Faculdade de Design, Tecnologia e Comunicação
Universidade Europeia

2024

MIGUEL MARQUES ALVES MAYA PINHEIRO

EMG-BASED GESTURE RECOGNITION: CNN FEATURE CONTRIBUTION AND ABLATION STUDY

Faculdade de Design, Tecnologia e Comunicação
Universidade Europeia

2024

MIGUEL MARQUES ALVES MAYA PINHEIRO

EMG-BASED GESTURE RECOGNITION: CNN FEATURE CONTRIBUTION AND ABLATION STUDY

Dissertação apresentada ao IADE - Faculdade de Design, Tecnologia e Comunicação da Universidade Europeia, para cumprimento dos requisitos necessários à obtenção do grau de Mestre em Creative Computing and Artificial Intelligence realizada sob a orientação científica da Doutora Cláudia Sofia Sevivas Ribeiro, *Professora Auxiliar* do *IADE – Faculdade de Design, Tecnologia e Comunicação* e do Doutor Hugo Humberto Plácido da Silva, *Professor Auxiliar Convidado* do *Instituto Superior Técnico, Universidade de Lisboa, Portugal*.

To my parents, whose love and guidance are with me in whatever I pursue.

> "A King may move a man, a father may claim a son, but remember that even when those who move you be Kings, or men of power, your soul is in your keeping alone. When you stand before God, you cannot say, 'But I was told by others to do thus.' Or that, 'Virtue was not convenient at the time.' This will not suffice. Remember that."
>
> — King Baldwin IV in *Kingdom of Heaven*

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Cláudia Sofia Sevivas Ribeiro, and my co-advisor, Prof. Hugo Humberto Plácido da Silva, whose expertise, understanding, and patience considerably added to my graduate experience. I appreciate their vast knowledge and skill in guiding me through the research and writing of this thesis.

I am thankful for the well-equipped facilities provided by IADE - Faculdade de Design, Tecnologia e Comunicação, Universidade Europeia, and a word of appreciation to IT - Instituto de Telecomunicações for all the support provided during the course of this work.

I am sincerely grateful to all my professors in the course who helped not only with the thesis but also in creating the skills necessary to reach this point.

Heartfelt thanks to my colleagues, whose friendship and insights helped me stay sane and motivated through the challenges of this thesis.

Abstract

Electromyography (EMG) has emerged as a pivotal technology in biomedical engineering, offering valuable insights into muscle activity with significant applications in adaptive prosthetics. This thesis delves into the application of EMG signal classification to enhance gesture recognition for prosthetic control. Utilizing the GRABMyo dataset, which includes comprehensive EMG data from 43 participants performing various hand and wrist gestures, this study provides a robust platform for testing and validating classification algorithms. The dataset was meticulously organized, ensuring clear separation between training and testing sets to facilitate rigorous evaluation. The research focuses on the implementation of Convolutional Neural Networks (CNNs) with an AlexNet architecture for their superior feature extraction capabilities. An ablation study was conducted to systematically assess the influence of different neural network components on the model's performance. By selectively disabling specific filters within the convolutional layers, the study aimed to identify the most critical features and components that significantly contribute to accurate gesture recognition. The objectives of this ablation study include optimizing the network by focusing on impactful elements and improving gesture classification accuracy. Following the ablation study, a recovery phase was implemented to restore and potentially enhance the model's performance. This phase tested the network's resilience and adaptability, providing insights into its built-in redundancy and ability to compensate for lost functionalities through targeted training strategies. The recovery effort not only aimed to regain lost performance but also to deepen the understanding of which features and layers are most critical to the network's operation. The final phase involved validating the optimized model on a test set comprising participants not included in the training phase. To visually demonstrate the model's effectiveness, the Unity engine was used to simulate the gestures based on the model's classifications, offering an intuitive and interactive representation of its capabilities. This visualization underscores the practical applications of the technology in adaptive prosthetics, bridging the gap between theoretical research and real-world implementation.

Keywords: Electromyography (EMG), Surface EMG (sEMG), EMG Signal Processing, Feature Extraction, Gesture Recognition, Prosthetic Control, Machine Learning, Deep Learning, Convolutional Neural Networks, Ablation Study, Model Recovery.

Resumo

A eletromiografia (EMG) emergiu como uma tecnologia fundamental na engenharia biomédica, oferecendo insights valiosos sobre a atividade muscular com aplicações significativas em próteses adaptativas. Esta tese explora a aplicação da classificação de sinais EMG para melhorar o reconhecimento de gestos no controle de próteses. Utilizando o conjunto de dados GRABMyo, que inclui dados EMG abrangentes de 43 participantes realizando vários gestos com as mãos e os pulsos, este estudo fornece uma plataforma robusta para testar e validar algoritmos de classificação. O conjunto de dados foi meticulosamente organizado, garantindo uma separação clara entre os conjuntos de treino e teste para facilitar uma avaliação rigorosa. A pesquisa concentra-se na implementação de Convolutional Neural Networks (CNNs) com arquitetura AlexNet devido às suas capacidades superiores de extração de características. Um estudo de ablação foi conduzido para avaliar sistematicamente a influência de diferentes componentes da rede neural no desempenho do modelo. Ao desativar seletivamente filtros específicos dentro das camadas convolucionais, o estudo visou identificar as características e componentes mais críticos que contribuem significativamente para o reconhecimento preciso de gestos. Os objetivos deste estudo de ablação incluem a otimização da rede, focando em elementos impactantes e melhorando a precisão da classificação de gestos. Após o estudo de ablação, uma fase de recuperação foi implementada para restaurar e potencialmente melhorar o desempenho do modelo. Esta fase testou a resiliência e adaptabilidade da rede, obtendo maior clareza sobre sua redundância inerente e a capacidade de compensar funcionalidades perdidas através de estratégias de treino direcionadas. O esforço de recuperação não visou apenas recuperar o desempenho perdido, mas também aprofundar o entendimento de quais características e camadas são mais críticas para a operação da rede. A fase final envolveu a validação do modelo otimizado em um conjunto de teste composto por participantes não incluídos na fase de treinamento. Para demonstrar visualmente a eficácia do modelo, o motor Unity foi utilizado para simular os gestos com base nas classificações do modelo, oferecendo uma representação intuitiva e interativa de suas capacidades. Esta visualização sublinha as aplicações práticas da tecnologia em próteses adaptativas, fechando a lacuna entre a pesquisa teórica e a implementação no mundo real.

Contents

List of Figures

List of Tables

To my parents, whose love and guidance are with me in whatever I pursue.

> "A King may move a man, a father may claim a son, but remember that even when those who move you be Kings, or men of power, your soul is in your keeping alone. When you stand before God, you cannot say, 'But I was told by others to do thus.' Or that, 'Virtue was not convenient at the time.' This will not suffice. Remember that."
>
> — King Baldwin IV in *Kingdom of Heaven*

Acknowledgements

I would like to express my deepest gratitude to my advisor, Prof. Cláudia Sofia Sevivas Ribeiro, and my co-advisor, Prof. Hugo Humberto Plácido da Silva, whose expertise, understanding, and patience considerably added to my graduate experience. I appreciate their vast knowledge and skill in guiding me through the research and writing of this thesis.

I am thankful for the well-equipped facilities provided by IADE - Faculdade de Design, Tecnologia e Comunicação, Universidade Europeia, and a word of appreciation to IT - Instituto de Telecomunicações for all the support provided during the course of this work.

I am sincerely grateful to all my professors in the course who helped not only with the thesis but also in creating the skills necessary to reach this point.

Heartfelt thanks to my colleagues, whose friendship and insights helped me stay sane and motivated through the challenges of this thesis.

Abbreviations

EMG	**Electromyography**
sEMG	**Surface Electromyography**
KNN	**K-Nearest Neighbors**
SVM	**Support Vector Machines**
CNN	**Convolutional Neural Networks**
RNN	**Recurrent Neural Networks**
AP	**Action Potential**
RN	**Radial Nerve**
UN	**Ulnar Nerve**
MN	**Median Nerve**
DIP	**Distal Interphalangeal**
PIP	**Proximal Interphalangeal**
AgCl	**Silver Chloride**
TMR	**Targeted Muscle Reinnervation**

PLI	**Power-Line Interference**
MCA	**Minimal Crosstalk Area**
DC	**Direct Current**
AC	**Alternative Current**
WT	**Wavelet Transform**
DWT	**Discrete Wavelet Transform**
CWT	**Continuous Wavelet Transform**
SURE	**Steins Unbiased Risk Estimate**
EEG	**Electroencephalography**
ACC	**Accelerometry**
EDA	**Electrodermal Activity**
mV	**Millivoltage**
In-Amp	**Instrumentation Amplifier**
Op-Amp	**Operational Amplifiers**
USART	**Universal Synchronous/Asynchronous Receiver/Transmitter**
CRC	**Cyclic Redundancy Check**
LSFR	**Linear Feedback Shift Register**
SNR	**Signal-to-noise Ratio**

1D-CNN	**One-dimensional Convolutional Neural Networks**
2D-CNN	**Two-dimensional Convolutional Neural Networks**
ReLu	**Rectified Linear Unit**
LSTM	**Long Short-Term Memory**
TANH	**Hyperbolic Tangent Function**
PCA	**Component Analysis**
LP	**Lateral prehension**
TA	**Thumb adduction**
TLFO	**Thumb and little finger opposition**
TIFO	**Thumb and index finger opposition**
TLFE	**Thumb and little finger extension**
TIFE	**Thumb and index finger extension**
IMFE	**Index and middle finger extension**
LFE	**Little finger extension**
IFE	**Index finger extension**
TE	**Thumb extension**
WF	**Wrist flexion**
WE	**Wrist extension**

FP	**Forearm pronation**
FS	**Forearm suponation**
HO	**Hand open**
HC	**Hand close**

Chapter 1

Introduction

1.1 Background

The human hand, a product of millions of years of evolution, is essential for interaction with our environment. It allows for a diverse array of complex movements, enabling users to perform numerous tasks. Losing a hand disrupts normal life profoundly, impacting both physical function and psychological well-being. Prosthetic hands have become crucial in assisting amputees who have lost their arms due to disease or injury, helping to restore some functionality.

However, despite the advancements in prosthetic technologies, approximately 50–60% of amputees use a prosthetic hand, and rejection rates are around 44% according to Salminger et al., 2022. Some reasons for this might be the discomfort that some user express and the challenge of replicating the bidirectional communication between the nervous system and the hand, a task that has proven difficult over the past few decades. Bionic hands, which closely resemble natural hands in form and function, have been a focus of scientific and engineering research which aims at improving their compatibility with natural hand movements.

The neuromuscular system, including nerves like the median, radial, and ulnar nerves, and their connection to skeletal muscles and bones, facilitates hand movement, acting as a communication network with the neurons in order to produce the hand movement (Bu et al., 2022; T. Kuiken, 2003; T. A. Kuiken et al., 2009; Ngeo et al., 2014; Röhrle et al., 2019). Recent advancements in Electromyography (EMG) signal acquisition and processing have brought the world closer to solving control issues for hand prosthetics (Chowdhury et al., 2013; T. A. Kuiken et al., 2009). By acquiring the electrical potentials associated with muscle activation, EMG technology offers insights into muscle activity, which is crucial for developing prosthetics that can emulate natural limb movements. This technology's ability to interpret these signals can be the key to designing responsive and intuitive prosthetic hands. One of the biggest challenges in prosthetic design is accurately predicting and replicating finger movements, given the complexity of hand actions (Ngeo et al., 2014).

1.2 Motivation and Aim

The main motivation for this work is to bring back mobility to those who for accident reasons or due to diseases have lost a member of the upper limb body, more specifically the hand. Over the past few decades, numerous studies have aimed to replicate hand movements. However, with recent advancements in artificial intelligence, the development of more responsive and intuitive prosthetic hands is increasingly within reach. (Bu et al., 2022). Although a lot of breakthroughs investigations have been realized in the more recent years there is still a lack of investigation on using AI algorithms to predict hand and finger movements through other muscles, such as the ones located in the forearm which are also innervated by the same nerves that innervate the hand, to train a model that is able to predict movement throw the contractions of the skeletal muscles (Ngeo et al., 2014).

Another reason for carrying out this study is to reduce the prosthetic rejection rates. To achieve this objective this study will focus on using less invasive method of acquiring EMG data using surface electrodes on the forearm, with the objective of a preliminary analysis of how these prosthetics can be more comfortable for patients to use.

Beyond the technical achievement, this work aims to contribute to a deeper understanding of muscle signal interpretation and its practical application in prosthetic technology (T. A. Kuiken et al., 2009). The ultimate goal is to pave the way for more adaptive, responsive, and user-friendly prosthetics, thereby enriching the lives of those they serve.

1.3 Thesis Structure

This thesis is organized into eight main parts, each addressing a crucial aspect of employing Electromyography (EMG) for the development of adaptive prosthetics. The structure guides the reader through a logical progression from foundational concepts to detailed applications.

The Introduction sets the stage by highlighting the importance of EMG in biomedical engineering, specifically for enhancing prosthetic control. It defines the research objectives and outlines the scope of the thesis. Following the introduction, the State of the Art section reviews existing literature on EMG technologies, focusing on signal processing techniques and classification methods. It discusses the advancements in neural networks applicable to EMG analysis and addresses the challenges related to signal interference.

In the Methodology chapter, the experimental design is detailed, including the selection and preparation of the GRABMyo dataset and the rationale behind choosing the AlexNet architecture for the convolutional neural network (CNN). This preparation leads into the Experiments section, which outlines the data processing and training of the CNN, setting up for the in-depth ablation study that follows.

The Experimental Evaluation section delves into the ablation study, aiming to pinpoint critical network components that significantly impact performance. It also covers the recovery

phase of the model, assessing the network's adaptability and resilience, and incorporates the use of the Unity engine to visually evaluate the model's gesture recognition capabilities.

The Discussion chapter compares the outcomes of this study to similar research, highlighting the unique contributions and limitations of the methods used. This analysis helps to contextualize the findings within the broader field of EMG research and prosthetic development.

The thesis concludes with the Conclusion section, which synthesizes the findings and reflects on their implications for improving prosthetic control technologies. It underscores the study's contributions to the field and potential impacts on future technologies. Finally, the Future Work section proposes directions for subsequent research, suggesting how further studies could refine gesture classification techniques or explore new applications, building on the foundational work of this research.

Chapter 2

State-of-the-Art

2.1 Neuromuscular System

The neuromuscular system represents a sophisticated network that is fundamental to the coordination and execution of movement in the human body. Central to this system are the neurons, muscle fibers, and neuromuscular junctions, which, according to Röhrle et al., 2019, collaborate to transform neural signals into muscle actions. Motor neurons, specialized neural cells, serve as critical transducers of these signals, acting like biological wires that deliver electrical impulses to the muscles, which essencially convert them into mechanical force and motion. The system does not function in isolation; it integrates commands from the central nervous system, which includes both conscious and automatic control, with sensory feedback that monitors the status of muscle length and contraction velocity. This feedback works like a constant change of information between the muscles and the brain, which ensures movements to be smooth and coordinated. This integration facilitates the nervous system's precise control over bodily movements and interactions with the environment.

The skeletal muscles, containing muscle fibers attached together by a connective tissue framework, are integral to this system. Röhrle et al., 2019 describe how this connective tissue, primarily made of collagen and elastin, is hierarchically organized into the endomysium, perimysium, and epimysium. Each layer plays a distinct role in muscle structure and function, which can be seen in Figure 1. These layers work together, each with a unique position that, when combined, create a strong and flexible muscle structure. The endomysium wraps each muscle fiber, linking to the basement membrane, while the perimysium segments the muscle into fiber clusters, providing pathways for blood vessels and nerves. These pathways guarantee that nutrients and signals get into the muscle, which is essential for the muscle to maintain its integrity. This arrangement not only maintains muscle integrity and shape, but also protects it against excessive muscle stretching and disperses forces to attenuate damage to the muscle fibers.

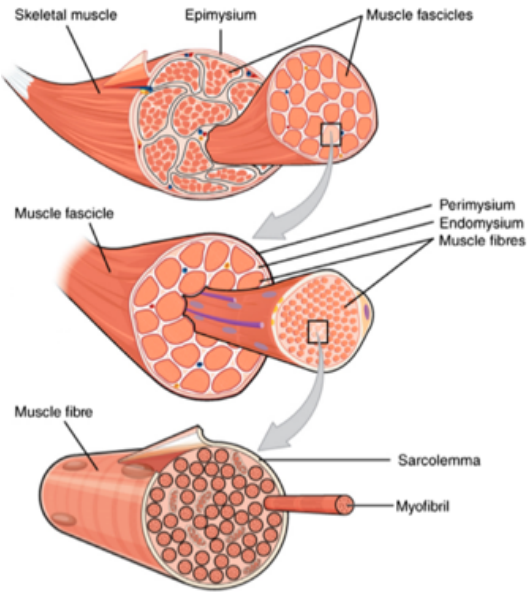


Figure 1: Skeletal muscle structure (image by OpenStax Anatomy and Physiology, as cited in Röhrle et al., 2019, page 8).

Furthermore, the Action Potential (AP) is a critical communicative signal within this system. As detailed by Röhrle et al., 2019, the neuron's AP, upon reaching the neuromuscular junction, initiates neurotransmitter release, triggering a cascade of events within the muscle fiber that results in the propagation of the AP. This leads to the release of calcium from the sarcoplasmic reticulum and once it binds with the troponin-tropomyosin complex, culminates in muscle contraction. The excitation-contraction coupling process, crucial for translating neural signals into muscle movement, is schematically detailed in Figure 2. This coupling is the essential link that converts an electrical signal into a physical movement. This mechanism underpins the essential communication between the nervous and muscular system, producing muscle movements.

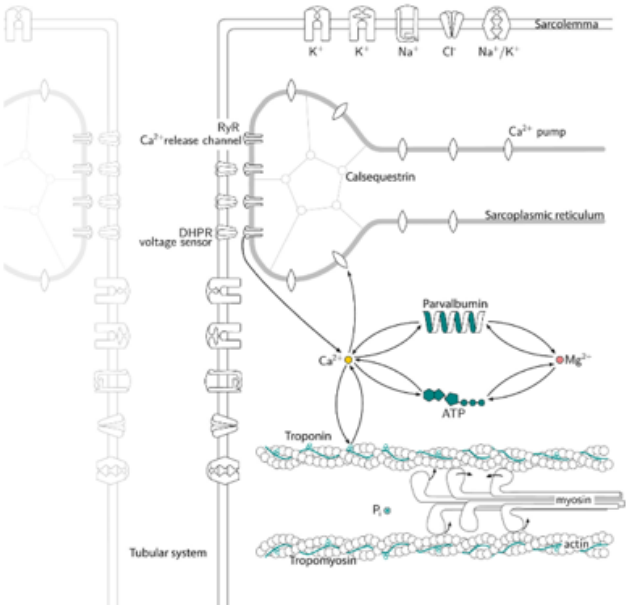


Figure 2: Schematic representation of the excitation-contraction coupling in a muscle fiber (image by Heidlauf et al., (2016) cited by Röhrle et al., 2019, page 11).

The upper limb muscular architecture, which works on the same principles of how action potentials drive muscles to contract, is an array of muscles that allow for an extensive range of movements and functions, from broad gestures to the finest movements. The AP triggers the contraction of major muscles in the shoulder, arm, forearm, and hand, that produce contractions and relaxations, enabling a synchrony in the movements for the spectrum of activities essential for daily life. According to Bu et al., 2022, it is important to note two different upper limb motion muscle groups, which are the elbow motion groups (composed by the muscles in the branchium), and the wrist motion groups (composed by the muscles located in the forearm). The focus of the thesis will be in the wrist motion muscle group, building upon studies by Ngeo et al., 2014 , which show that the muscles responsible for the more precise movement of the hand and fingers are located in this muscle group (Table 2). Nevertheless, it is important to note that the study done by Bu et al., 2022 shows that the shoulder articulation, the biceps brachii, and triceps brachii contributions are essential to understand more complex movements of the wrist and hand gestures.

Table 2: Target muscles and their corresponding actions on the hand and fingers (original table from Ngeo et al., 2014, page 3)

Target Muscle	Hand/Finger Affected
Abductor Polliscis Longus	Abductor Polliscis Longus
Flexor Carpi Radialis	Wrist, hand flexor and abduction
Flexor Digitorum Superficialis	2-5th finger PIP flexion
Flexor Digitorum Profundos	2-5th finger DIP flexion
Extensor Digitorium	2-5th finger extension
Extensor Indices	Index finger
Extensor Carpi Ulnaris	Wrist extension and abduction
Extensor Carpi Radialis	Wrist and thumb

To effectively position surface electrodes for EMG readings, it is crucial to analyze the key nerves and their associated muscles involved in finger movement. This understanding will enable the creation of a valuable dataset, which is essential or adequately training an algorithm designed to predict finger motions, facilitating the development of adaptive prosthetic hands. In Figure 3 we can see the main nerves that affect the movement of the hand. These nerves are composed by the Radial Nerve (RN) the Ulnar Nerve (UN) and the Median Nerve (MN). According to T. Kuiken, 2003 and Ngeo et al., 2014 the RN is connected to the Adductor Pollicis Longus (represented in Figure 4), the Extensor Digitorum (represented in Figure 4), the Extensor indicis, and Extensor Carpi Radialis (represented in Figure 4), which are not only responsible for extending the wrist and finger movement (form all the 5 fingers extensions) but also for the abduction of the thumb and even the extension of the elbow. The EMG readings of the muscles innervated by the RN provide crucial information to enable a prosthetic hand to execute these precise extensions and movements.

