y. Zhang, "Deep Convolutional Neural Network-Based Epileptic Electroencephalogram (EEG) Signal Classification," *Front. Neurol.*, vol. 11, p. 375, 2020. doi: 10.3389/fneur.2020.00375.
- [33] H.P. Chan, R.K. Samala, L.M. Hadjiiski, and C. Zhou, "Deep Learning in Medical Image Analysis," *Adv. Exp. Med. Biol.*, vol. 1213, pp. 3-21, 2020. doi: 10.1007/978-3-030-33128-3_1.
- [34] F. Es-sabery, A. Hair, J. Qadir, B. Sainz de Abajo, B. Garcia-Zapirain, and I. De la Torre Díez, "Sentence-Level Classification Using Parallel Fuzzy Deep Learning Classifier," *IEEE Access*, vol. 9, pp. 1-1, 2021, Art no. 3053917, doi: 10.1109/ACCESS.2021.3053917.
- [35] M. Gazzaniga, R. Ivry, and G. Mangun, "Cognitive Neuroscience: The Biology of the Mind," 3rd ed. New York, NY, USA: W. W. Norton & Company, 2019.
- [36] B. Marr, "Bernard Marr - Best-Selling Author, Keynote Speaker & Consultant," Available: <https://bernardmarr.com/what-are-artificial-neural-networks-a-simple-explanation-for-absolutely-anyone/>. [Accessed: Feb. 20, 2023]
- [37] Serokell, "Introduction to Convolutional Neural Networks," Available: <https://serokell.io/blog/introduction-to-convolutional-neural-networks>. [Accessed: Feb. 26, 2023].
- [38] CNN.com - Breaking News, U.S., World, Weather, Entertainment & Video News. CNN. Available: <https://anhreynolds.com/blogs/cnn.html>. [Accessed: Feb. 20, 2023].
- [39] Aggarwal & Reddy, 2018; BP.O. Box: 3619995161ShahroodIran. "Pooling Methods in Deep Neural Networks, a Review." arXiv preprint arXiv:2009.07485, 2020.engio, 2009
- [40] Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT Press.
- [41] A. Graves, "Supervised Sequence Labelling with Recurrent Neural Networks," Springer, 2012.
- [42] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," in *Proceedings of the 32nd International Conference on Machine Learning (ICML)*, Lille, France, 2015, pp. 448-456. doi: 10.1145/3045118.3045167.
- [43] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the Inception Architecture for Computer Vision," in *Proceedings of the IEEE Conference on Computer*

Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 2016, pp. 2818-2826. doi: 10.1109/CVPR.2016.308.

- [44] G. E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, and R. R. Salakhutdinov, "Improving neural networks by preventing co-adaptation of feature detectors," arXiv preprint arXiv:1207.0580, 2012.
- [45] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.
- [46] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533-536, 1986.
- [47] V. Nair and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines," in *Proceedings of the 27th International Conference on Machine Learning (ICML-10)*, pp. 807-814, 2010.
- [48] C. M. Bishop, "Neural networks for pattern recognition," vol. 1, Oxford University Press, 1995.
- [49] J. S. Bridle, "Training stochastic model recognition algorithms as networks can lead to maximum mutual information estimation of parameters," in *Advances in neural information processing systems*, pp. 211-217, 1990.
- [50] A. L. Maas, A. Y. Hannun, and A. Y. Ng, "Rectifier nonlinearities improve neural network acoustic models," in *Proc. ICML*, vol. 30, no. 1.
- [51] I. Sutskever, J. Martens, and G. Hinton, "On the importance of initialization and momentum in deep learning," in *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*, pp. 1139-1147.
- [52] S. Shah, "What is a Loss Function?" *Towards Data Science*, June 25, 2018. [Online]. Available: <https://towardsdatascience.com/what-is-loss-function-1e2605aeb904>. [Accessed: March 10, 2023].
- [53] Brownlee, J. (2021). *How to Evaluate Machine Learning Algorithms*.
- [54] S. Singha Roy, K. Samanta, S. Modak, S. Chatterjee, and R. Bose, "Cross Spectrum Aided Deep Feature Extraction Based Neuromuscular Disease Detection Framework," in *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1-4, Jun. 2020, Art no. 6000704. doi: 10.1109/LSSENS.2020.2992452.
- [55] I. Yoo, J. Yoo, D. Kim, I. Youn, H. Kim, M. Youn, J. Won, C. Woosup, Y. Myong, S. Kim, R. Yu, S.-M. Kim, K. Kim, S.-B. Lee, and K. Kim, "Deep learning-based algorithm versus physician judgement for diagnosis of myopathy and neuropathy from needle electromyography," *bioRxiv*, p. 232845, 2023. doi: 10.1101/2023.01.13.232845.
- [56] A. Sengur, M. Gedikpinar, Y. Akbulut, E. Deniz, V. Bajaj, and Y. Guo, "DeepEMGNet: An Application for Efficient Discrimination of ALS and Normal EMG Signals," in *International Conference on Computational Science and Its Applications*, pp. 619-625, Springer, Cham, 2018. doi: 10.1007/978-3-319-65960-2_77.