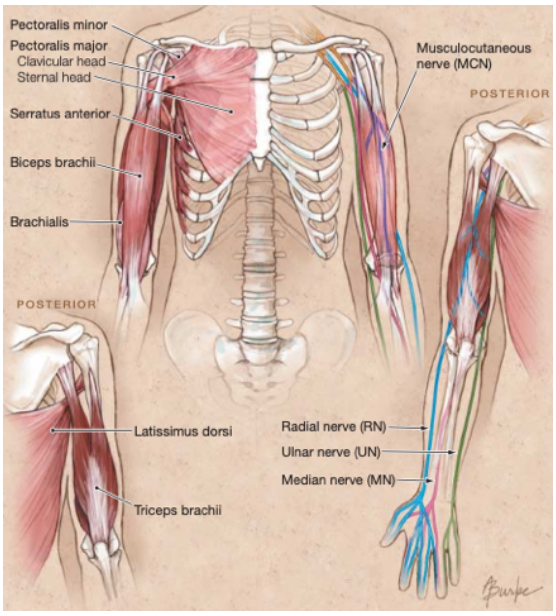


Figure 3: Representation anatomy of the main muscles and nerves in the upper body limb (image from T. A. Kuiken et al., 2009, page 2)

The UN innervates muscles such as the Extensor Digitorium (represented in Figure 4) and the Extensor Carpi radialis (represented in Figure 4) that control the intricate movements of the hand, affecting grip strength and the movement of the smaller fingers such as the pinky and ring fingers; the EMG data from ulnar-innervated muscles are utilized for the recognition and control of fine movements patterns.

The MN controls several muscles, including the Flexor Digitorum Profundus. This muscle is located deep within the forearm and connects to the distal phalanges bases of the fingers. It enables the bending of the fingertips by flexing the Distal Interphalangeal (DIP) joints. Above it lies the Flexor Digitorum Superficialis, depicted in Figure 5(b). This muscle is more superficial, situated just over the Flexor Digitorum Profundus. It attaches to the fingers' middle phalanges and flexes the Proximal InterPhalangeal (PIP) joints. Together, these muscles coordinate to facilitate finger movements. It also innervates the Extensor Carpi Radialis (represented in Figure 4), which is responsible for wrist and thumb movements that are fundamental to perform most of the hand grips. Acquiring EMG signals from median-innervated muscles are vital for the dexterity and broad range of functional grasps.

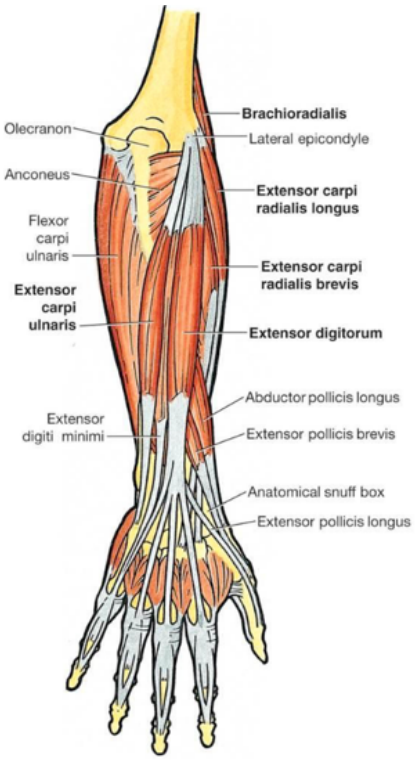


Figure 4: Representation of certain muscles in the forearm

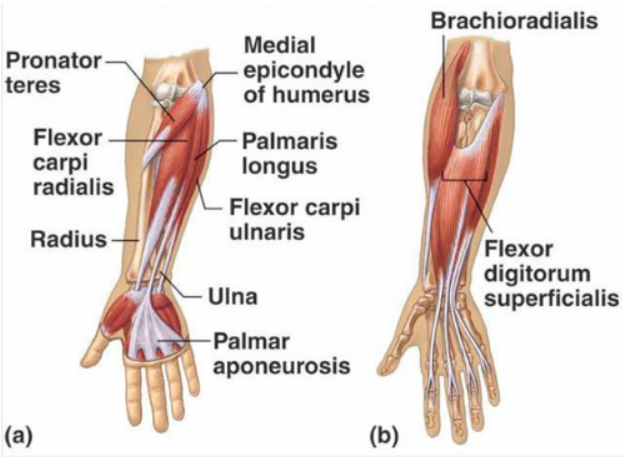


Figure 5: Anatomy of the arm with: (a) Representation of certain muscles in the inside face of the forearm; (b) Representation of certain muscles in the outside face of the forearm

In the study by Ngeo et al., 2014, it explores the EMG as a means to control prosthetic devices; researchers recorded muscle activity, from the muscles listed in Table 2, using eight silver-silver chloride (Ag-AgCl) electrodes with a 20 mm inter-electrode spacing. These electrodes were strategically placed on the subjects, as shown in Figure 6, ensuring optimal signal acquisition. Grounding was established at the elbow's pointy part, known as the olecranon with a single electrode. Signals were amplified in the analog domain and sampled through a 12-bit Analog-to-Digital converter, with real-time monitoring to validate signal integrity. Participants, positioned to ensure stability, executed three distinct tasks to articulate their fingers and hand. The initial

tasks required the isolated movement of each finger, while the second task was to move the fingers all at the same time; the final task was more focused on a free choice movement defined by the participant, with subjects moving their fingers freely. The outcome Ngeo et al., 2014 research highlight the potential of EMG-based methods in manipulating a multi degree-of-freedom device such as hand prosthetics.

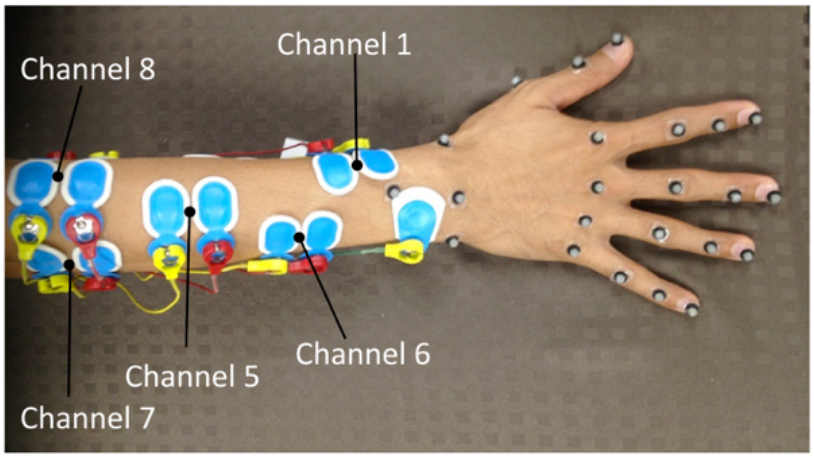


Figure 6: EMG placement on the forearm muscles from Table 2 (image from Ngeo et al., 2014, page 4).

2.2 Electromyography (EMG)

Electromyography (EMG) is a procedure used to measure and record the electrical signals generated by skeletal muscles, in other words, muscle tissue that is attached to bones by tendons, these muscles are essential to generate movement throughout the body. The EMG records the electric signals from these muscles. The measured activity is generated when muscle fibers are activated, either due to voluntary contraction or in response to external stimulation. EMG provides a window into the range of signals that make muscle movements possible, from the subtle twitch of a finger to the full grip of a hand. The characteristics of the EMG signals can also be influenced by the internal structure of the subject. This includes factors like skin formation, blood flow velocity, skin temperature and tissue structure, such as muscle and fat percentages and the anatomical location in which these signals recorded from. EMG signals are becoming increasingly important in many applications, including prosthesis and rehabilitation devices, human machine interactions, and more.

In the context of prosthetics for hands, gesture classification based on EMG is remarkable for its groundbreaking capabilities and interpreting the unique patterns of muscle signals; an EMG classifier can associate a signal to the intended movement, creating the possibility of using machine learning algorithms to predict what EMG signals are related to the intended movement of an individual. According to T. A. Kuiken et al., 2009, an algorithm can be designed in order to recognize patterns that can be interpreted as surface EMG signals from regenerated muscles, enabling natural control of motorized prothesis of the elbow, wrist and hands. This is especially

crucial for those who have lost a limb. Understanding and classifying EMG signals can create a path for the development of prothesis that have the capacity to move and function in a similar way to natural limbs. As a result, in addition to the use of EMG for muscle activity assessment, it can also be used for prosthetic limb control. The ultimate goal is to develop a prosthesis that mimics natural movements. This would mark a significant step in giving hand amputees a renewed sense of normalcy and functionality of a limb that was once deemed irrecoverable.

However, using EMG signals for these multiple applications still have a major problem to overcome, EMG signal noises, in order to achieve a improved performance in the usage of EMG signals. Studies by Chowdhury et al., 2013 demonstrate that in EMG, the detection, processing and classification of the data are essential to enable a uniform and detailed understanding of results.

EMG signals require precise recording techniques to ensure accuracy and clarity. These recording methods can be broadly classified into three primary categories: invasive, partially invasive, and non-invasive (surface) techniques. Each category offers different sets of advantages and challenges, and need to be tailored to the specific applications. In order to understand which one is better suited for the study that's being conducted we first need to understand them.

2.2.1 Intramuscular Electrodes

Intramuscular electrodes, an invasive method, are composed of thin needle-like electrodes, such as needle electrodes or fine wire electrodes, that are placed directly into the muscle tissue. This method is used to record signals from deep muscles or to capture activity from individual motor units. This method has the advantage of being highly specific, but as a disadvantage they are very invasive can cause discomfort to the user and the technology developed is not suited yet for long-term monitoring. In addition, the user of this type of method needs to undergo surgery in order to place the electrodes.

Among the invasive electrodes methods, Targeted Muscle Reinnervation (TMR) has been more successful in this category; this method involves redirecting remaining arm nerves to different muscle locations. According to T. A. Kuiken et al., 2009, once these nerves have been reinnervated, the target muscles can generate eletromyography (EMG) signals that create patterns that can be detected on the skin's surface and later analyzed.

In a study conducted by T. A. Kuiken et al., 2009, the TMR method was targeted to analyze patterns produced by the combination of EMG signals.

2.2.2 Subdermal Electrodes

Subdermal electrodes are a partially invasive electrodes method, in which the electrodes are inserted just beneath the skin but do not penetrate deep into the muscular tissue; this method still requires some medical procedure but not surgery. The most common approach is subcutaneous needle electrodes, which provides a compromise between signal quality and invasiveness. One

advantage of this method is that it involves an easier medical procedure that gives a better signal quality than surface electrodes. It also reduces the interference from adjacent muscles but, on the other hand, still causes some discomfort and pain to the patient. Due to insertion of the needles through the skin, potential skin irritation can occur and besides that, any subsequent pain can typically arise if the needle is close to a nerve. According to Daube and Rubin, 2009, this pain can be reduced by making slight adjustments to the needle position. Nevertheless, this method is not advised for long term usage.

One of the notable methods of subdermal electrodes is the Subcutaneous Needle Electrode Technique (SNET). This technique emphasizes the strategic placement of electrodes beneath the skin to harness refined neuromuscular signals. As highlighted in the article written by Daube and Rubin, 2009, the subcutaneous needle electrodes has the advantage of recording EMG activity with minimal interference from surrounding muscles, due to their fixed-size recording surface.

2.2.3 Surface Electrodes

Surface electrodes, a non-invasive method, are placed on the skin over the muscle or muscle group of interest. The skin needs to be cleaned and sanitized to reduce electrode impedances. While the actual impendance is not critical to performance it can improve signal quality. The underlying principle to surface electrodes is based on the electrical currents produced by muscle fibers, which can be detected by the surface electrodes. Due to the fact of only receiving these signals at the skin, usually very small voltages are involved, in the level of millivolts, having the need of amplification in order to make a suitable detection. This method of recording EMG signals has a significant challenge; since the electrodes are at the surface level, these can capture interferences, which consist in unwanted frequencies from other sources besides the muscles, for example, electrical noises from nearby equipments or even adjacent muscles. In order to improve the performance of the acquired signals, these need to go through pre-processing methods to eliminate these possible noises. This method is the most commonly used for EMG recordings and has the advantage of being non-invasive; electrodes are easy to apply and suitable for long-term monitoring , causing minimal to no-discomfort to the user. On the other hand, this method has some restrictions, such as only recording superficial muscles, leaving behind some of the deep muscle activity, and can potentially suffer from interference from adjacent muscles.

2.3 Electromyography (EMG) Interference

After exploring the nuances of EMG signal acquisition, it is pertinent to highlight a significant challenge in this field which are the interferences in EMG reading; these are named EMG noises. According to Chowdhury et al., 2013, the non-invasive nature of surface electrodes, while advantageous, also makes them prone to external disturbances. Such noises can considerably alter the genuine muscle activity signals. In the forthcoming sections, the various sources of

these interferences will be reviewed, together with their effects on EMG data, and the methods adopted to counteract these disruptions, ensuring the integrity and accuracy of EMG analyses.

As stated by Bau et al., 2001, the most common recording electrodes used are silver chloride (Ag/AgCl) electrodes, which can vary in sizes; these have been found to be the most adequate when talking about signal-to-noise ratio for surface electrodes, due to their good conductive properties and biocompability. There were also studies such as the one made by Chowdhury et al., 2013, that shows that when increasing the electrode recording surface the impendance will decrease. However, this same study also show that the electrode shouldn't be very large (with a maximum size of 10×1 mm for optimal signal-to-noise ratio in order to have a precise recording of the intended muscle; these will also vary depending on the circuit design and the quality of the instruments used.

2.3.1 Motion Interference

When a muscle is mobilized, not only does the muscle itself move, but the skin and the electrode also shift, generating artifacts known as motion noise. Chowdhury et al., 2013 highlight that this type of interference typically occurs within a frequency range of 1-10 Hz and has a voltage amplitude similar to that of an EMG signal. To mitigate this motion noise, several steps can be taken such as cleaning the skin where the electrode will be placed and applying a conductive gel between the skin and the electrode interface. While these measures help minimize the motion of the skin and electrode, they are less effective against differential skin layer impedance.

Bau et al., 2001 suggest that soaking the targeted skin areas in warm water for five minutes can reduce the original skin layer impedance by up to 50% compared to non-soaked readings. However, this method may not significantly impact the overall readings to justify the inconvenience of soaking multiple parts of the upper body limbs of participants.

2.3.2 Ambient Interference

The human body is constantly exposed to electrical and magnetic radiation. This exposure often results in unwanted signals overlaying the EMG readings, which can sometimes mask the signals captured during the EMG reading. This type of interference is called electromagnetic/ambient noise, and the amplitude of this type of interference can be one to three times bigger than the EMG signals of interest and it's impossible to completely negate. According to Chowdhury et al., 2013, this type of noise is due to the omnipresent electronic sources created by the nature and man-made. One of the primary sources of this noises is the radiation from power sources with a typical frequency between 50-60 Hz. This type of interferences arises due to the different resistances to alternating current which can come not only from the outside but also from the cables that connect the electrodes to the patient. It is particularly problematic giving the fact that these frequencies are close to the frequency range of the EMG signals; this type of interference its termed Power-Line Interference (PLI). To mitigate PLI in EMG recordings, one effective

method involves post-recording processing. A common technique is to use a high-pass filter to reduce or eliminate the effects of PLI. This process typically requires a reference PLI, which is subtracted from the EMG readings contaminated with PLI to produce a denoised signal.

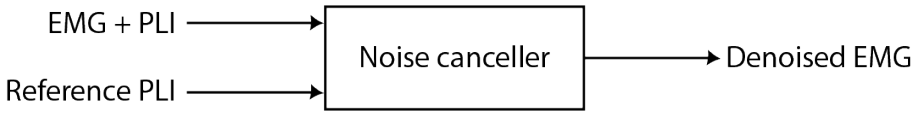


Figure 7: General block diagram of PLI cancelling system (image author (Chowdhury et al., 2013 page 4).

To better understand the interference caused by PLI, Chowdhury et al., 2013, provided a mathematical representation. The equation (1) constructs the 50 HZ PLI along with its four harmonics, namely 100, 200, 300, 400 Hz and is represented by:

$$\mathrm{PLI}_{\mathrm{ref}} = \cos(2\pi 50t) + \cos(2\pi 100t) + \cos(2\pi 200t) + \cos(2\pi 300t) + \cos(2\pi 400t) \qquad (1)$$

In the mathematical equation (1), the variable **t** stands for time. The equation employs the cosine function (represented as cos) and multiplies it with 2 times the mathematical constant π The frequencies used in the equation, specifically 50, 100, 200, 300, and 400 Hz, corresponds to the fundamental frequency of the PLI, which is 50 Hz, and its subsequent four harmonics.

2.3.3 Crosstalk Interference

Besides the "outside" noises the EMG can also capture "inside" noises, this can occur when the EMG captures EMG signals that come from other muscles besides the primary focus of the monitoring ones. This interference is called crosstalk noises and this type of noise can lead to misinterpretation of the EMG signals, making it seem like there is muscle activity when it might not be or vice versa. According to Chowdhury et al., 2013 studies, crosstalk noises can be influenced by various physiological parameters, such as the thickness of the layer beneath the skin which can increase these noises since fat can insulate and alter the EMG signals making them harder to capture accurately. The size of electrodes and the distance between them also play a big role in this type of noise, but by carefully choosing the correct parameters of the electrodes Chowdhury et al., 2013 arrived to the conclusion that the best size for the electrodes should be 10 x 1 mm, while the distance between electrodes should be 1-2 cm and combining it with the correct mathematical differentiation the crosstalk can be minimized. The primary cause of crosstalk is the generation of non-propagating signal components, which can occur due to the loss of action potential in the tendons. Crosstalk doesn't work like other interferences, it has its unique shape and bandwidth when compared to direct muscle signals, differently from other interferences cross-correlation and high pass filters methods are not reliable to reduce it, making it challenging to simply filter it out. Studies by Chowdhury et al., 2013, demonstrated that to reduce crosstalk noises, surface electrodes should be applied in the minimal crosstalk area

(MCA) which is specifically identified as the regions where the interference from co-contracting muscles is the lowest. To pinpoint the MCA of a targeted muscle its necessary to accurately measure the distance between two anatomical reference points, or bony landmarks. Once the MCA is determined it can effectively reduce interferences from adjacent muscles. Mezzarane and Kohn, 2009 presented a mathematical algorithm that represented the relation between the targeted muscle and the crosstalk.

$$\mathbf{Tb}^2 = \mathbf{Rb_c}^2 + \mathbf{Ti}^2 + \mathbf{Ob_c}^2 \tag{2}$$

Equation (2) explains the baseline EMG activity recorded at the target muscle (represented by **Tb**) which is a combination of the inherent activity of the targeted muscle itself (represented by **Ti**), the crosstalk from the remote muscle (represented by $\mathbf{Rb_c}$) and the crosstalk from other muscles (represented by $\mathbf{Ob_c}$). Assuming that the signals **Ti**, $\mathbf{Rb_c}$ and $\mathbf{Ob_c}$ are independent from each other, the variability in **Tb** is the cumulative result of the variant in **Ti**, $\mathbf{Rb_c}$ and $\mathbf{Ob_c}$.

Other type of inside noise that happens in our body its due to the influence of some factors that are inherent to the induvidual body, this factors termed internal noises include the number of muscle fibers in each unit, the depth and location of active muscle fibers and the amount and type of tissue present in the muscle, these factors can distort and alter the EMG signal reading making them harder to interpret with accuracy. Chowdhury et al., 2013 tells us that, when analyzing the effects of physical capacitance (the ability of a material to store electrical charges) are assumed to be negligible, but also refers that this assumption is not always true for muscle fibers, both conductivity (how good is a material to conducts electric field) and permittivity (how good is a material to store electric field) of the muscle depend on the frequency. Chowdhury et al., 2013 gives the example of the skin, which has low conductivity but high permittivity, meaning that capacitive effects, related to the storage of electrical charge, can be significant in the skin, contributing to the internal noise of the EMG signal.

The tissue's amount and thickness between the contracting muscles and the electrodes play a crucial role in determining the EMG signal's strength. Hemingway et al., 1995 presented that with a steady muscle contraction force, the thicker it was the subcutaneous (fatty) layer between the recording and the contracting muscle the weaker the capture EMG signal would be. This was observed in a group of 20 subjects with varying amounts of subcutaneous tissue, who were asked to contract their muscles over the period of time of 45 seconds. The presence of excess body fat can act as an internal noise for the EMG, this is due to the fact that the fat increases the distance between the active muscle fibers and the electrodes, making it harder to get a clear signal, reducing the fat layer surgically under the recording sites leads to an increase in the amplitude of the surface EMG signal. Other way to reduce these impendances without the need of going into surgery would be by using high pass spatial filters, these filters would allow signals to go above a certain frequency to pass through while attenuating signals bellow that frequency, this would not have the same effects as the surgery but can partly reduce the internal noise in a

more convenient manner.

2.4 Electromyography (EMG) Signal Processors

Having delved into the complexity of EMG noises, it becomes evident that the raw material captured from the EMG signals, while full of information, are often affected by multiple interferences. These signal noises can significantly distort the representation of the muscle activity readings, leading to misinterpretations. This highlights the main role of EMG signal processing. This process is not only a supplementary step but a foundation when we talk about translating raw muscle electrical activity into meaningful data that then can be analyzed. EMG signal processing serves multiple purposes such as noise reduction, feature extraction, dimensional reduction and pattern recognition.

In the realm of EMG studies, signal processing plays a multifaceted role. Firstly, given the susceptibility of EMG readings to various noises, it's imperative to employ processing techniques that can filter out these interferences, ensuring a valid data, beyond the mere capture of the signals. The essence of the accuracy of these readings lies in the extraction of some specific features. Whether it is in the time-domain, frequency-domain or the time-frequency representations, each feature offers a unique lens into the muscle behavior. Furthermore, the inherently vast and high-dimensional nature of EMG data necessitates dimensional reduction, streamlining the data for efficient analysis without compromising on its richness. Lastly, a pinnacle of EMG studies is the ability to discern patterns within muscle activities. Through signal processing, these patterns can be classified, paving the way for groundbreaking applications, such as prosthetic control, where a recognized muscles patterns are seamlessly translated into a specific movement command.

Electrodes attached to the skin surface record EMG signals, which reflects the electrical activity of the neuromuscular system. According to Zhang et al., 2010, these generated electrical signals usually have an electrical activity amplitude ranging from 0.1 to 5.0 mV. The energy of these EMG signals is primarily found within the 0 to 1000 Hz frequency range. However, raw EMG signals often contain unwanted interference. This interference can manifest as low-frequency signals (close to 0 Hz often referred as Direct Current (DC)) and high-frequency signals beyond the 1000 Hz.

The most valuable and informative portion of the EMG signals lies between 10 – 500 Hz. More specifically, the frequency range of 50 – 150 Hz is of particular interest because it captures the primary characteristics of muscle activity. The noise in EMG readings is typically attributed to the high-frequency signals. These high-frequency interferences can distort the true representation of muscle activity, making it essential to filter them out for accurate analysis and interpretation.

In essence, EMG signal processing serves as a bridge that connects the raw, noisy data to insightful information. It refines the data, extracts the essence and presents it in a format ready

for interpretation.

In the last decade, the wavelet technique has emerged as a preferred method for time-frequency analysis. The strength of the wavelet transform (WT) method lies in its ability to provide a time-frequency representation, focused in capturing non-stationary characteristics of the signals making it especially valuable in context like EMG analysis. WT is a method that provides a more detailed view of the signals, by decomposing them into different bands. Features can then be extracted from each of these different bands.

Zhang et al., 2010 sates the WT time-frequency analysis is more apt for managing the non-stationary properties of EMG signals. Based on the affirmation made by Zhang et al., 2010 its commonly understood that noises primarily originate from high-frequency signals, which are typically considered to follow a Gaussian distribution. The subsequent equation (3) offers a basic representation of the EMG signal:

$$f(t) = s(t) + n(t) \tag{3}$$

Equation (3) states that the observed EMG signal $\mathbf{f}(\mathbf{t})$ is the sum of the actual muscle electrical activity $\mathbf{s}(\mathbf{t})$ and any noise $\mathbf{n}(\mathbf{t})$ that might be present.

Zhang et al., 2010 says that the wavelet coefficients of $\mathbf{n}(\mathbf{t})$ it is consistent across scales, but their amplitude diminishes as the scale expands. Given that noise coefficient is more prevalent in smaller scales and EMG signals in larger scales, wavelet denoising focuses on employing a precise method to determine the transform coefficient of the raw signals, eliminating the noise-induced spectral component. In order to perform a wavelet de-noising Zhang et al., 2010, suggests dividing it in several steps. Starting by performing a multi-scale decomposition to examine the wavelet coefficients of the signals. Next, estimate the noise level and select an appropriate threshold. Apply this threshold to the wavelet coefficients. Finally, use these revised coefficients to reconstruct a clearer version of the EMG signals.

Based on the findings of the Lolure and Thool, 2015 WT can be subdivided into two main types, discrete wavelet transform (DWT) and continuous wavelet transform (CWT). As per Chowdhury et al., 2013, the DWT method processes signals more quickly due to its use of down-sampling, which reduces the data volume at each decomposition level. In contrast, the CWT provides more detailed analysis with uniform resolution across all scales, but this comprehensive processing makes it less time-efficient than DWT. While DWT is effective in analyzing non-stationary signals like surface EMG, it generates a feature vector with high dimensionality, which can sometimes complicate subsequent analysis and classification tasks. Conversely, the CWT's ability to offer a detailed and flexible representation of signals makes it suitable for applications that demand high precision and granularity in the time-frequency domain.

To further delve into the intricacies of the DWT, it's essential to understand its foundational principles. The DWT is a tool used to analyze signals at different frequency bands, allowing

for a multi-resolution analysis. This means that a signal can be observed on both broad and fine scales. The DWT achieves this by limiting the changes in translation and scale, typically of powers of 2.

The DWT is calculated through consecutive low-pass and high-pass filtering in the discrete-time domain. The foundational formula for DWT is represented by:

$$\mathrm{DWT}_{x(t)}[k,l] = \int x(t)\psi_{k,l}(t)\,dt \tag{4}$$

Equation (4) states that $\mathbf{DWT_{x(t)}[k,l]}$ represents the DWT coefficient of the signal $\mathbf{x(t)}$ at a specific scale $\mathbf{k}$ and position $\mathbf{l}$; each coefficient provides information about the signals behavior at a particular frequency band and time location. The integral symbol represented by $\int$ this is used to compute the inner product between the signal $\mathbf{x(t)}$, which represents the EMG signal as a function of time $\mathbf{t}$, and the wavelet function $\psi_{\mathbf{k,l}}(\mathbf{t})$. This operation essentially measures the similarity between the signal and the wavelet at a specific scale and position.

Its also important to note that:

$$\psi_{k,l}(t) = 2^{k/2}\psi(2^k t - l) \tag{5}$$

Equation (5) represents a formula for the scaled and translated wavelet. The scaling and translation allow the wavelet to capture signal details at different frequencies and time locations. ψ is a mathematical function that has certain properties suitable for signal analysis, the subscript $\mathbf{k}$ corresponds to the analysis of the signal at a higher frequency while the subscript $\mathbf{l}$ represents the position or location in the signal.

In order to denoise an EMG signal through the DWT process, a suitable wavelet, also called a mother wavelet, needs to be chosen. This wavelet will be used to analyze the signals at various scales or resolutions. There are multiple wavelets that can be chosen for this process, such as Daubechies, Symlets, and Coiflets among others. Besides choosing the appropriate mother wavelet, it's also important to choose a threshold to be applied to the wavelet coefficients. Coefficients with absolute values below this threshold are set to zero, effectively removing them. As with the mother wavelets, there are also different methods to determine the threshold value, some of these methods are the Steins Unbiased Risk Estimate (SURE) threshold and Bayes threshold.

After choosing the mother wavelet and the threshold method, it's important to filter the EMG signals. The DWT uses two sets of filters: a low-pass filter and a high-pass filter. These filters are derived from the chosen mother wavelet. Once the EMG signal is passed through both filters, the low-pass filter provides the approximation coefficient, which represents the general trend of the signal. The high-pass filter provides the detail coefficient, which represents the high-frequency component of the signal. According to Sobahi, 2011, after filtering, the resulting signals are down-sampled by a factor of 2. This means that only every second sample is kept. This process reduces the number of coefficients by half for each filtering operation, making the transform

computationally efficient.

Following the filtering and down-sampling process, its essential to delve deeper into the foundation principles that underpin the DWT such as the recovery transform, scaling, wavelet definition and filtering back techniques.

In the study of Sobahi, 2011, the author emphasizes that recovery transform is a crucial aspect of the DWT, allowing for the reconstruction of the original signal from its wavelet coefficients. The DWT is often introduced in terms of its recovery transform, which can be represented by the equation:

$$x(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k,l) 2^{-k/2} \psi(2^{-k}t - l) \tag{6}$$

Equation (6) represents the recovery transform in the DWT, where $\mathbf{x(t)}$ represents the original signal as a function of time $\mathbf{t}$, $\mathbf{d(k,l)}$ denotes the wavelet coefficients at specific scale $\mathbf{k}$ and position $\mathbf{l}$. ψ represents the wavelet function, which is used to analyze the signal at various scales and resolutions. The summation over $\mathbf{k}$ and $\mathbf{l}$ indicate that the transformation is computed across all scales and positions, capturing both the general trends and intricate details of the signals.

Sobahi, 2011, further elaborates on the scaling function, often termed the "smoothing function", which plays an important role in the DWT. It aids in the efficiency of computation by providing a smoothed version of the signals. The scaling function can be defined using a dilatation or a two-scale relation:

$$\Phi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2} c(n) \Phi(2t - n) \tag{7}$$

Equation (7) defines the scaling function, or smoothing function. $\mathbf{\Phi(t)}$ represents the scaling function as a function of time $\mathbf{t}$. $\mathbf{c(n)}$ are the coefficients that define the specific scaling function. The summation over $\mathbf{n}$ indicates that the scaling function is computed using a combination of these coefficients, providing a smoothed version of the signal.

Within this context Sobahi, 2011 explains that in the DWT, the wavelet is derived from the scaling function and is represented by the equation:

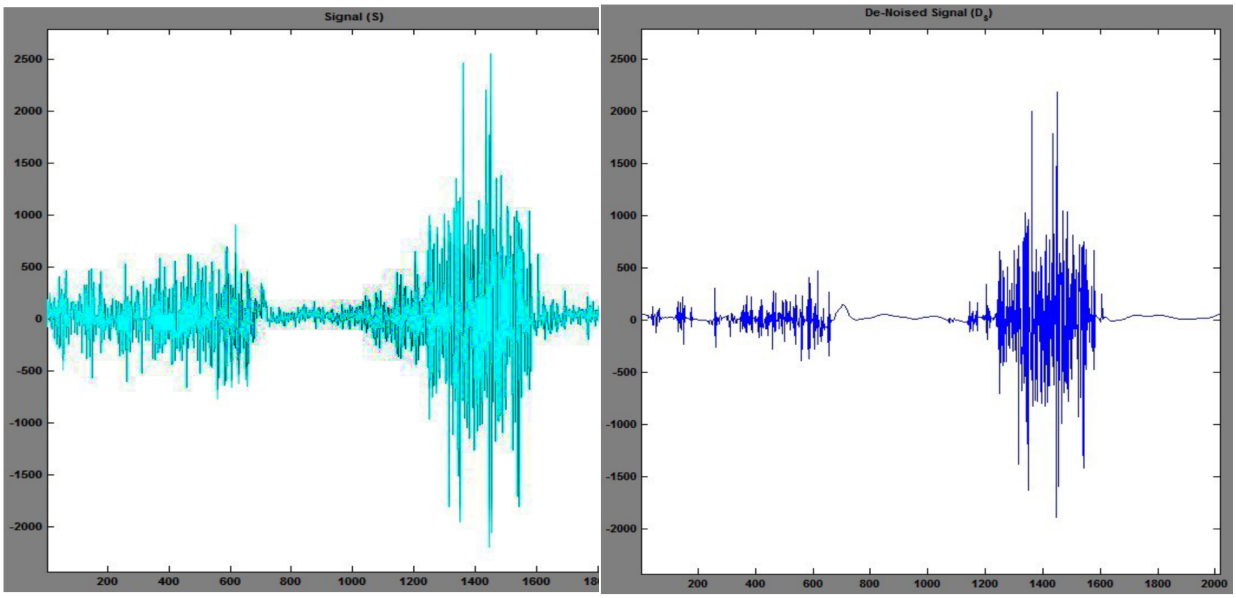
$$\psi(t) = \sum_{n=-\infty}^{\infty} \sqrt{2} d(n) \Phi(2t - n) \tag{8}$$

Equation (8) defines the wavelet function $\boldsymbol{\psi(t)}$ in terms of the scaling function $\mathbf{\Phi(t)}$. The wavelet function $\boldsymbol{\psi(t)}$ is represented as an infinite sum of scaled and shifted versions of the scaling function. Here, $\mathbf{\sqrt{2} d(n)}$ are coefficients that modulate each term of the series, where $\boldsymbol{d(n)}$ determines the contribution of each scaled and shifted scaling function, $\mathbf{\Phi(2t - n)}$. The summation over $\boldsymbol{n}$ from $\mathbf{-\infty}$ to $\boldsymbol{\infty}$ illustrates how the wavelet function is synthesized from a sequence of these modified scaling functions.

Besides these Sobahi, 2011 refers that beyond the mathematical equations, the DWT practical implementation often leverages the filter bank technique. This approach employs a pair of

filters, derived from the mother wavelet, to analyze the signal at different frequency bands. The technique efficiency lies in its ability to provide a multi-resolution analysis, capturing both the general trends and the intricate detail of the signal.

In the study made by Zhang et al., 2010, the method utilized was a Butterworth low pass filter of order 10 with a 500 Hz cutoff frequency in order to eliminate high-frequency noises. Additionally, the Sym2 wavelet was also chosen to break down the initial EMG into five levels. Subsequently the authors implemented a soft threshold de-noising technique using the minmax rule in MATLAB7. With these conditions Zhang et al., 2010, demonstrates the impact of the wavelet de-noising technique by comparing the raw EMG signal with the version processed through the WT (see Figure 8).



(a) Raw EMG Signal **(b)** Processed EMG Signal

Figure 8: Comparison of the raw EMG signal (a) and the processed through the WT (b) (image from Zhang et al., 2010 page 3).

From the comparative analysis, made from the of Zhang et al., 2010, its evident that the WT de-noising method effectively retains the peak and result image transitional aspects of the raw EMG signals, ensuring the preservation of the signals primary characteristics.

To gain a comprehensive understanding of the CWT, it's crucial to grasp its foundational principles. Chun-Lin, 2010, describe the CWT as a tool that provides a continuous representation of the signal in the time-frequency domain, allowing for a detailed analysis of its transient component. Unlike the DWT, which operates on discrete scale and position, the CWT offers a continuous view. This makes it particularly suitable for analyzing non-stationary signals like EMG, as emphasized by Arts and van den Broek, 2022.

The foundational formula for CWT, as presented by Belkhou et al., 2019, is represented by:

$$c_x(\tau, \sigma) = \int_{-\infty}^{+\infty} x(t)\psi_{\tau,\sigma}(t)\, dt \tag{9}$$

In equation (9), $c_x(\tau, \sigma)$ denotes the CWT coefficient of the signal $x(t)$ at scale τ and position σ. The integral symbol represented by $\int$, is used to compute the inner product between the signal $x(t)$ and the continuously scaled and translated wavelet function $\psi_{\tau,\sigma}(t)$. This operation measures the similarity between the signal and the wavelet function, continuously scaled by factor τ and translated by σ.

The continuous wavelet function can be defined as:

$$\psi_{\tau,\sigma}(t) = \frac{1}{\sqrt{\sigma}}\psi\left(\frac{(t-\tau)}{\sigma}\right) \tag{10}$$

In equation (10), $\psi_{\tau,\sigma}(t)$ represents the wavelet function that has been continuously scaled by factor τ and translated by factor σ. The term $\frac{1}{\sqrt{\sigma}}$ acts as a normalization factor, ensuring that the wavelet has a constant energy across all scales. The function $\psi\left(\frac{(t-\tau)}{\sigma}\right)$ is the basic wavelet, which is transformed by the scaling and translation parameters to analyze the signal at different scales and time locations. This equation provides a mechanism to generate wavelet of varying widths and positions, enabling the detailed analysis of a signal at multiple scales and time locations.

When analyzing EMG signals using the CWT, the choice of the mother wavelet is crucial. Different wavelets, such as the Morlet, Mexican Hat, or Gaussian wavelets, can be employed based on the specific characteristics of the signal and the analysis objectives.

The strength of the CWT lies in its ability to provide a high-resolution time-frequency representation, making it especially suitable for capturing transient or short-duration events in the EMG signals. This is particularly beneficial when analyzing rapid muscle contractions or detecting onset times of specific movements.

In the realm of EMG signal analysis, Arts and van den Broek, 2022, highlights the significance of the CWT in capturing the intricate details of muscle activations. The continuous nature of the transform ensures that no information is lost, making it a powerful tool for detailed signal analysis.

Beyond the mathematical formulations, the practical implementation of the CWT often involves the use of specialized algorithms and techniques to ensure computational efficiency and accuracy.

While both DWT and CWT have their unique strengths and applications, the choice between them depends on the specific requirements of the EMG signal analysis. For detailed, high-resolution time-frequency representations, especially for non-stationary signals, the CWT stands out as a powerful tool. However, for computational efficiency and multi-resolution analysis, the DWT is often preferred, as it provides a compact representation of the signal across different frequency bands and is good at capturing both the general trends and intricate details of the EMG signals.

2.5 Electromyography (EMG) Sensors

Initially, the plan was to personally collect EMG data. However, due to physical and ethical constraints, this approach had to be reconsidered. Despite the shift to using an open access dataset, discussing EMG sensors within the thesis' state of the art remained crucial. Given the limited information about the sensors used in the open access dataset, BITalino was chosen as an illustrative example. Renowned for its affordability and user-friendly design, BITalino is particularly notable in commercial settings for biosignal acquisition and analysis. As stated by Da Silva et al., 2014 the BITalino hardware was designed with a compact "Credit Card" form factor, incorporating various measurement sensors for both bioelectrical and biomechanical data acquisition. It's a handy tool for collecting and studying data in medical projects. What makes BITalino special compared to other EMG sensors is how easy it is to use, its low price, and its ability to change and adapt.

According to Da Silva et al., 2014, the default setup of the system arrives as a single board with the sensors already connected to the analog and digital ports on the control block. However, the design of the control, power, and communication blocks, along with the firmware, is entirely general-purpose. This allows individuals to utilize just the digital backend of the BITalino with their own tailored sensors and actuators design. Each distinct block can be physically separated from the main board, offering a variety of usage possibilities. As highlighted by Da Silva et al., 2014, this structure facilitates three configurations.

The first configuration is 'Board Configuration', represented in Figure 9 , the BITalino is used without any modifications, allowing individuals to easily experiment with the onboard sensors for prototyping activities or real-time observation of various physical phenomena.

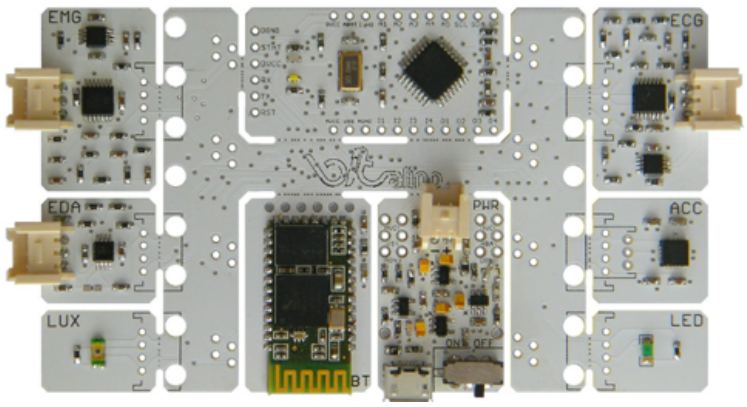


Figure 9: BITalino biosignal acquisition board configuration (image from Da Silva et al., 2014 page 2).

The second configuration is 'Plugged Configuration', represented in Figure 10, in this mode plugs are added to the BITalino and the individual sensor blocks are disconnected from the main board. This leaves only the control, power, communication, and auxiliary connectivity blocks, enabling individuals to use different sensors combinations interchangeably.

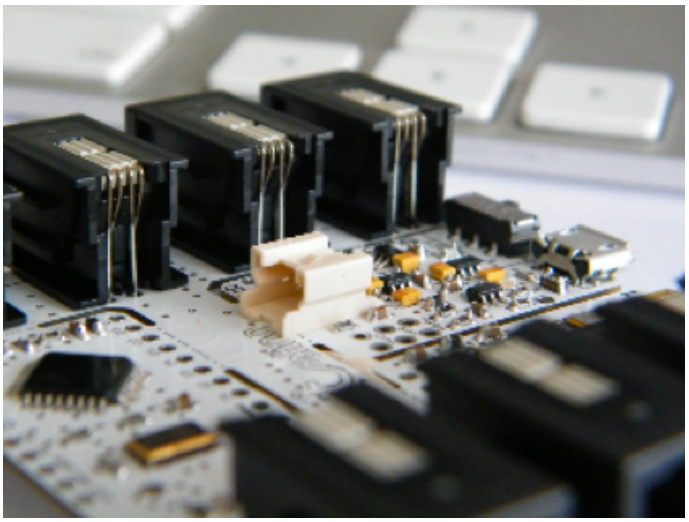


Figure 10: BITalino biosignal acquisition plugged configuration (image from Da Silva et al., 2014 page 4).

The third configuration is 'Freestyle Configuration', represented in Figure 11, in this configuration, all the individual blocks are detached from the BITalino main board, allowing individuals to mix and match them in any manner that aligns with their project ideas and applications

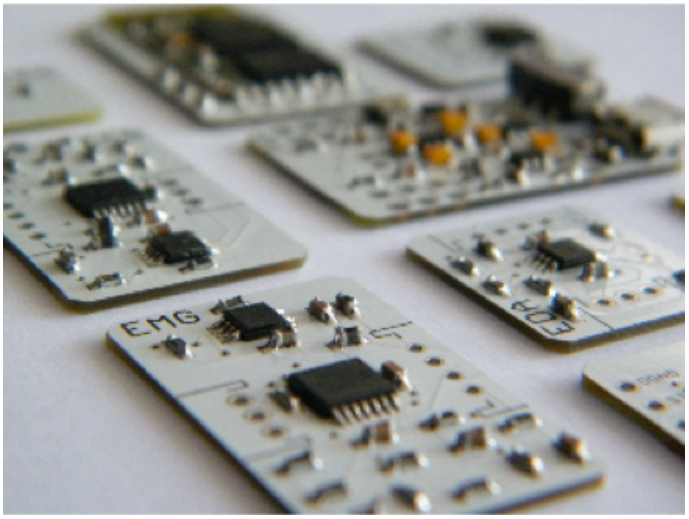


Figure 11: BITalino biosignal acquisition freestyle configuration (image from Da Silva et al., 2014 page 4).

BITalino comes with many sensors that can not only measure EMG signals but also other body signals such as, electroencephalography (EEG) signals, accelerometry (ACC) signals, and electrodermal activity (EDA), all the specifications of BITalino can be seen in Table 3. This part of the thesis will only focus on the EMG signal acquisition properties of BITalino.

According to Guerreiro et al., 2013 the EMG sensor operates on the principles of voltage potential differentials, which is the difference in electrical voltage between two points in a circuit. To capture the subtle voltage variations associated with these signals, which are typically in the millivoltage (mV) range, the sensor incorporates a precision instrumentation amplifier (InAmp). This amplifier offers a high common-mode rejection capability, registering at 110dB

when the gain (G) is 10 or more. Additionally, the sensor is equipped with high-speed, low-noise operational amplifiers (Op-Amp) that execute both bandpass filtering and signal amplification.