- [57] I. Elamvazuthi, N. Duy, Z. Ali, S. Su, M. A. Khan, and S. Parasuraman, "Electromyography (EMG) based Classification of Neuromuscular Disorders using Multi-Layer Perceptron," *Procedia Computer Science*, vol. 76, pp. 223-228, 2015. [Online]. Available: <https://doi.org/10.1016/j.procs.2015.12.346>.
- [58] A. M. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *Biomedical Circuits and Systems*, *IEEE Transactions on*, vol. 6, pp. 156-166, 2012.
- [59] T. Ahmed and M.K. Islam, "J. Phys.: Conf. Ser. 1921 012043," 2021. [Online]. Available: DOI: 10.1088/1742-6596/1921/1/012043.
- [60] S.V. Patel and S.L. Sunitha, "Analysis and Classification of Muscular Paralysis Disease using Electromyography (EMG) Signal with Machine Learning," *International Journal of Health Technology and Innovation*, vol. 1, no. 2, pp. 35-42, 2022.
- [61] K. M. N. Hassan et al., "ALSNet: A Dilated 1-D CNN for Identifying ALS from Raw EMG Signal," pp. 1181-1185, 2022, doi: 10.1109/ICASSP43922.2022.9747366.
- [62] Python Software Foundation. (2021). Python. Available: <https://www.python.org/>. [Accessed: May. 06, 2023].
- [63] Educative, "Python Jupyter Notebook: A Guide for Beginners," August 26, 2021. [Online]. Available: <https://www.educative.io/blog/python-jupyter-notebook> . [Accessed: May. 06, 2023].
- [64] E. Zucchi, V. Bonetto, G. Soraru, I. Martinelli, P. Parchi, R. Liguori, and J. Mandrioli, "Neurofilaments in motor neuron disorders: towards promising diagnostic and prognostic biomarkers," *Molecular neurodegeneration*, vol. 15, no. 1, pp. 1-20, 2020.
- [65] Y. Lin and J. Wu, "A Novel Multichannel Dilated Convolution Neural Network for Human Activity Recognition," *Mathematical Problems in Engineering*, vol. 2020, Article ID 5426532, 10 pages, 2020. [Online]. Available: <https://doi.org/10.1155/2020/5426532>

Appendix A

This appendix presents the second model we developed using Long Short-Term Memory (LSTM). The model was not included in the previous chapter due to resource constraints. Specifically, we ran out of time and computing resources to train the model on our dataset. To train the model, we needed a graphics processing unit (GPU) accelerator, since it would take hours to run one epoch using a central processing unit (CPU). We used Kaggle's GPU accelerator, but the limit for the session was 12 hours, which was not enough time for the model to complete training. In this appendix, we present the model that we developed, and the results that we could obtain using the available resources.

A.1 LSTM Model

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is well-suited for processing sequential data such as, in our case, Electromyography (EMG) signals. LSTMs are particularly useful for analyzing EMG data because they can capture long-term dependencies in the signal, such as changes in muscle activation over time, and use this information to make accurate predictions or classifications.

The proposed model consists of three consecutive Long Short-Term Memory (LSTM) layers. The first and second LSTM layers contain 128 memory units and is set to return sequences, meaning it outputs a sequence for each input timestep. The third LSTM layer, which also contains 128 memory units, differs by outputting only the final hidden state instead of sequences. The model concludes with a dense layer consisting of three units, which employs the softmax activation function to compute probabilities for each of the three classes. **Figure A.1** summarizes the model used.

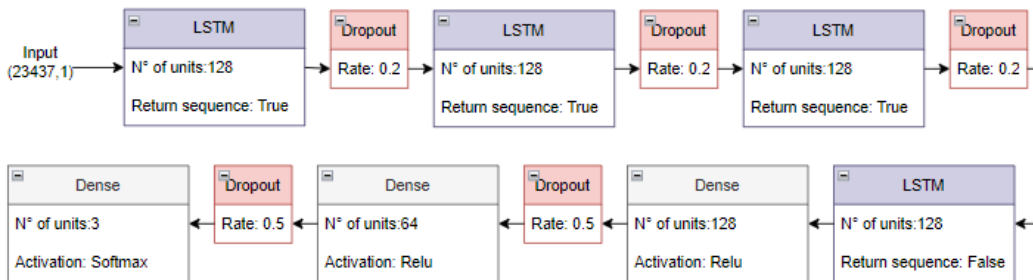


Figure A.1: LSTM model for multiclass classification.