Table 3: BITalino specifications

Sampling Rate	1, 10, 100 and 1000 Hz
Analog Ports	4 inputs (10 bits) + 2 inputs (6 bits)
Digital Ports	4 inputs (1 bit) + 4 inputs (1 bit)
Actuators	LED
Sensors	EMG; ECG; EDA; ACC; LUX
Battery	3.7V Lithium Ion
Weight	30 grams
Size	105x60 millimeters
Data Link	Class II Bluetooth v2.0 with a range up to 10 meters

The block diagram of the EMG sensor circuit is visually represented in Figure 12, details the components and their interconnections within the EMG sensor system. At the heart of this system is the In-Amp with alternating current (AC) Coupling. This In-Amp is tailored to amplify the subtle electrical potentials associated with muscle activity. The inclusion of AC coupling ensures that the amplifier is attuned to pass only AC signals, thereby filtering out any constant DC voltages. This design choice emphasizes the amplification of the dynamic AC components of the EMG signal. To further enhance the signal quality, the system employs a Butterworth 4th Order Lowpass Filter. This specific electronic filter is characterized by its ability to allow low-frequency signals to pass while attenuating high-frequency noise and interference. Its 4th order nature denotes the filter's sharp transition between the frequencies, ensuring a flat frequency response in its passband. Another noteworthy feature of the circuit is its dedicated mechanism for processing common-mode voltage. This part of the circuit is designed to manage unwanted signals that appear identically on both the positive and negative inputs of an amplifier. It takes this common-mode signal, inverts it, and then redirects it back to the user through a designated reference electrode. This process aids in further reducing interference and noise, ensuring the clarity and accuracy of the measured EMG signal.

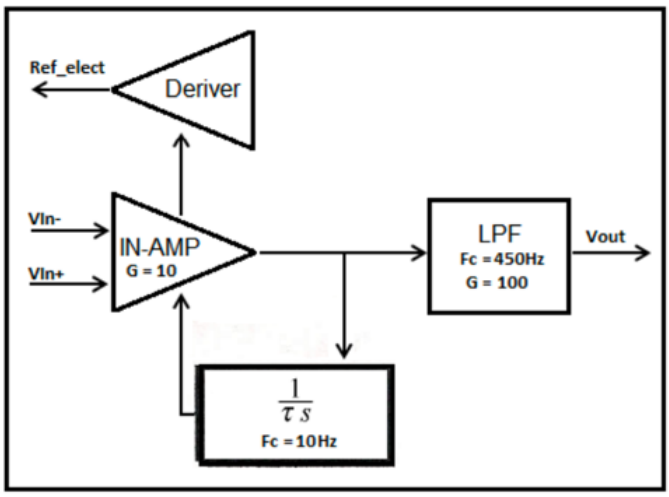


Figure 12: The EMG sensor block diagram features an In-Amp with AC coupling, designed to eliminate DC input voltages. This is succeeded by a Butterworth 4th order lowpass filter. Additionally, a circuit dedicated to processing common-mode voltage inverts the common-mode signal, redirecting it back to the user via the reference electrode (Image from Guerreiro et al., 2013 page 3).

The frequency response of the EMG sensor block represented in Figure 13 provides information about how the sensor responds to different frequencies of input signals. It can give insights into the range of frequencies the sensor can effectively capture and any attenuation or amplification at specific frequencies.

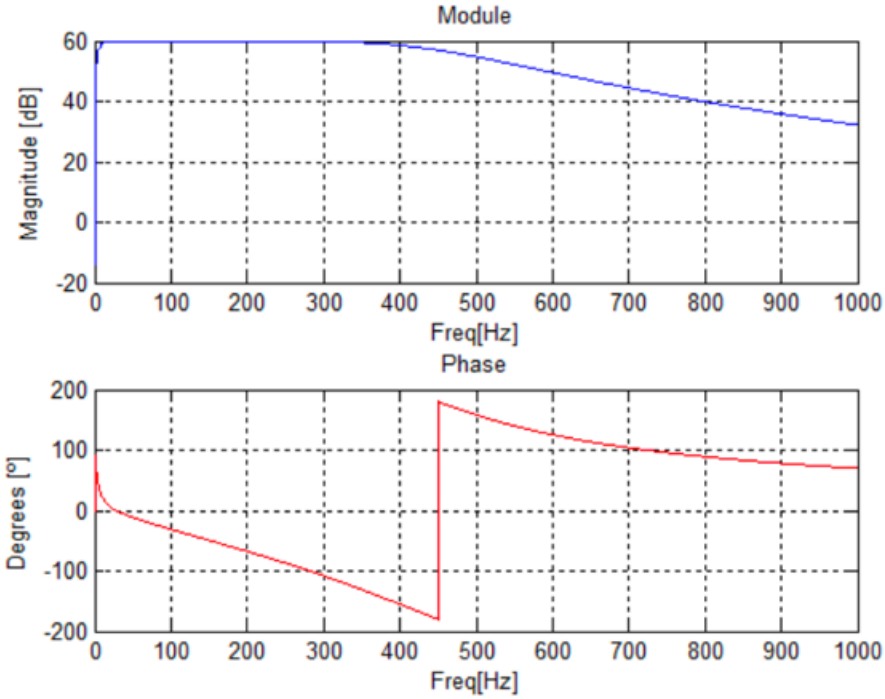


Figure 13: Frequency response of the EMG sensor block (Image from Guerreiro et al., 2013 page 3).

The transfer function for the EMG sensor is represented by Equation (11) mathematically describes the relationship between the input and output of a system. In the context of the EMG sensor, it would describe how the input EMG signal is transformed (amplified, filtered) to

produce the output signal. Equation (11) provides a comprehensive understanding of how the input voltages are processed to produce the final output voltage in the system.

$$V_{\text{out}} = (V_{\text{IN+}} + V_{\text{IN-}}) \times 1000 + V_{\text{ss}} \tag{11}$$

The equation (11) describes the output voltage, $V_{\mathbf{out}}$, of a differential instrumentation amplifier. In this equation, $V_{\mathbf{IN+}}$ represents the positive input voltage, while $V_{\mathbf{IN-}}$ denotes the negative input voltage. The difference between these two input voltages is amplified by a factor of 1000, indicating the gain of the amplifier. $V_{\mathbf{ss}}$ is the common-source or reference voltage for the system, which can be perceived as an offset voltage added to the amplified difference of the input voltages.

Experiments made by Guerreiro et al., 2013 shows that to accurately understand the analog EMG circuits functionality, the amplification level, or gain, was set to 100 times using an In-Amp. According to Guerreiro et al., 2013 this specific setting was chosen to ensure the resulting output voltage of the circuit consistently stayed within the controlled range of 0 to 3.3 volts. Figure 14 visually represents the frequency response of these EMG circuits, illustrating how they react to different input frequencies. A closer look at the plots in this figure (located on the top) indicates that the output signal has characteristics of a chirp wave, a type of signal that varies in frequency, with its amplitude fluctuating between its highest and lowest points approximately 3.2 volts. One standout feature of this circuit design is its ability to attenuate signals at both the very high and very low ends of the frequency spectrum. This selective frequency response is a direct result of the integrated bandpass filter, which is designed to allow only a specific range of frequencies to pass through while blocking out the rest. For testing and evaluation purposes Guerreiro et al., 2013, synthesized a chirp wave that was introduced to the circuit. This test signal had a frequency range from 0 to 500 Hz, a duration of 1 second, a peak-to-peak voltage of 28 millivolts, and a baseline set to half of the circuits supply voltage $V_{\mathbf{ss}}$. In order to ascertain the circuit responsiveness the author used a specialized test known as a transient analysis. The findings on this test from Guerreiro et al., 2013 revealed a slight delay in the circuits response, amounting to 0.146 seconds. According to the authors in a real-world scenario, the EMG sensor was applied directly to the biceps branchii, a prominent muscle in the upper arm, using electrodes pre-coated in gel. The data collected from this application can be viewed in Figure 15, showcasing the practical capabilities of the BITalino sensors.

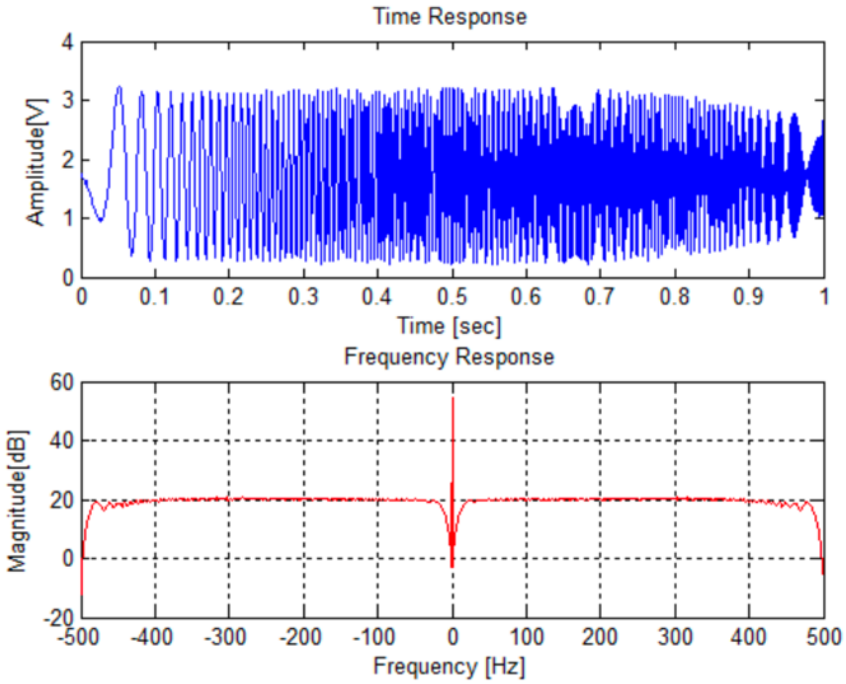


Figure 14: Frequency response of the EMG sensor (Image from Guerreiro et al., 2013 page 6).

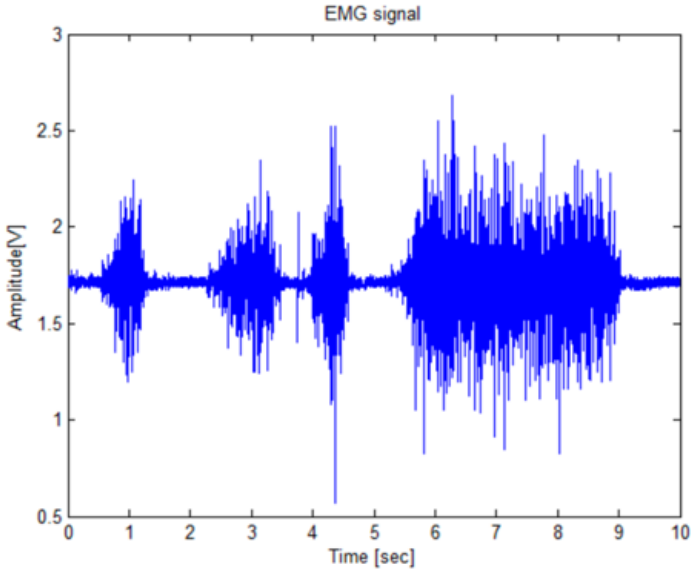


Figure 15: Example of an EMG signal (Image from Guerreiro et al., 2013 page 6).

BITalino is not only a hardware product but also comes with accompanying software. As stated by Stanek et al., 2018, this software displays the acquired voltage, facilitating real-time data collection and offline viewing. According to Guerreiro et al., 2013, the firmware governs the overall behavior of the system and manages data transmission via Bluetooth.

Figure 16 illustrates the general operation workflow. Figure 17 summarizes the process of modifying system settings by sending 1 byte commands from the base station to the device, delineating the modes and commands recognized by the system.

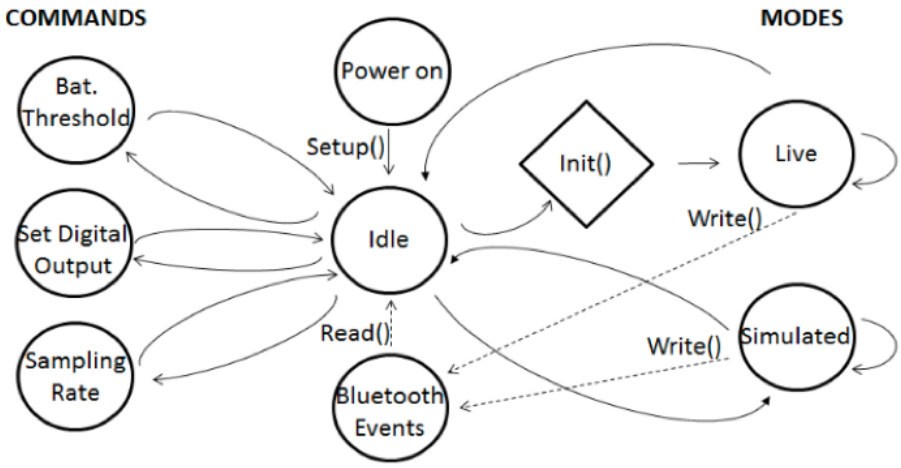


Figure 16: Firmware operation state diagram (Image from Guerreiro et al., 2013 page 4).

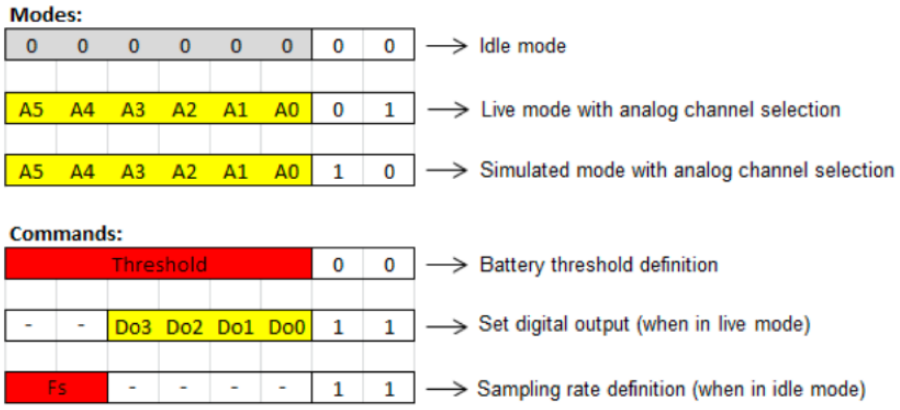
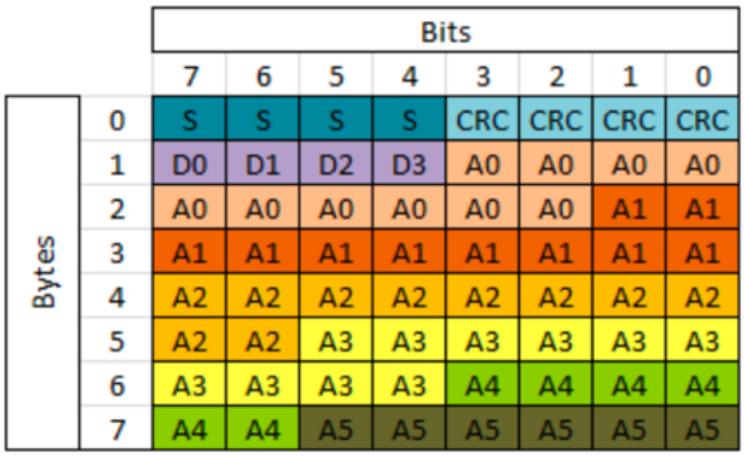


Figure 17: System operation modes and commands (Image from Guerreiro et al., 2013 page 4).

According to Guerreiro et al., 2013 the BITalino system operates in three primary modes, specifically in Idle, Live, and Simulated, depending on the user need. In Idle mode, the system exits any active mode and enter a stand-by state, awaiting a command to transition to either Live or Stimulated modes. This mode serves as a neutral state, pausing any active data acquisition or simulation. In Live mode, the system is designed to constantly monitor both analog and digital channels. It aggregates the data into distinct byte groupings before transmitting them through the Universal Synchronous/Asynchronous Receiver/Transmitter (USART) controller, a mechanism facilitating serial communication between various devices. As per Guerreiro et al., 2013, the optimization of packet size is essential to ensure the effective utilization of the communication channel's bandwidth, with the packet dimensions being dependent on the volume of channels being accessed in each cycle. The packet, with a size flexibility ranging from 3 to 8 bytes, and it includes a sequence number and a 4-bit Cyclic Redundancy Check (CRC) functionality grounded in Linear Feedback Shift Register (LSFR) methodology, aid in the identification and rectification of any discrepancies within the message. The composition of these packets leverages a bitwise operation technique, a process illustrated in Figure 18.

In the Stimulated mode, despite resembling Live mode, the system emulates data collection by transmitting fabricated signals. According to Guerreiro et al., 2013, this mode generates sinusoidal waves on the channels A2 to A4, square waves on channels D0 to D3, white noise following a normal distribution on channel A1 as shown in Figure 18. Guerreiro et al., 2013 also mentions that the structure of the data packet remains consistent with that of a Live mode, which facilitates the examination of the communication and interaction between the base station and the device without a user connection. This mode is particularly beneficial for developmental purposes, as it streams synthetic data instead of actual sensor data, thus serving as a valuable resource for testing or developmental projects. The configuration of these operational modes demonstrates the adaptability and user-oriented design of BITalino.



Bytes	Bits 7	6	5	4	3	2	1	0
0	S	S	S	S	CRC	CRC	CRC	CRC
1	D0	D1	D2	D3	A0	A0	A0	A0
2	A0	A0	A0	A0	A0	A0	A1	A1
3	A1	A1	A1	A1	A1	A1	A1	A1
4	A2	A2	A2	A2	A2	A2	A2	A2
5	A2	A2	A3	A3	A3	A3	A3	A3
6	A3	A3	A3	A3	A4	A4	A4	A4
7	A4	A4	A5	A5	A5	A5	A5	A5

Figure 18: Data packet structure (Image from Guerreiro et al., 2013 page 4).

Besides the operational modes of BITalino it's essential to understand the commands that take care of the functionalities and customizations of this system. These commands essentially serve as control knobs, allowing users to tweak the system settings in alignment with their specific requirements and project objectives. According to Guerreiro et al., 2013 among these commands, three of them stand out for their role in ensuring the effective operation of the BITalino system, namely Threshold, Set Digital Output, and Sampling Rate. The Threshold command serves the purpose of setting a specific limit for the low battery LED alert. Through the continuous monitoring of the battery voltage level via one of the analog input port, the system is designed to trigger the integrated red LED to illuminate once the battery voltage descend below the predefined threshold value. This feature acts as a visual indicator, notifying the user of a low battery sate. The Set Digital Output command is utilized to toggle the physical digital output ports between active and inactive states, guided by the specific outlined in the channel mask. This command is crucial for either interacting with external devices or signifying specific conditions within the system, showcasing BITalino capability of establishing connections with additional components or systems. Through this command, the flexibility of BITalino in managing digital outputs is distinctly highlighted, aiding in the broader control and signaling processes essential

for various applications. The Sampling Rate command establishes the rate at which the data is captured from the sensors. It essentially sets the pace for data acquisition, the options for this command are illustrated in Table 4. This ability to adjust the sampling rate gives the power to the user to find a middle ground between granularity of the data and the performance of the system. The provision of this command, along with the other commands highlights the adaptability and user focus architecture of the BITalino system.

Table 4: BITalino Sampling Rate definitions

Sampling Rate	1000 Hz	100 Hz	10 Hz	1 Hz
10^{Fs}	10^3	10^2	10^1	10^0

Choosing the BITalino platform as the EMG sensor for this study is substantiated by its affordability, user-friendly design, and its adaptability to various configurations, as highlighted by Da Silva et al., 2014 and Guerreiro et al., 2013. Its modular design facilitates tailored setups for precise EMG data acquisition, while its software enables real-time data collection and analysis. Additionally, the technical competence of BITalino in capturing and processing subtle EMG signals with high clarity and accuracy aligns with the objective of ensuring reliable data for analysis.

2.6 Electromyography (EMG) Classifiers

Once the EMG data is collected it is important to understand what type of EMG classifier is going to be used. There are multiple methods that can be applied so the algorithm can make the bridge between the raw EMG data and a prediction of the movement that the user wants to transmit throw the electrical activity generated by the muscle contraction, these can be composed by a single algorithm based system or a hybrid algorithm based system (Rupom et al., 2020). Some methods that can be applied for the EMG classifier can range from more traditional methods like k-Nearest Neighbors (KNN) and Support Vector Machines (SVM) (Rupom et al., 2020) to Convutional Neural Networks (CNN) (Li et al., 2023; Tuncer and Alkan, 2022) and Recurrent Neural Networks (RNNs) (Azhiri et al., 2021; Li et al., 2023). The choice of classifier depends on the needs of the project, as it determines the system's accuracy, speed, and robustness in interpreting the user's intent. Tests made by Rupom et al., 2020 concluded that the usage of a hybrid algorithm technique improved recognition and error correction. The author conducted a comparative analysis between systems based on single algorithms and the hybrid algorithms, focusing on their performance in terms of outcome percentages. Hybrid systems demonstrated a significant leverage comparing to individual classifiers and in this way obtaining an improved performance (Li et al., 2023; Rupom et al., 2020; Tuncer and Alkan, 2022).

KNN is a simple yet effective algorithm widely used in machine learning for both classification and regression tasks (Lyu et al., 2022; Rupom et al., 2020). The core concept of KNN revolves around the idea that similar data points are often in close proximity. When classifying a new sample, KNN looks at the "k" closest existing data points, known as "neighbors", and assigns the most common category among these neighbors to the new sample (Lyu et al., 2022). For instance, in a binary classification case, if more neighbors belong to Class A than Class B, the new sample is classified as Class A. This method is particularly beneficial when there is limited prior knowledge about the distribution of the data. KNN has been successfully applied in various fields, including signal processing of sEMG signals. According to Lyu et al., 2022 studies have demonstrated its effectiveness in classifying different motion signals and in prosthetic arm control, often achieving high accuracy and better performance in terms of processing time compared to some other methods, especially with lower signal-to-noise ratio (SNR) signals. Despite its simplicity and effectiveness, KNN has a notable drawback, it is sensitive to the local structure of the data, meaning that it can be influenced by noises in the nearest points.

Support Vector Machines (SVM) are a type of generalized linear classifier used in supervised learning for classification tasks and its one of the most used of the kernel learning techniques for non-linear classification methods (Lyu et al., 2022). The primary goal of SVM is to identify a decision boundary, or maximum-margin hyperplane, represented in Equation (12), that effectively separates data points belonging to different classes (Lyu et al., 2022). This decision boundary is not just a simple line but can be a plane or a hyperplane, especially in high-dimensional spaces.

$$w^T x_i + b = 0 \tag{12}$$

The Equation (12) describes the optimal hyperplane for SVM classification task (Rupom et al., 2020), in which w represents a weight vector, w^T denotes the transpose of vector w, ensuring the dot product with x_i is computationally feasible, x_i is a feature vector, an individual data point in the dataset, and b is the bias term which shifts the hyperplane away from the origin and its crucial to adjust its position. For classification equation can be changed to Equation (13) or Equation (14), depending on the prediction of the data point x_i.

$$w^T x_i + b \geq 0 \tag{13}$$

For equation (13) the data point x_i is predicted to belong to one class

$$w^T x_i + b < 0 \tag{14}$$

For equation (14) the data point x_i is predicted to belong to the other class.

SVM operates by mapping data to a higher-dimensional space where the decision boundary can be more clearly defined. It then focuses on maximizing the margin, which is the distance between the hyperplane and the nearest data points of each class (Lyu et al., 2022). The algorithm

uses the Hinge loss function to minimize empirical risk and incorporates regularization terms to enhance structural risk, leading to a sparse and robust classification model (Lyu et al., 2022). One of SVM's significant strengths lies in the fact that is one of the kernel methods, which allow it to efficiently handle non-linear classifications.

The effectiveness of SVM is not only limited to theoretical applications but extends to practical uses, such as in the classification of sEMG signals. Researchers Alkan and Günay, 2012 have successfully applied SVM in combination with discriminant analysis to categorize different arm motions, achieving high accuracy rates. Demonstrating SVM's capability to provide precise and reliable classifications in both statistical analysis and real-world applications.

Selecting the optimal kernel function parameters for SVM is a significant challenge that can substantially affect the model's accuracy. Addressing this challenge, a hybrid SVM-KNN model can be utilized in order to simplify the parameter determination process by integrating the KNN classifier (Rupom et al., 2020) . This hybrid model differs from the traditional SVM, which typically identifies a single optimized point to represent a class in the feature space (Rupom et al., 2020). Instead, it utilizes multiple representative points from the same class, enhancing the class representation and leveraging the combined strengths of both SVM and KNN, as shown in Figure 19.

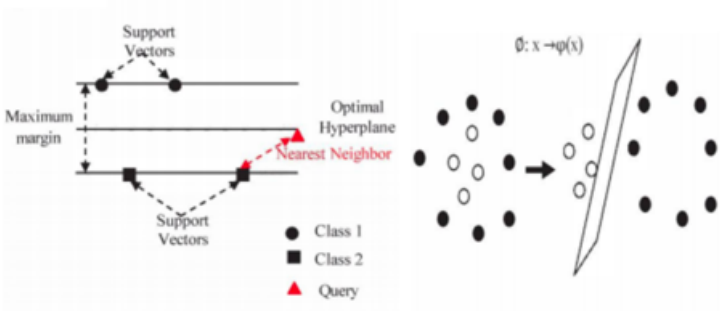


Figure 19: Representation of KNN-SVM classifier (image from Rupom et al., 2020 page 5).

The SVM-KNN hybrid employs the SVM's capability to sort and initially classify data by assessing the proximity of data points. When the proximity falls within a certain threshold, he KNN algorithm is activated, bringing its advantage in local classification to the forefront (Rupom et al., 2020). KNN, functioning as a local classifier, does not necessitate the selection of kernel parameters and adeptly manages non-linear classification challenges. It identifies the nearest reference neighbors, refining the exploratory attributes critical for class definition (Rupom et al., 2020).

The collaboration between SVM and KNN in this hybrid model offers a robust approach to classification (Rupom et al., 2020). It combines the SVM's proficiency in establishing an optimal hyperplane for global decision-making with the KNN's local decision-making process, reducing the reliance on precise SVM kernel parameters. This hybrid model not only improves classification by considering local data structures but also contributes to a more accurate and reliable kernel framework within SVM, thereby boosting overall classification precision.

Convolutional Neural Networks (CNNs) are a specialized deep learning architecture designed to process data that has a grid-like format, such as images or multi-dimensional time series (Li et al., 2023; Tuncer and Alkan, 2022). This deep learning subset uses multiple layers of non-linear processing to learn, extract features, and transform data efficiently. Over the past decade, CNNs have become the standard in image classification due to their ability to perform feature learning, particularly with two-dimensional input data (Li et al., 2023). This same feature learning process is applied to one-dimensional (1D-CNN) data sequences, in which 1D-CNNs skillfully apply feature learning directly to raw data, thus bypassing the need for manual feature engineering (Li et al., 2023).

The architecture of CNNs facilitates a hierarchical modeling of data, where the output of each layer serves as the input for the next, creating a depth of layers that allows the system to build complex features from simpler ones (Tuncer and Alkan, 2022). This layered approach allows CNNs to develop high-level features from low-level ones, thereby achieving multiple levels of abstraction that lead to a more effective data representation (Tuncer and Alkan, 2022). The fundamental components and functions used in CNNs contribute to this powerful framework, allowing for robust applications in machine learning without losing effectiveness regardless of the diversity of the input data (Tuncer and Alkan, 2022).

Due to the capacity of CNN to adapt to various data structures, when talking about EMG signal data, can benefit from changing the time-series data from this EMG signal acquisition to a spectrogram, like we can see in Figure 20, depending on what is beeing analyzed (Tuncer and Alkan, 2022). Time-series data would be better suited if the focus is on the raw temporal patterns, fluctuations, and the amplitude of the signals over time, and in this case the best approach would be the 1D-CNN, which are designed to process sequential data and can effectively identify local patterns within the time-series data (Tuncer and Alkan, 2022). In the other hand if the focus is in the signal intensity at various frequencies and times, the change from time-series data to a spectrogram could be useful since spectrograms are essentially images where different colors represent the different values of these metrics, and can reveal distinct features like dominant frequencies, patterns across time, and energy distributions which are often critical for signal classification task. For this method the best approach should be with CNN that can capture both spatial (frequency and time) and temporal (sequence of frames) features simultaneously (Tuncer and Alkan, 2022).

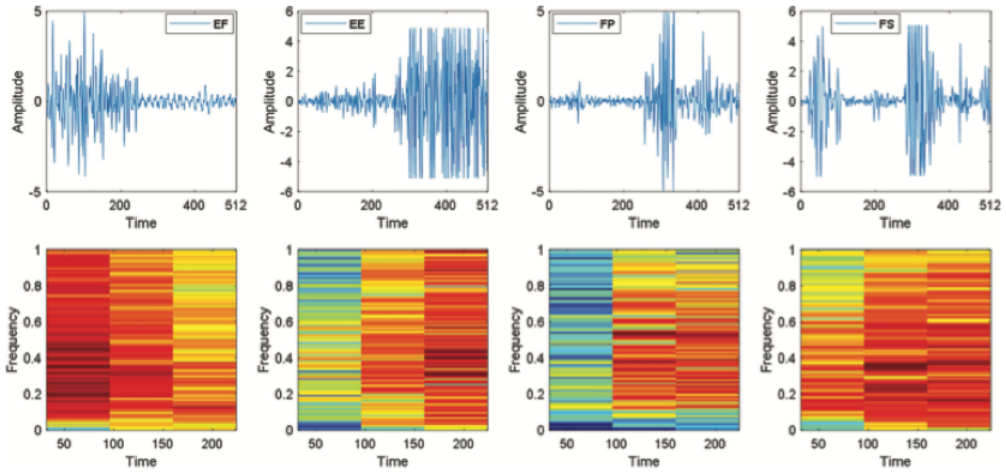


Figure 20: Raw EMG signals and spectrogram images for four different classes (image from Tuncer and Alkan, 2022, page 4).

Some features are more effective than other in representing data accurately (Tuncer and Alkan, 2022). Deep learning methods streamline this process by employing advanced algorithm for hierarchal feature extraction, which can automatically identify the most representative features of the data. This contrast with the traditional methods where features are often manually selected. Some of the fundamental layers and functions inherent to deep-learning architectures are the convolutional layer, Rectified Linear Unit (ReLu), pooling, fully connected, dropout, and softmax.

While the foundational principles of CNNs have revolutionized the field of machine learning (Tuncer and Alkan, 2022), the evolution of specific architectures has further pushed the boundaries of what is possible. Architectures such as AlexNet, which is a popular pretrained network has set new standards in the depth and complexity of neural networks, giving the fact that this architecture has learned to extract features that can generalize across various domains, including EMG signals.

The AlexNet model features a network architecture consisting of five convolutional layers spread with three max-pooling layers (Tuncer and Alkan, 2022). The convolutional layers employ filters of different sizes and utilizes the ReLu activation function. After the final convolutional layer, AlexNet includes fully connected layer, with outputs of 9216 and 4096, facilitating complex pattern recognition and classification tasks.

Hybrid CNN-SVM algorithms utilizes the feature extraction process of CNN with the classification skill of SVM (Tuncer and Alkan, 2022). This hybrid model combines the depth of CNN in identifying intricate patterns and characteristics from raw data, such as images or spectrograms, and pairs it with the SVM ability to define clear decision boundaries. The hybrid model aims to utilize the strengths of each approach, the CNN layers capture and transform the data into feature set, while the SVM layers act on these features to perform the final classification, often resulting in a better accuracy compared to using either method alone.

Figure 21 presents the architecture of a novel hybrid model that combines the strengths of

AlexNet-CNN with SVM for the classification of EMG signals. In this model, the conventional fully connected and softmax layers found in pretrained networks like AlexNet are replaced with an SVM classifier. This hybrid approach inputs the feature vector extracted from the spectrogram images by each CNN architecture directly into the SVM, following the max pooling of the AlexNet. The study by Tuncer and Alkan, 2022 outlines several training parameters, including a learning rate of 0.01 seconds, a maximum of 30 epochs, a mini-batch size of 128, and a validation frequency set at 50 Hz.

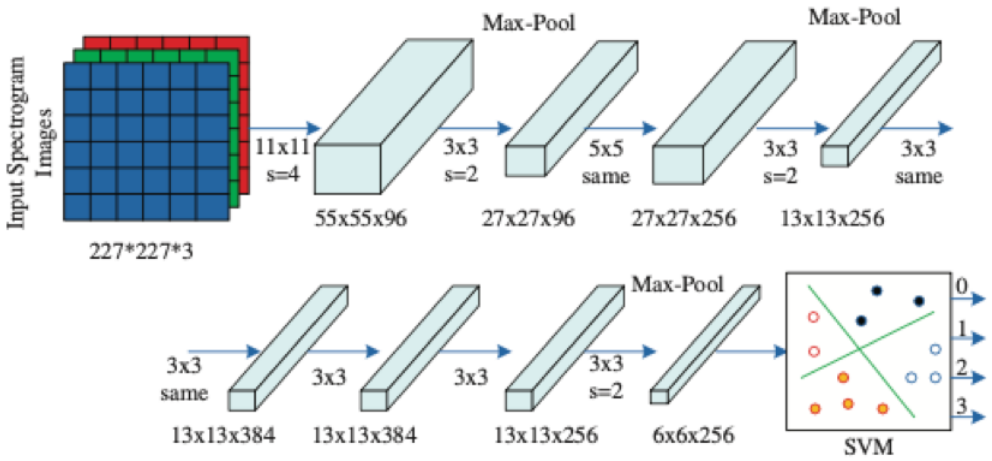


Figure 21: Hybrid AlexNet-SVM architecture (image from Tuncer and Alkan, 2022, page 6).

In the study conducted by Tuncer and Alkan, 2022, 70% of the 400 EMG signals were allocated for training, while the remaining 30% were set aside for testing. SVM classification utilized ten-fold cross-validation to ensure an equal distribution of training and test data across all classes. The authors assessed the performance of the CNN-SVM architecture using a confusion matrix, focusing on the results for the accuracy, precision, recall, and kappa statistics.

Tuncer and Alkan, 2022 concluded that the usage of AlexNet-SVM hybrid model tailored for the classification of four fundamental arm movements, achieved satisfactory classification results. The authors also added that the improvement of this model can be greater if there is an increment in the volume of training data. Despite this, the authors also concluded that the promising results underscore the potential of such hybrid architecture in developing assistive devices for patients with mobility impairments, like the loss of hand or arm function.

Recurrent Neural Networks (RNNs) are an integral part of deep learning, especially known for their effectiveness in processing sequential data, such as sEMG pattern detection (Li et al., 2023). As ilustrated in Figure 22, RNNs are distinct in their ability to modify their internal state through recurrent connections, a feature that sets them apart from architectures like CNNs (Azhiri et al., 2021). This ability enables RNNs to handle tasks with temporal dynamics, making them well-suited for managing time-dependent signals and processing sequences of varying lengths (Azhiri et al., 2021).

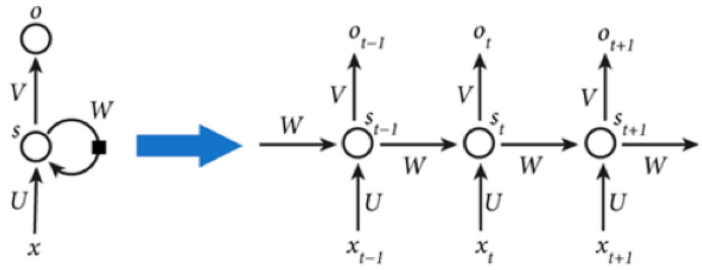


Figure 22: The RNN structural units unfolding over time (image from Li et al., 2023, page 7).

Figure 22 illustrates how the structural units of an RNN unfold over time (Li et al., 2023). On the far left, the RNN is shown in its condensed form, with a self-loop representing the recurrent connection that allows the network to pass information from one step to the next. This recurrent structure is then expanded into a sequential chain, where each node symbolizes the RNN's state at a given time step. In this sequence, x represents the input vector to the network at each time step t, s denotes the value of the hidden layer, and o indicates the value of the output layer. The matrix U corresponds to the weights from the input layer to the hidden layer, while V signifies the weights from the hidden layer to the output layer. The matrix W represents the weights applied to the hidden layer's output from the previous time step, reflecting the RNN's capacity to carry forward information through its hidden states. The RNN's unfolding over time shows that the value of the hidden state s_t at any time step is determined by both the current input x_t and the previous hidden state s_{t-1}, modulated by their respective weights. This dependency chain enables the RNN to effectively process and learn from sequential data, with each state influencing the next (Li et al., 2023).

In practical applications of RNNs, particularly in sequence classification tasks, the incorporation of a post-processing step is crucial to enhance the outcomes (Azhiri et al., 2021). This involves making class decisions for each segment of the sequence, such as windows of an EMG signal, followed by post-processing phase that minimizes ambiguities and reduces the likelihood of false misclassifications (Azhiri et al., 2021). The post-processing step plays a key role in synthesizing the results obtained from each segment, thereby significantly refining the overall efficacy of the RNN in handling sequence data.

The hybrid 1D-CNN-RNN model is designed to leverage the real-time data processing capability of 1D-CNNs with the temporal sequencing strengths of RNNs. This combination is proposed to enhance motion pattern recognition from surface electromyography (sEMG) signals, providing a balance between speed and accuracy (Li et al., 2023).

In the study conducted by Li et al., 2023, participants engaged in two sets of experiments, one offline and one online, to evaluate pattern recognition capabilities. The 1D-CNN-RNN model, was utilized to recognize finger and wrist movements in real-time, integrating feature extraction efficiency of CNNs with the sequential data handling of RNNs.

RNN has the critical issues of vanishing gradients and the need for long-term dependency management in sequence data processing. Li et al., 2023 also state that RNNs struggle with

retaining information over long sequences, as the data moves through each step, pieces of information are lost, diminishing the influence of early input on the network's current state. This inherent flaw makes RNNs less effective for tasks requiring long-term memory, as distant inputs become virtually inconsequential in the later stages due to gradient vanishing (Li et al., 2023).

The authors solution for this problem was utilizing Long Short-Term Memory (LSTM) network, an enhancement over the traditional RNN, with its specialized architecture that is a variant of RNNs, that excels in overcoming these challenges (Li et al., 2023). It is designed with mechanisms to remember information for extended periods, allowing it to maintain a trace of initial inputs throughout the network's progression. This makes LSTMs particularly good at handling tasks where long-term temporal dependencies are critical (Li et al., 2023).

Figure 23 showcases the architecture of the hybrid 1D-CNN-RNN model developed in the research conducted by Li et al., 2023. The authors divided the model in four distinct modules, the first two modules were convolutional, each with two layers of one-dimensional convolutions, ReLU activation, batch normalization, and max pooling, the second module including an additional dropout layer to mitigate overfitting. Following the convolutional modules, the third module features two LSTM cells with hyperbolic tangent function (TANH) activation and two dropout layers to boost the model's capacity to remember and process temporal sequences. The final module integrates a Dense layer, a Flatten layer, and a Softmax layer for classification. This architecture is crafted to capitalize on CNN's robust feature extraction and RNN's temporal memory, effectively addressing the challenge of time delays often encountered in CNNs (Li et al., 2023).

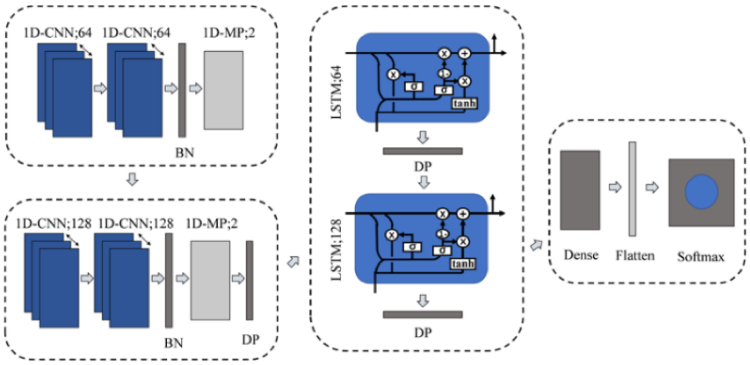


Figure 23: Diagram of 1D-CNN-RNN model (image from Li et al., 2023, page 9).

In the study conducted by Li et al., 2023, the efficacy of CNN, LSTM, and the hybrid 1D-CNN-RNN models was evaluated for classifying multi-channel sEMG signals associated with twenty distinct finger and wrist movements. The hybrid CNN-RNN model demonstrated superior performance, achieving an impressive 98.88% accuracy on the test set, thus outshining the standalone CNN and LSTM models. This model's ability to accurately recognize a wide range of motion patterns was confirmed, with over 97% accuracy, as illustrated in the confusion matrix in Figure 24. Among the models tested by Li et al., 2023 under identical preprocessing

conditions and hyperparameter settings, the CNN-RNN model emerged as the most effective for pattern recognition, with the standalone CNN model being the least effective.

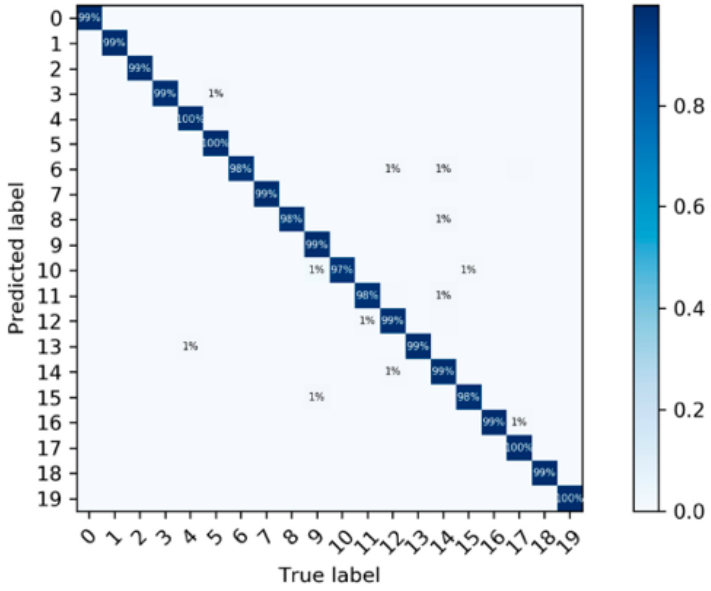


Figure 24: CNN-RNN model pattern recognition confusion matrix (image from Li et al., 2023, page 11).

The authors concluded that the hybrid 1D-CNN-RNN model's structure, which combines convolutional layers with LSTM units, effectively reduces input dimensionality, leading to more efficient computation and enhanced performance (Li et al., 2023). The integration of batch normalization within the CNN-RNN model facilitates feature extraction, while the inclusion of dropout layers guards against overfitting, yielding a robust model architecture.

In conclusion, the exploration of various machine learning and deep learning classifiers, including KNN, SVM, CNN, and RNN, along with their hybrid combinations, reveals different possibilities for EMG signal classification. The traditional methods like KNN and SVM offer simplicity and efficiency, especially in cases where the computational resources are limited or the dataset characteristics are well-suited to these algorithms (Alkan and Günay, 2012; Lyu et al., 2022; Rupom et al., 2020). On the other hand, advanced deep learning models such as CNNs and RNNs, and particularly their hybrid forms, demonstrate a remarkable ability to handle complex, high-dimensional data of EMG signals. These models excel not only in feature extraction but also in capturing temporal dynamics, which is crucial for accurate motion pattern recognition (Li et al., 2023).

The hybrid models, such as CNN-SVM and CNN-RNN, bring together the feature detection capabilities of CNNs and the classification and temporal pattern recognition strengths of SVMs and RNNs, respectively (Li et al., 2023; Tuncer and Alkan, 2022). These hybrids have great potential in enhancing accuracy, reducing overfitting, and improving the generalizability of the models. They incorporate a significant advancement in the field of EMG signal analysis, opening new avenues for more accurate and reliable prosthetic control.