A.2 Results and Discussion

The model was trained using the Adam optimizer, with an initial learning rate of 0.001, which is a small value that helps to prevent the model from overfitting the training data. The model was trained for 60 epochs, with a batch size of 32. The training and validation accuracies were 67,22% and 63,87% respectively. The test accuracy was 64,86%. The loss and accuracy plots illustrated in **Figure A.2** show fluctuation in the values, which implies that more training time might give better results.

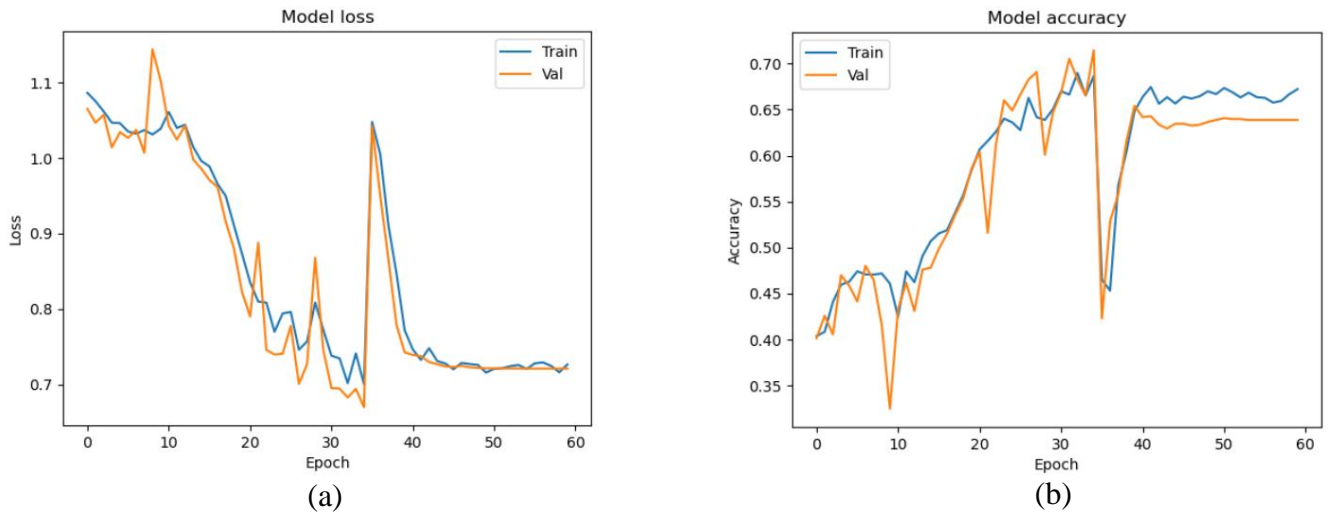


Figure A.2: Train and Validation Plots for Multiclass Classification Train-Test Split Method

(a) Loss Plots. (b)Accuracy Plots.

Figure A.3 below represents the confusion matrix for this experiment. Overall, the recall values for ALS and myopathy are relatively high, which suggests that the model is able to identify these conditions well. However, the recall value for normal is relatively low, which suggests that the model may have difficulty identifying healthy patients.

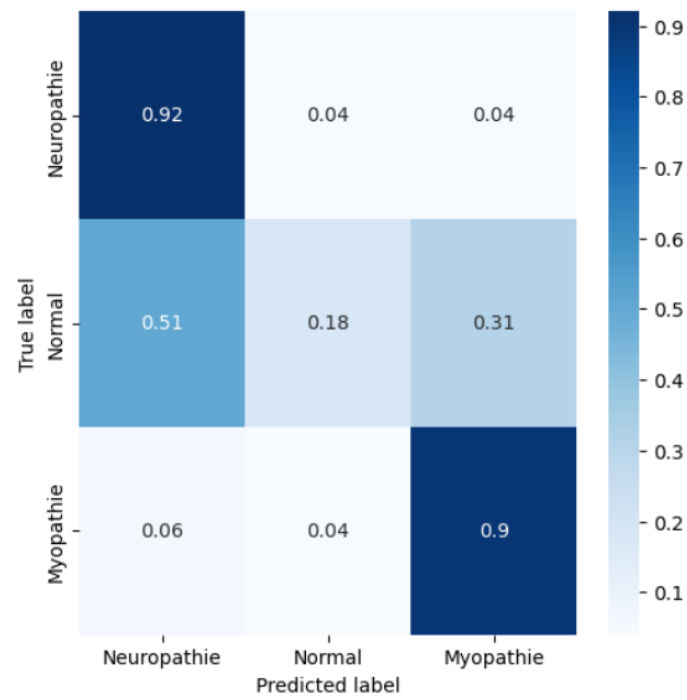


Figure A.3: Confusion Matrix for Multiclass Classification Train-Test Split Method.