Chapter 3

Methodology

The data collection is an essential step to make an accurate classification of the movement, while devices such as BITalino are effective for EMG signal acquisition, the practicalities of collecting high-quality EMG data present significant challenges. These challenges include a lengthy data collection period, estimated to range from weeks to months with multiple trials, as well as the meticulous preparation of participants.

Given these considerations, along with the extensive time and effort required for collecting and preparing data, the decision to utilize an existing high-quality dataset was deemed strategic and effective. This led to the adoption of the open-access GRABMyo dataset developed by Pradhan et al., 2022. The GRABMyo dataset has already undergone the extensive data collection and participant preparation process, thereby ensuring the dataset's quality and reliability for our study. This choice allowed to bypass the logistical complexities of primary data collection, focusing instead on the analysis and application of the data to real-world scenarios.

This strategic decision aligns with the research objectives to conduct a detailed and efficient analysis using reliable data that utilizes real-world muscle activation patterns. Using an established dataset accelerates the research process and ensures that the findings are based on robust and comprehensive data, enhancing the validity of the model prediction capabilities and the ablation study results.

The GRABMyo dataset (Pradhan et al., 2022) was created with inrtention of providing an open-access dataset to develop customized gestures as passwords to enhance security. The focus is on the development of wearable biometric devices, such as wristbands and bracelets, for both clinical and consumer objectives. The idea underlying the creation of this dataset came with the fact that the authors noticed limitations in the existing available datasets, which faced small subject pools or brief data collection these aspects, which are problematic in multi-day scenarios, where factors such as electrodes position shifting and varying skin conditions could impact EMG signal consistency. Recognizing these gaps, the authors decided to create this open-access dataset which stands out as one of the largest EMG dataset to date (Pradhan et al., 2022).

A total of 43 participants were enrolled (23 males and 20 females), with an age ranging

between 24 to 35 years old and an forearm length from 23.41 to 26.89 cm, measured from the wrist styloid to the elbow olecranon. Subsequently the authors positioned the electrodes between those areas; in total 28 electrodes were placed, 16 on the forearm and 12 in the wrist, these electrodes were placed with an inter-electrode spacing of 2cm apart from each other, measured center-to-center. To ensure uniform electrode placement across participants, the first electrode in each of the ring division was aligned with the elbow crease centerline.

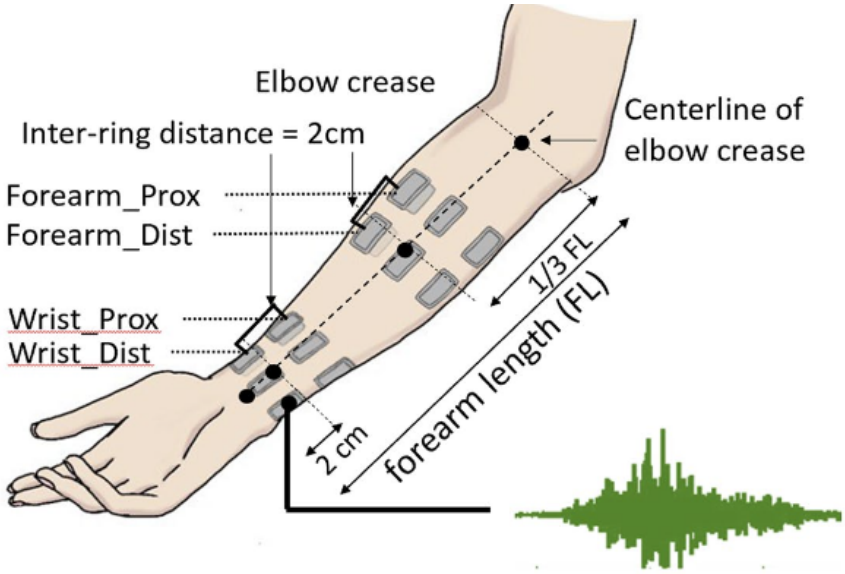


Figure 25: Electrode position (image from Pradhan et al., 2022 page 2).

During the experiments the participants sat comfortably with theirs arms in a resting position. The participants were conducted to perform 16 hand and wrist gestures, and after each gesture, they were instructed to go back to the resting position, performing a total of 17 gestures. The participants did 3 trials during a time period of 29 days, totaling 129 recordings in total with each participant making 119 contractions (Pradhan et al., 2022).

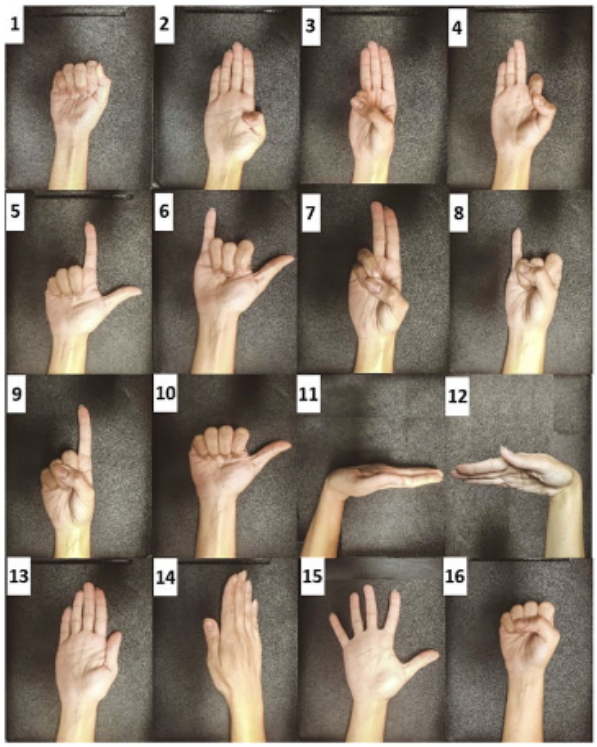


Figure 26: Gesture list (image from Pradhan et al., 2022 page 2).

While the GRABMyo dataset was initially created for developing password-based gesture recognition (Pradhan et al., 2022), its extensive and high-quality data align perfectly with the objectives of this research. Utilizing GRABMyo mitigates the challenges of primary data collection, allowing the focus on algorithm comparison and understanding the impact of muscle contraction features on movement classification. This strategic choice not only ensures data quality and reliability but also addresses ethical concerns in data gathering, enhancing the feasibility and relevance of the study for adaptive prosthetics.

Having established the GRABMyo dataset as a robust and reliable source for EMG data, the next steps that needs to be done is to rearrange the dataset in order to facilitate the analyses of the data and visualize the data. The first step involves a Python script to reorganize the dataset. This process was essential to align the data in a format conducive to efficient processing and analysis.

The dataset was organized into two main folders: **train** and **test**. The **train** folder, intended for algorithm training, contains data from the first EMG signal collection, including participants 1, 3-7, 9-11, 13-24, 26, 28-32, 34-38, 40, and 42, totaling 34 participants. The **test** folder is designed for algorithm evaluation and includes data from a separate EMG signal collection, featuring participants 2, 8, 12, 25, 27, 33, 39, 41, and 43, totaling 9 participants. This division ensures a clear separation of data for training and validation purposes.

Each of these folders are then divided into 4 different folders, **General**, **Session1**, **Session2**, and **Session3**. The **Session** folders gather the EMG data collected between each session and the **General** folder includes all the gestures from the participants in the 3 different sessions. The

General folder is then subdivided into 17 different gestures folders, and each containing the data gathered of each participant for that specific gesture during all the 3 trials On the other hand the **Session** folder is subdivided into the participants and then divided by other 17 folder corresponding to each gesture performed in the correspondent session. This format simplifies the choice of participants for each phase and also facilitates the process of training and testing the algorithm.

Once the dataset is reorganized it was time to visualize and process the data. To accomplish this, the BioSPPy library played a crucial role. This powerful Python toolkit is an open-source library designed for biomedical signal processing (Carreiras et al., 2015) and provides a suite of functions and utilities that were instrumental in both visualizing the EMG data and preprocessing it for further analysis. BioSPPy visualization capabilities are able to plot EMG signals for each gesture and channel, providing a clear view of the signal patterns and characteristics. This visual inspection was a key point to understand the nature of the EMG data, including its variability and the distinct features associated with different gestures. BioSPPy includes functions for effectively filtering the EMG signals providing advanced filtering techniques to remove noise and artifacts from EMG signals. This includes a bandpass filter to isolate the frequency range relevant to muscle activity and notch filtering to eliminate PLI. By applying band-pass filter its possible to obtain a clean and accurate data processed. An essential step in EMG signal processing is to extract meaningful features from the data; BioSPPy facilitates the extraction of different temporal and frequency domain features, such as mean absolute value, waveform length, and power spectral density. These features were crucial for the subsequent classification task, as they encapsulate the essential information contained within the EMG signals.

By leveraging the capabilities of BioSPPy, the original raw EMG data from the GRABMyo dataset was transformed into a format that was not only visually interpretable but also of great value for the integration of appropriate classification algorithms.

After reorganizing and processing the GRABMyo dataset, the next critical step in our research involves selecting the most suitable classifier algorithm for analyzing EMG signals. This decision significantly impacts the accuracy, speed, and robustness of gesture classification from the EMG data. Various classifier algorithms, ranging from traditional methods like KNN and SVM to more advanced deep learning architectures such as CNN and RNN, have been evaluated based on a literature review and preliminary tests.

KNN is simple and has been efficient in various motion signal classifications, it struggles with large, complex datasets like GRABMyo. Its performance deteriorates due to noise and data structure complexity, making it difficult to accurately classify 17 different gestures from the information from the 28 electrodes palced in the participants. This complexity is aggravated by the high dimensionality of the data and the inherent variability in EMG patterns between different participants. Such factors can significantly impact the algorithm's ability to distinguish between different movements, posing additional challenges in achieving accurate classification

SVM is known for effective nonlinear classification, SVM faces challenges with the large,

high-dimensional GRABMyo dataset. Its computational complexity increases with larger datasets, leading to extended training times and practical limitations in real-time applications. Moreover, adapting SVM for multi-class scenarios involves additional complexity, which can exacerbate computational burdens.

CNNs are suitable for the time-series data like EMG signals due to their ability to learn features directly from raw data. The convolutional layers in CNNs perform critical operations to filter input data, extracting key patterns such as waveforms and frequency characteristics from EMG signals. This automatic feature extraction presents a significant advantage over traditional methods requiring manual engineering, allowing CNNs to identify the most relevant features for gesture classification. The hierarchical nature of these networks enables them to learn feature at multiple levels of abstraction, from basic signal patterns in lower layers to complex gesture-specific features in deeper layers. Additionally, CNNs can be employed as 1D-CNNs for temporal pattern analysis or as 2D-CNNs for handling multi-channel image data, such as RGB images, effectively capturing both spatial and temporal features. RNNs excel in handling sequential data, making them particularly well-suited for EMG pattern recognition. Their defining feature is the ability to maintain a memory of previous inputs through internal state feedback loops and effectively process variable sequence lengths. This capability is crucial for EMG signals, since the order and timing of muscle activation play a big role in the process. The architecture of RNNs enables them to model complex relations within the data, learning both simple and intricate features of muscle activity. RNNs often incorporate post-processing steps to refine classification accuracy, reducing ambiguities and minimize misclassifications. Traditional RNNs can face challenges with long-term dependencies, but advanced variants like LSTM can address this issue maintaining the information over longer sequences.

Considering the strengths and weaknesses of each classifier, CNNs are chosen for their superior ability in automatic feature extraction and hierarchical learning, which significantly reduces the need for manual feature engineering. The architecture of CNNs allows them to effectively learn multi-level features through layers that mimic a form of hierarchical pattern recognition. This capability is critical for the accurate and robust classification of EMG-based gestures, where the subtleties of muscle activity must be precisely captured and interpreted.

CNNs are adaptable in managing both spatial and temporal data representations, configurable as either 1D-CNNs or 2D-CNNs. This adaptability makes them particularly effective for handling the spatial-temporal dynamics of EMG signals. 1D-CNNs are adept at analyzing time-series data by capturing temporal dependencies and sequence patterns. In contrast, 2D-CNNs can process multi-channel data like RGB images or spectrograms, allowing for the analysis of spatial relationships and features within the data. This dual capability is advantageous for EMG signal analysis, which can benefit from both temporal sequence analysis and spatial feature extraction from multi-channel setups.

CNNs are also favored for their scalability and efficiency in processing large datasets, such as GRABMyo. The ability to handle large volumes of data without a decline in performance

is essential, given the extensive data involved in EMG signal analysis for gesture recognition. This scalability ensures that CNNs can manage the growth in data volume efficiently, pivotal for real-world applications where timely and accurate gesture recognition is required.

Moving forward, the experiments will rigorously test the performance of CNNs, assessing their accuracy, robustness, and speed in classifying EMG signals. These tests are designed to evaluate how well CNNs meet the stringent requirements of prosthetic control applications in real-world scenarios.

As part of a comprehensive examination of CNN capabilities in EMG signal classification, this research incorporates an ablation study designed to dissect the network layer by layer. This methodical approach offers deeper insights into each component's contribution to the model's overall performance. By systematically removing certain layers or filters within the CNN and observing the impact on model accuracy, critical features essential for precise gesture recognition are identified. The methodology for this ablation study is adapted from Meyes et al., 2019 methodology, with modifications to enhance the granularity of information gathered from each layer. Notably, the scope of ablation extends beyond the typical 1%, 5%, 10%, and 25%, also including 50% and 75% to better determine the model's resilience and identify its critical threshold. Additionally, ablations are performed on the all the convolutional layer at the same time (general ablation) but will also be performed in individual layers to provide further insights into their hierarchical significance in maintaining performance accuracy. These individual layers undergo ablation at the same percentages as the general ablation.

The primary objective of this ablation study is to ascertain the importance of various features extracted by the CNN. This involves selectively disabling parts of the network, such as specific layers or filters, and measuring the resultant effects on the model's accuracy in classifying EMG signals.

Following the ablation, efforts will be made to recover the performance of the ablated models. This phase is critical as it tests the network's resilience and adaptability, providing insights into the redundancy and flexibility of the neural architecture. By retraining the network with the remaining active components and adjusting training parameters, there is an aim to restore or even enhance the original performance. This recovery process will also help identify whether certain ablated features can be compensated by other parts of the network or if they are irreplaceable. This methodical approach of ablation followed by recovery offers dual benefits. First, it provides a clear understanding of each component's importance, which is invaluable for refining the model. Second, it tests the network's robustness against potential losses of functionality, simulating scenarios where certain network parts might fail or perform suboptimally. The final stage of the research is to use Unity Engine to visualize the performance of the CNN used for EMG signal classification. By simulating the gestures predicted by the CNN in a virtual environment, Unity provides an interactive platform to visually assess the accuracy and practical utility of the model in real-time. This visualization is vital for demonstrating how the classified gestures correspond to actual human movements, thereby validating the model's effectiveness in applications such

as prosthetic control. The use of Unity helps bridge the gap between theoretical outcomes and practical implementation, offering a vivid demonstration of the technology's potential in enhancing adaptive prosthetics.

Chapter 4

Model Development

Advancements in prosthetic technology increasingly rely on sophisticated machine learning models to accurately interpret and classify sEMG signals for real-time control. However, the success of these models depends not only on their architecture but also on an understanding of which features most significantly contribute to accurate gesture classification. This study employs an ablation study on a CNN with an AlexNet architecture to gain critical insights into the model's classification process. By systematically removing and evaluating different layers within the network, it provides a deeper understanding of which features are more influential in accurately classifying sEMG signals. This understanding is pivotal in designing more targeted sEMG signal collection strategies. The experiment utilizes the GRABMyo dataset, a rich source of sEMG data featuring diverse hand gestures, to train a model with the AlexNet architecture. The choice of this dataset ensures a comprehensive evaluation under varied conditions, supporting robust training and validation of the proposed model. This section of the thesis details the experimental methodologies, from data reorganization and the transformation of sEMG signals into trainable formats to the rigorous implementation of the ablation study and its recovery. Each step is designed to incrementally reveal insights into the intricate relationship between EMG signal characteristics and their interpretability. Through these methodical experiments, the research seeks to provide a foundational analysis that not only enhances the performance of prosthetic control systems but also becomes more efficient in utilizing computational resources. It also offers a template for future investigations into the effective use of EMG data with neural networks in medical applications.

4.1 Dataset

The success of any machine learning model, particularly in the field of prosthetic control, is heavily dependent on the quality and comprehensiveness of the dataset used for training and evaluation. GRABMyo dataset is an open-access collection specifically designed for EMG signal research. Initially collected for use in gesture-enabled passwords, the GRABMyo dataset

features a comprehensive range of data from diverse participants performing various gestures. Despite its original purpose, the dataset's electrode placements on the forearm and wrist make it highly suitable for analyzing hand and finger movements, aligning perfectly with the objectives of this thesis. The GRABMyo dataset has a total of 43 different participants, the data was collected through 3 sessions with 7 trial per session. The dataset includes recordings of 16 distinct hand and wrist gestures plus a resting state, providing a rich variety of signals.

Each participant performed 16 distinct gestures which are represented in Figure 27 organized in a two-column format, with the order of listing from top to bottom, starting from the left column and then moving to the right column, plus a resting state. The data was recorded using 32 channels, with 16 electrodes placed on the forearm and 12 on the wrist, while 4 electrodes remained unused, as depicted in Figure 28. Specifically, the forearm channels are numbered from 1 to 16, the wrist channels from 18 to 23 and 26 to 31, with the unused channels being 17, 24, 25, and 32. The sEMG signals were captured at a sampling frequency of 2048 Hz, using a hardware bandpass filter ranging from 10 Hz to 450 Hz and a gain of 500. The data is stored in .dat and .hea file formats, where each .dat file contains the raw sEMG data for all 32 channels recorded during a session. These binary files are accompanied by corresponding .hea files, which provide metadata about the recordings, including the number of channels, the sampling frequency, and calibration details for each channel. This metadata is crucial for correctly interpreting the raw data stored in the .dat files.

TABLE
GESTURE LIST

Gesture	Description	Gesture	Description
	Lateral prehension (LP)		Index finger extension (IFE)
	Thumb adduction (TA)		Thumb extension (TE)
	Thumb and little finger opposition (TLFO)		Wrist flexion (WF)
	Thumb and index finger opposition (TIFO)		Wrist extension (WE),
	Thumb and little finger extension (TLFE)		Forearm pronation (FP)
	Thumb and index finger extension (TIFE)		Forearm supination (FS)
	Index and middle finger extension (IMFE)		Hand open (HO)
	Little finger extension (LFE)		Hand close (HC).

Figure 27: Gesture list with correspondent description. (Image from GestureList.JPG in the GRABMyo dataset Pradhan et al., 2022).

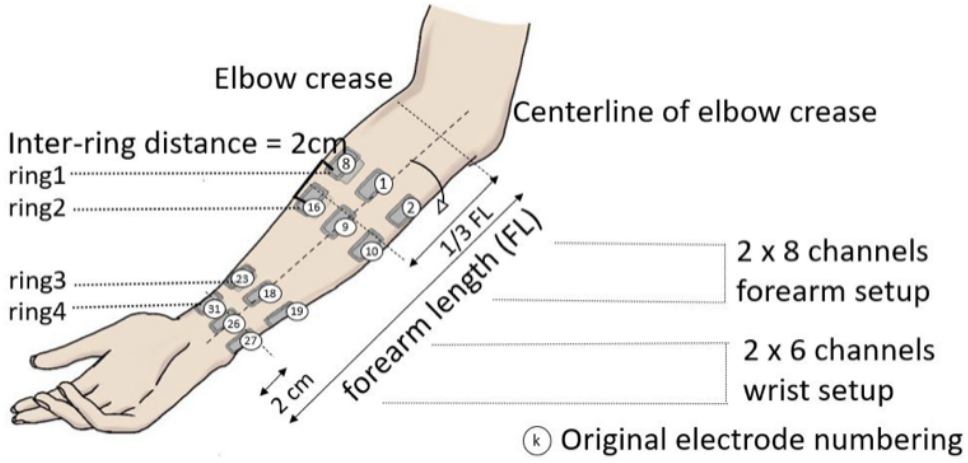


Figure 28: Electrode position schematics. (Image from Electrodelocation.pdf in the GRABMyo dataset Pradhan et al., 2022).

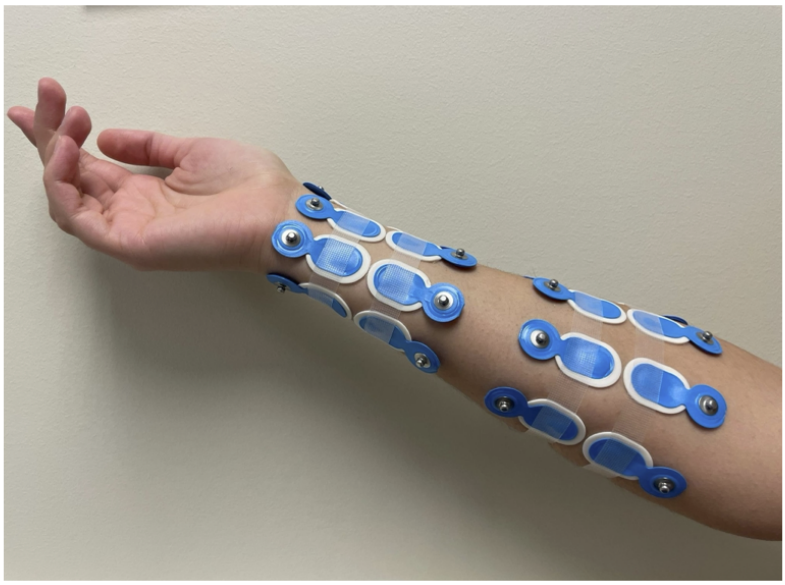


Figure 29: Real life electrode position. (Image from Electrodelocation.pdf in the GRABMyo dataset Pradhan et al., 2022).

Electrode placement on the forearm and wrist is critical as these areas exhibit clear muscle signal responses to hand movements. The electrodes on the forearm capture signals from muscles primarily responsible for moving the fingers and thumb, essential for performing the variety of gestures in the dataset. Meanwhile, the wrist electrodes capture movements and positions that involve the wrist itself. The electrodes are organized in multiple rings around the forearm and wrist, each maintaining a center-to-center distance of 2 cm. This structured arrangement allows

for systematic and uniform signal capture from the muscular activities involved in hand and wrist gestures, as shown in 29. Twelve monopolar sEMG electrodes, identical to those on the forearm, were arranged into two wrist rings. Each ring consisted of six electrodes, evenly spaced to form six bipolar pairs, ensuring comprehensive and balanced signal capture from the wrist's muscular movements, as described by N.Jiang et la., (2022). In a bipolar electrode configuration, each pair of electrodes measures the electrical potential difference between two closely positioned points above the muscle. This setup reduces interference from external noise and more accurately reflects the true muscle activity because it captures the difference in voltage caused by muscle fiber orientations under the electrode sites. This configuration is crucial for reducing artifacts in sEMG data, thus providing cleaner and more reliable signals for analysis. The consistent spacing and positioning, including the anatomical alignment of the first electrode in each ring at the elbow crease, ensure the reproducibility and accuracy of the recordings across different sessions and participants. With this configuration, the data collected is assured to effectively capture data from important muscles involved in hand and wrist movement such as the Abductor Pollicis Longus, affecting thumb movements; the Flexor Carpi Radialis and Extensor Carpi Radialis, important for wrist actions; and various flexors and extensors that control finger movements. These muscles, outlined in studies such as by Ngeo et al., (2014), play a significant role in hand and finger movement.

4.2 Data Processing

To facilitate robust training and unbiased evaluation of the model, the dataset was reorganized into training and test sets. The division was carefully planned to ensure that the model's evaluation was based on previously unseen data, thus eliminating any potential bias. The training set was composed by 34 participants while the test set was composed by 9 participants. This strategic partitioning allows for comprehensive training under varied conditions and supports a rigorous validation process.

To transform the .dat and .hea files into visual data, these files needed to be processed using matplotlib. Through this processing, the files were converted into time series plots, such as the ones in Figure 30 allowing for a clear visualization of the sEMG signals over time.

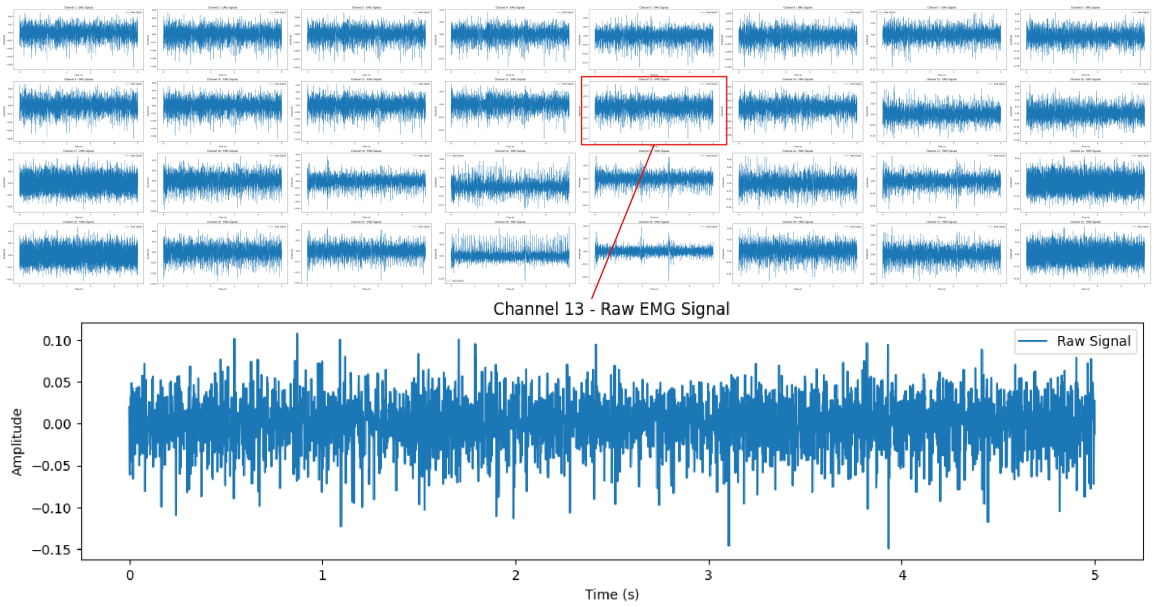


Figure 30: Raw data time series plot

These time series plots provided an initial visualization of the raw sEMG data, offering a clear representation of the signal patterns captured by the electrodes. Figure 30 showcases an example of the raw data visualization from participant 2, session 1, and trial 1, representing Gesture 1 (LP). The figure illustrates the raw EMG signal data across 32 channels, captured during a specific gesture from one trial. The main focus is on Channel 13 (FP), which is highlighted with a detailed, zoomed-in view at the bottom of the figure, showing signal fluctuations over a 5-second period. While these images effectively showcased the raw data, further processing was necessary to enhance the quality of the signals and extract meaningful features for machine learning.

To enhance the raw sEMG signals and extract meaningful features, the BioSPPy library was employed. BioSPPy is a versatile library for biosignal processing that offers various tools for filtering and analyzing physiological signals. The 'biosppy.signals.emg' module was utilized for filtering the signals, detecting muscle activation periods, and generating summary plots of the processed data. This module applies several techniques, such as bandpass filtering to eliminate noise outside of the frequency range typical of human muscle activity and highlighting the essential features of muscle activation.

To validate and visually compare the enhancements made by BioSPPy, the processed signals were overlaid on the raw data in time series plots. This overlaying provided a clear visual representation of the improvements, making it easier to observe the reduction in noise and the extraction of significant features, as illustrated in Figure 31. By directly comparing the raw and processed signals, the effectiveness of the processing steps could be easily verified.

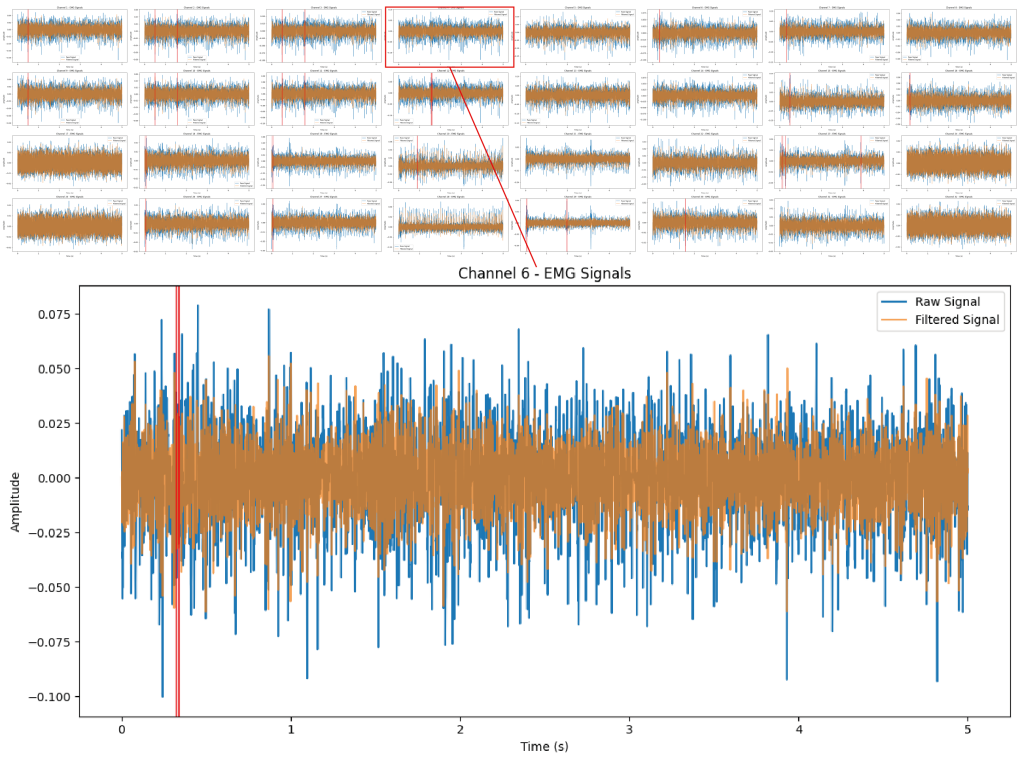


Figure 31: Time Series Plot of BioSPPy-Processed Data Overlaid on Raw EMG Signals

While these time series plots were valuable for initial data exploration and validation, they are not the most effective format for training a CNN. CNNs excel at interpreting spatial hierarchies in images, and time series plots do not fully leverage this capability. To address this limitation, the data was converted into spectrograms, which represent the frequency spectrum of the signals over time, using the same BioSPPy module. Spectrograms provide a richer, two-dimensional representation of the data, making them more suitable for CNN training. This transformation allows the CNN to better capture the temporal and frequency characteristics of the EMG signals, ultimately enhancing the model's ability to accurately recognize gestures.

To prepare the time series EMG data for CNN training, the data was transformed into spectrograms using a custom Python script, using libraries such as librosa and BioSPPy. This transformation is crucial, as CNNs are highly effective at analyzing visual patterns in images, and spectrograms encapsulate both time and frequency information, essential for capturing the complex dynamics of EMG signals.

The script leveraged several specialized libraries to ensure robust data handling and signal processing. wfdb was used for reading waveform databases, providing reliable access to extensive EMG data sets. librosa offered advanced signal processing capabilities, ideal for handling the sophisticated demands of EMG analysis. matplotlib was employed for plotting, which facilitated the visualization of spectrograms and enabled preliminary assessments of signal quality and content. Each EMG signal file was read at its corresponding sampling rate of 2048 Hz, a common choice for capturing nuanced variations in muscle activity. The script processed the data from each of the 32 channels individually, enhancing the signal with a bandpass filter

applied via the BioSPPy library's EMG module. This filter, typically set to isolate the frequency band significant for muscle movement analysis, helped reduce noise and improve the clarity of the EMG signals.

Following the initial processing, a log Mel spectrogram for each filtered signal was computed using librosa. The Mel scale is particularly beneficial for EMG signal processing as it provides a detailed representation of the signal's frequency content over time, making it exceptionally useful for identifying patterns inherent in EMG signals.

The resulting spectrograms were then organized into a 32-subplot grid (8x4), with each subplot representing one channel, as shown in Figure 32. This layout was chosen to ensure a comprehensive overview without clutter, omitting axis labels to enhance visual clarity. Plot settings were adjusted to eliminate spaces between subplots, and the complete grid was saved as a single image file in PNG format. This final image, capturing the temporal and spectral dynamics of the EMG signals across all channels, provides a rich, multidimensional dataset ready for CNN training, optimizing the model's learning phase by presenting the data in a format that maximizes the CNN's pattern recognition capabilities.

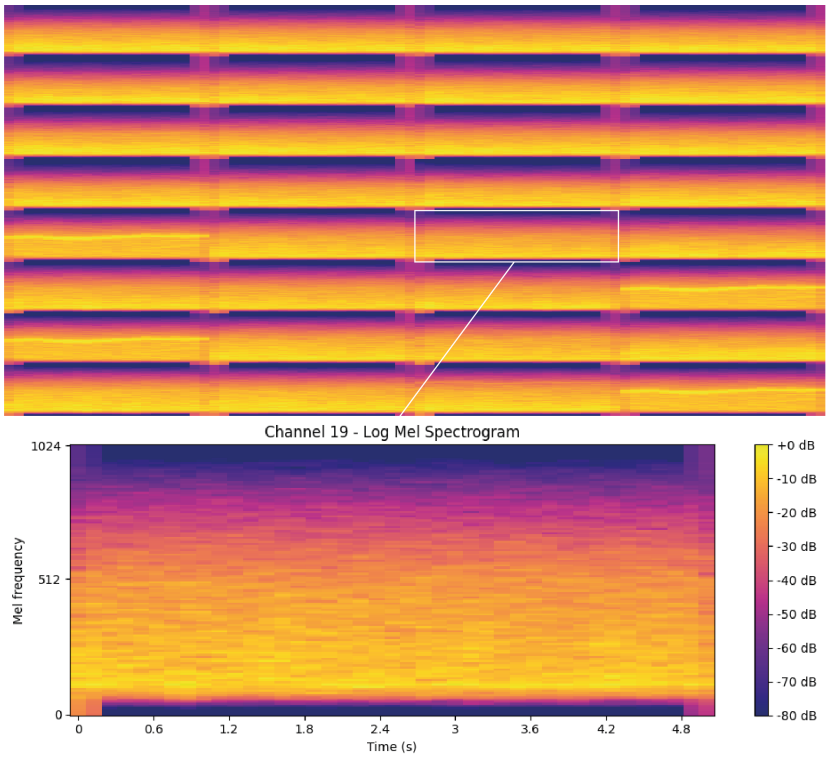


Figure 32: Log Mel spectrogram of BioSPPy-Processed Data.

The image shown in Figure 32 presents a detailed spectrogram representing the EMG signal from participant 2, session 1, and trial 1, representing Gesture 1 (LP), with its focus on Channel 19, selected randomly to illustrate the typical spectral dynamics observed across all channels. This channel, like others in the dataset, provides a visual depiction of how frequency components of muscle activity are distributed over time during a specific gesture. The spectrogram employs a color scale ranging from -70 dB to +10 dB, indicating the intensity of frequency components.

Warmer colors (yellows to reds) highlight stronger signal presence at specific frequencies and times, while cooler colors (blues) indicate weaker signals. This color mapping is crucial for assessing the energy levels within the muscle activity captured in the EMG data. The spectrogram's vertical axis represents frequency, measured in Hertz (Hz), scaling up to 1024 Hz, which showcases the range of frequencies generated by muscle contractions. The horizontal axis tracks time over a 5-second period, corresponding to the total duration of the data captured. Notable features in the spectrogram include areas of concentrated energy, which may correspond to specific phases of the gesture being performed. For instance, an increase in energy in mid-range frequencies at certain times could indicate more vigorous muscle activity. These patterns are particularly valuable for training CNNs, which can learn to recognize these patterns for accurate gesture classification.

4.3 Training

With the EMG signals successfully converted into spectrogram images, the data was now in a format well-suited for training the CNN. This preparation phase ensured that the CNN could effectively learn and recognize the nuanced differences in EMG signals corresponding to various hand and wrist movements. Consequently, the spectrograms served as the primary input for the training process, facilitating the development of a robust model capable of accurate gesture classification.

The AlexNet architecture was chosen for this study due to its proven effectiveness in handling image recognition tasks. Its suitability for interpreting the complex patterns in spectrogram images of EMG signals makes it an ideal choice. AlexNet's deep learning framework is designed to capture hierarchical features from images, as represented in Figure 33, and it is crucial for accurately recognizing gestures.

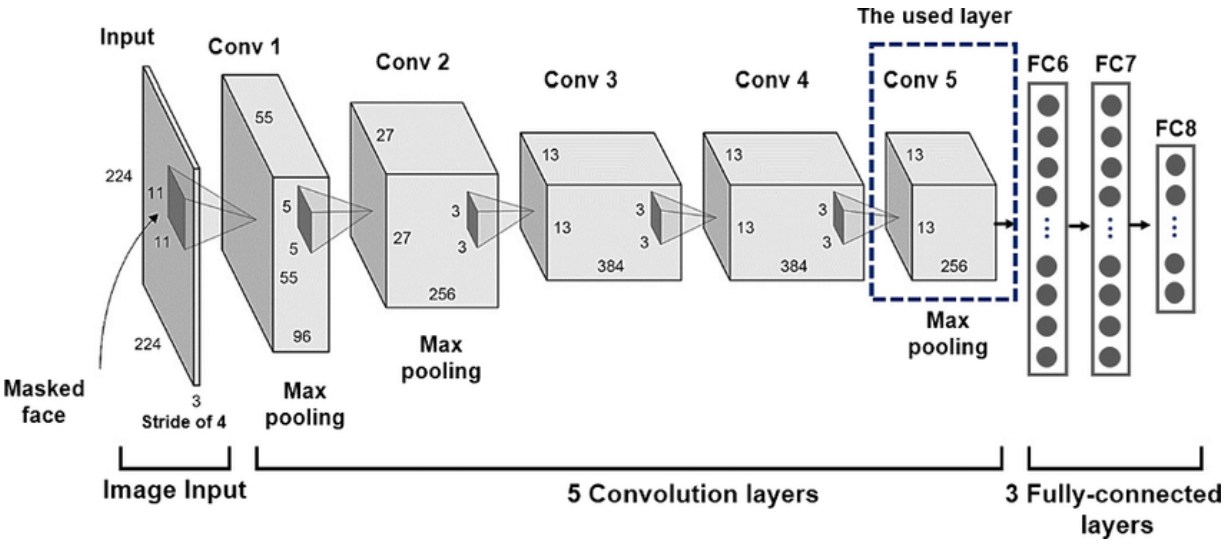


Figure 33: AlexNet arquitecture representation Image from Hariri, 2022

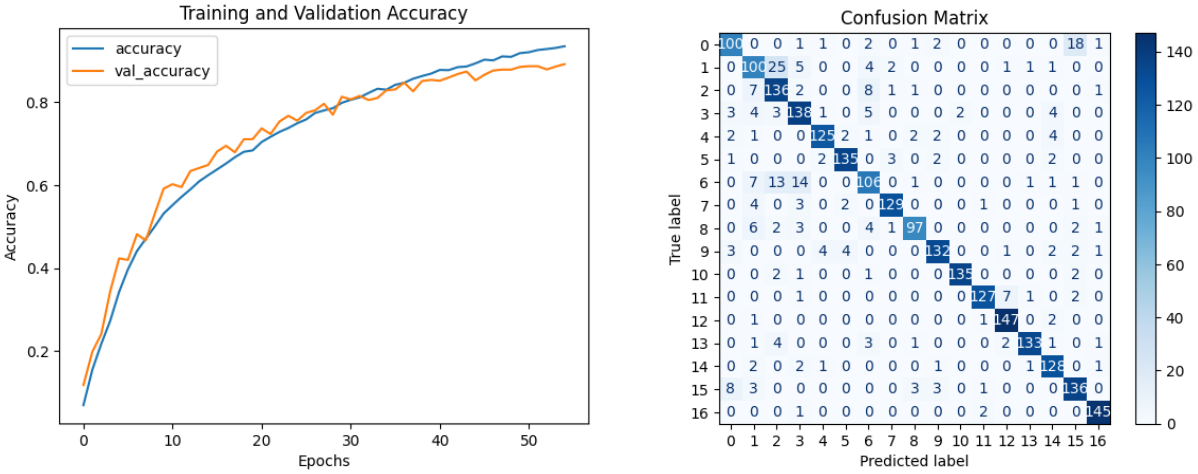
To ensure the model's robustness and to avoid data leakage, the dataset was initially partitioned such that participants were divided into two distinct groups: one for training and another for testing. This approach ensured that the test set contained data from participants whose signals were not seen during the training phase, thereby eliminating any potential bias. After this initial partitioning, the training data was further split into 80% for training the model and 20% for validation. This method allowed for rigorous evaluation of the model's performance on unseen data and ensured that the model could generalize well to new participants.

After selecting the AlexNet architecture for its proven capabilities in image recognition, the next critical step involves tuning the hyperparameters that will be used for the training process of the CNN. These parameters, particularly the batch size and learning rate, play a pivotal role in determining how effectively the network learns from the spectrogram data and how efficiently it can converge to a high-performing model.

The selection of appropriate batch size and learning rate is crucial for the successful training of the CNN. These hyperparameters significantly influence the convergence speed and overall performance of the model, affecting everything from computational load to model accuracy and generalization.

The batch size determines the number of training samples used in one forward and backward pass through the network. A smaller batch size often leads to noisy gradient updates, which can help the model escape local minima and explore the model's loss landscape more comprehensively, albeit possibly slowing down convergence. Conversely, a larger batch size provides more stable gradient updates but requires more memory and can potentially lead to overfitting. Various batch sizes were systematically tested using a structured approach such as grid search to ensure a thorough evaluation, with the model's performance assessed based on several key metrics: accuracy, validation accuracy, confusion matrices, overall results.

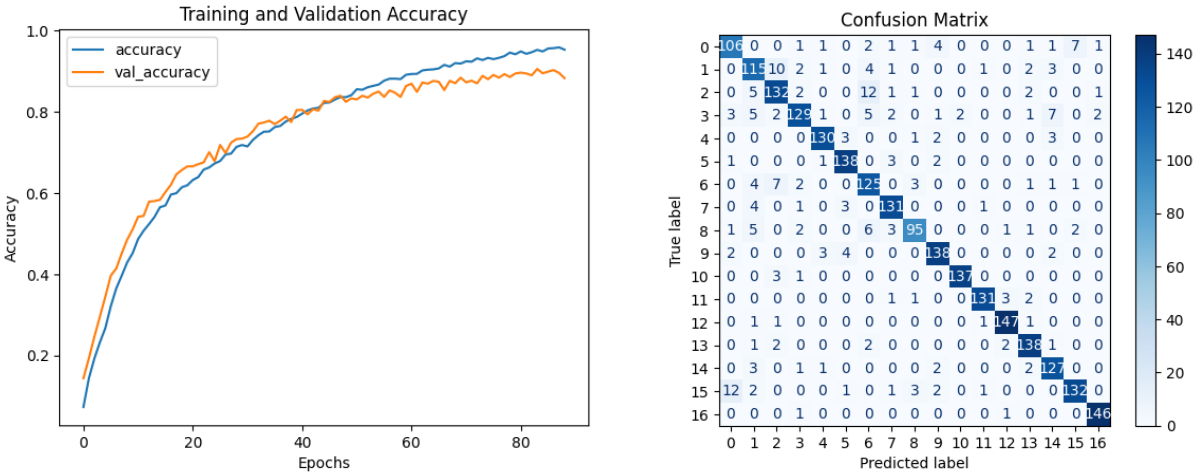
The model plot, trained with a batch size of 16, represented in Figure 34 a), shows a steady and consistent improvement in both training and validation accuracy, indicating stable learning over the epochs. The closely aligned accuracy curves suggest that the model generalizes well to unseen data, contrary to expectations of instability from using a smaller batch size. The confusion matrix, Figure 34 b), corroborates the effective learning, displaying high correct classification rates across most classes. However, there are notable misclassifications, such as between predicted label 2 and true label 1, which could indicate areas where the model might still be improved for better specificity between similar classes. These misclassifications do not necessarily reflect unstable learning but highlight specific challenges in distinguishing between certain gestures.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 34: Accuracy and validation plot (a) and confiusion matrix (b) of model with batch 16.

The model plot, trained with a batch size of 32, represented in Figure 35 a), shows a balance between computational efficiency and model performance, shows excellent learning and generalization. Both training and validation accuracies demonstrate high and closely matched values, suggesting that the model is not overfitting and is generalizing well to new data until epoch 60 approximal, where the accuracy is higher than 80%, using early stop to stop the training can solve this overfitting. The confusion matrix, Figure 35 b), reinforces this observation, with strong diagonal entries indicative of high correct classification rates across nearly all classes. some misclassifications are minimal but present.

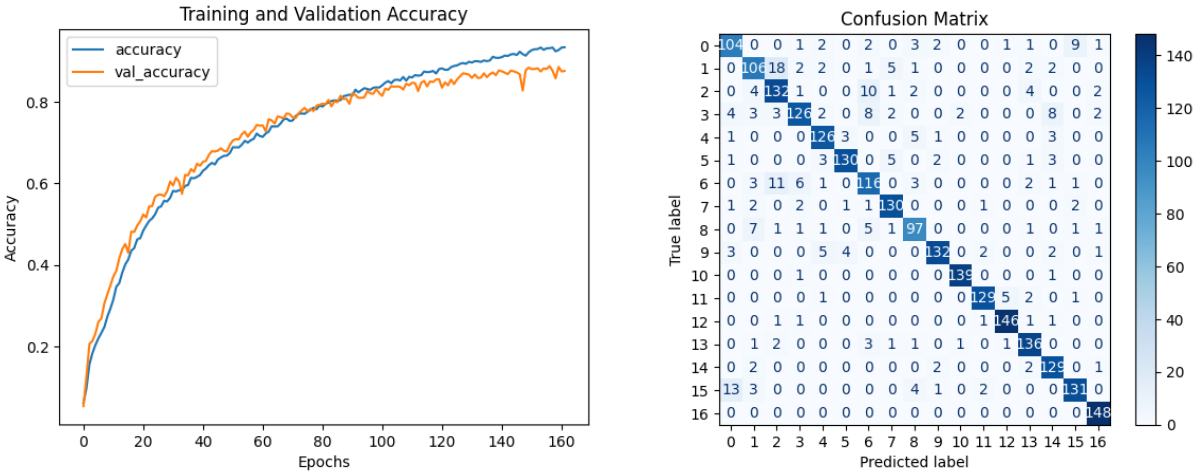


(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 35: Accuracy and validation plot (a) and confiusion matrix (b) of model with batch 32.

The model plot, trained with a batch size of 128 as represented in Figure 36 a), demonstrates robust learning capabilities, achieving high training and validation accuracies over 160 epochs. Despite that it can be seen some divergence between the training and validation curves in the later stages, the overall close alignment indicates effective learning with some signs of overfitting

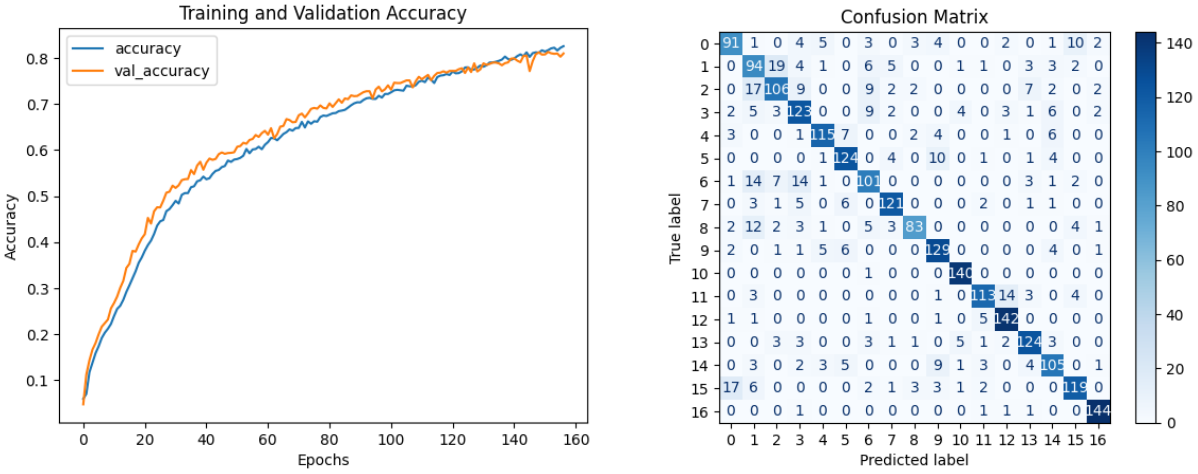
as training progresses. The confusion matrix, Figure 36 b), corroborates the performance seen in the accuracy plot, with strong correct classification rates across most classes. However, the model also presents some specific misclassifications in some classes.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 36: Accuracy and validation plot (a) and confiusion matrix (b) of model with batch 128.

The model plot, trained with a batch size of 254 as represented in Figure 37 a), The model's performance over a training period of 160 epochs demonstrates an efficient learning curve, as illustrated by the closely aligned training and validation accuracy curves in the plot. The high and stable accuracies indicate that the model generalizes well, with almost no overfitting, besides a small spike towards the end of the training period. The confusion matrix, Figure 37 b), supports the effectiveness of the model, showing high correct classification rates across most classes, with some exceptions where misclassifications suggest areas for potential improvement.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 37: Accuracy and validation plot (a) and confiusion matrix (b) of model with batch 256.

In selecting the optimal batch size for training the model, batch size 32 was chosen due to its

superior balance between computational efficiency and high model accuracy, achieving the best overall performance among the tested sizes, as seen in Figure 38. This batch size demonstrated the highest accuracy, peaking at over 90%, indicating an effective learning process without the trade-offs observed in smaller or larger batch configurations. Notably, batch size 32 showed a significant advantage in managing computational resources better than larger batches, while avoiding the instability associated with smaller batches. The training and validation accuracy curves for batch size 32 remained closely aligned throughout the training process, highlighting its capability to generalize well to unseen data. The use of early stopping around 60 epochs, as indicated by the accuracy trends, can further enhance training efficiency by preventing overfitting and conserving computational efforts. Moreover, the confusion matrix corresponding to this batch size revealed strong diagonal entries with minimal misclassifications, confirming the model's robustness in distinguishing between classes effectively. In comparison, larger batches required more computational power and did not proportionally increase accuracy, while smaller batches, though faster to compute, did not achieve the same level of accuracy or stability. Therefore, batch size 32 was identified as the most practical and effective choice for the model, optimizing both performance and operational efficiency.

The learning rate is a critical hyperparameter in the training of machine learning models as it controls the step size used to update the model weights during each iteration. An appropriately set learning rate ensures that the model can converge to the optimal weights efficiently without overshooting, which could lead to divergence, or moving too slowly, which can result in prolonged training times without significant gains in accuracy. Similar to the batch size, various learning rates were tested to identify the one that achieves the best balance between convergence speed and stability. Through systematic testing across a range of learning rates, it was possible to assess their impact based on key performance metrics including model accuracy and loss.

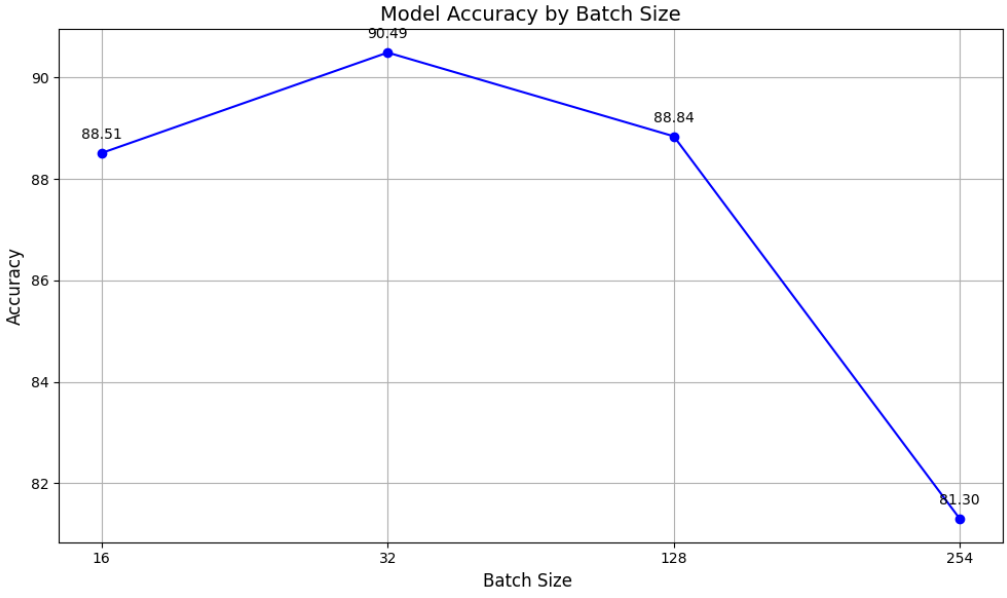
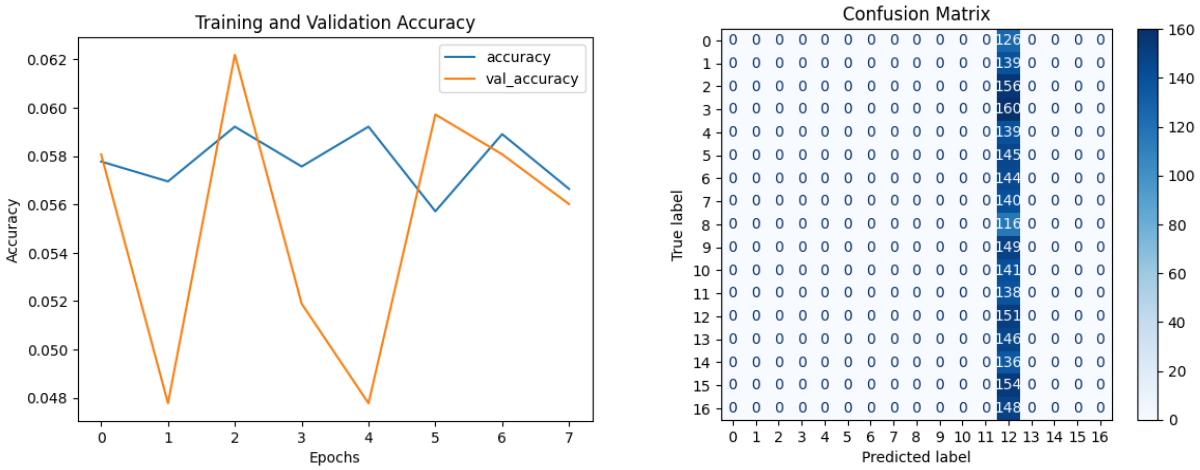


Figure 38: Model accuracy by batch size

The model plot, trained with a learning rate of 0.01, represented in Figure 39 a), shows that the plot haves a wild fluctuation, such drastic swings in accuracy indicate instability in the model's learning process under this learning rate. Both training and validation accuracies are extremely low, hovering around 5% only, which is just above random guessing for 17 gesture classification. This is also supported by the confusion matrix, represented in Figure 39 b), which indicates poor performance across all classes. All true labels are largely misclassified across the predicted labels, as shown by the absence of significant values along the diagonal.

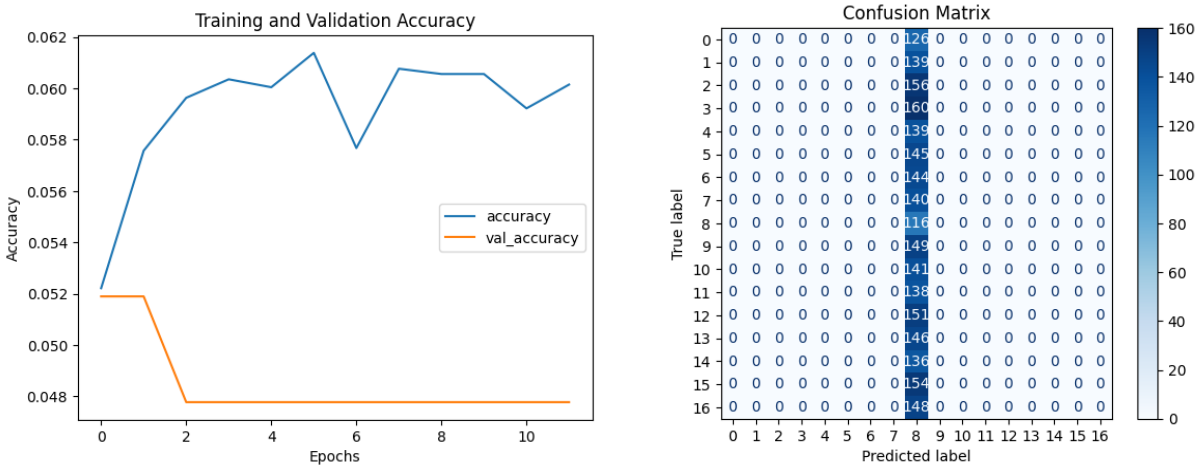


(a) Training and validation accuracy plot

(b) Confusion matrix

Figure 39: Accuracy and validation plot (a) and confiusion matrix (b) of model with learning rate 0.01.

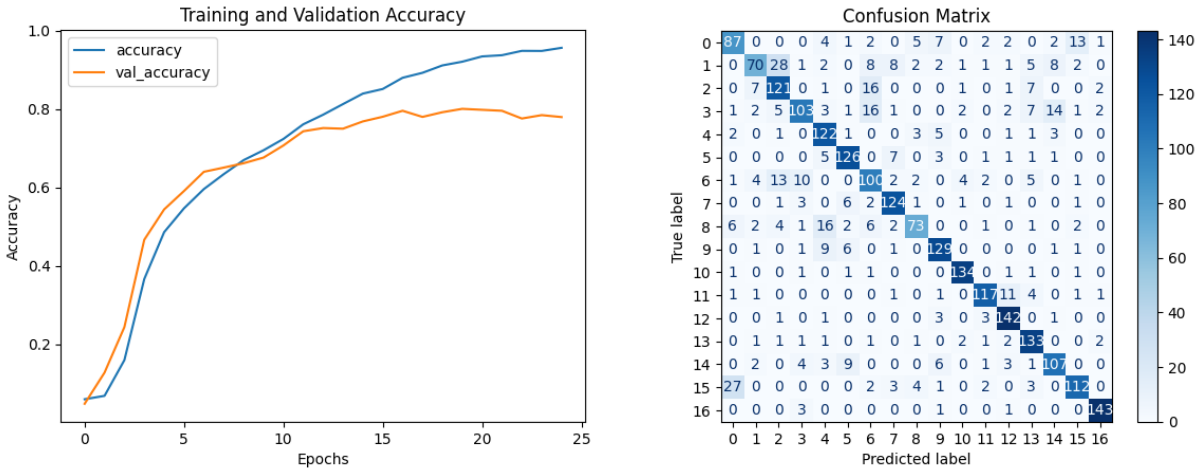
The model plot, trained with a learning rate of 0.001, represented in Figure 40 a), presents a stark disparity between the training and validation accuracies. The training accuracy shows a rapid ascent initially but then plateaus around 6%, the validation accuracy is essentially flat and near zero, suggesting the model has no predictive power on unseen data. The confusion matrix, Figure 40 b), shows similar results the model represented in Figure 40 b), with extremely poor performance, this result suggests that the model has not learned any distinguishing features of any class.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 40: Accuracy and validation plot (a) and confiusion matrix (b) of model with learning rate 0.001.

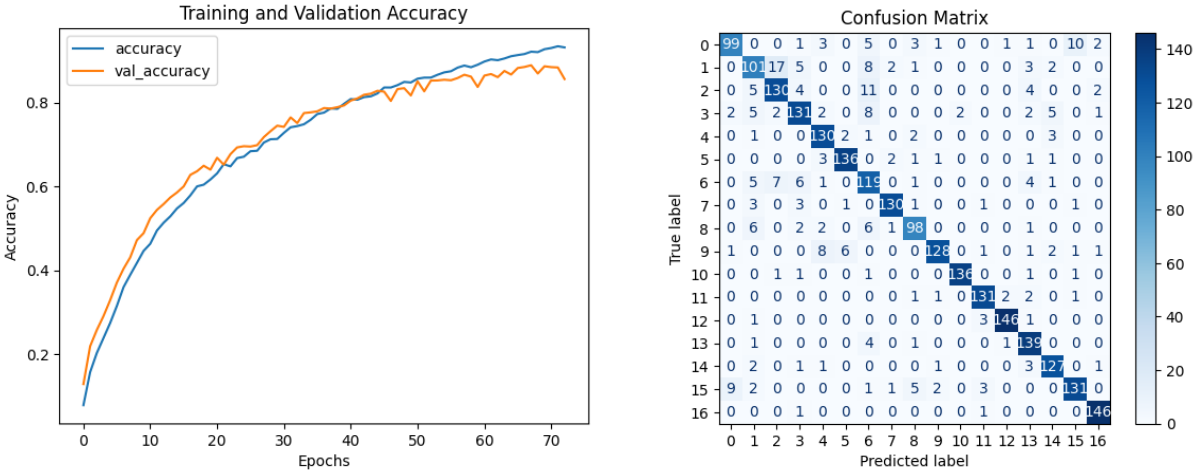
The model plot, trained with a learning rate of 0.0001, represented in Figure 41 a), shows some stability over the first 10 epochs with, Training accuracy eventually plateaus around 90%, while validation accuracy stabilizes around 80%. This suggests some overfitting as training accuracy exceeds validation accuracy by a noticeable margin. The confusion matrix represented in Figure 41 b), demonstrates significant diagonal entries, with some classes showing relatively high correct classification rates.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 41: Accuracy and validation plot (a) and confiusion matrix (b) of model with learning rate 0.0001.

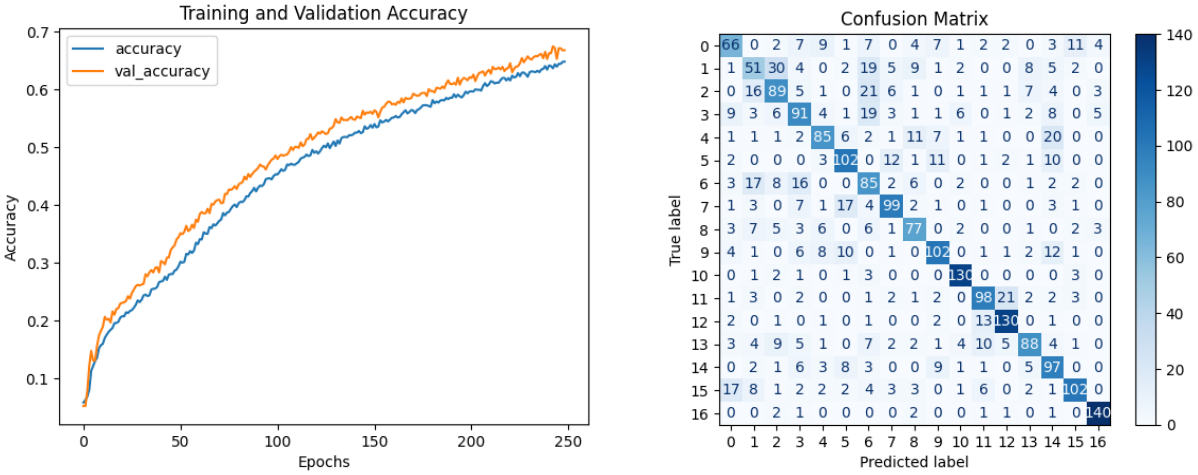
The model plot, trained with a learning rate of 0.00001, represented in Figure 42 a), reveals a steady increase in both training and validation accuracy, with training accuracy peaking slightly above validation, indicating a good fit. The convergence of both curves suggests that the model is well-tuned, though the small gap that develops and persists might suggest slight overfitting. The confusion matrix, Figure 42 b), shows an overall good performance, with some classes achieving very high correct classification, with some misclassifications.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 42: Accuracy and validation plot (a) and confiusion matrix (b) of model with learning rate 0.00001.

The model plot, trained with a learning rate of 0.000001, represented in Figure 43 a), shows a gradual and consistent increase in both training and validation accuracy over a large number of epochs (250), suggesting a slow but steady learning rate. The plot indicates that the model is learning progressively without significant jumps or falls in accuracy, which is a good sign of stability. In the other hand the confusion matrix, represented in Figure 43 b) shows moderate performance, with several classes displaying reasonable correct classification rates. The matrix also illustrates some degree of confusion between classes.



(a) Training and validation accuracy plot **(b)** Confusion matrix

Figure 43: Accuracy and validation plot (a) and confiusion matrix (b) of model with learning rate 0.000001.

In the process of training the model, selecting the optimal learning rate was crucial due to its impact on the convergence speed and stability of the model's learning curve. As illustrated in Figure 44, a comprehensive range of learning rates was tested, with the goal of finding a balance that maximizes model accuracy while minimizing the risk of overfitting and prolonged training times. The learning rate of 0.00001 emerged as the most effective, achieving the highest

peak accuracy of approximately 88.88%. Compared to other models, this setting presents an optimal number of training epochs, making it computationally efficient and demonstrating a good classification rate, as seen in Figure 42 b). The performance of the model with this learning rate, as depicted in Figure 42 a), reveals a steady increase in both training and validation accuracy, with the training accuracy peaking slightly above validation, which suggests a good fit. The convergence of both curves illustrates that the model is well-tuned, though the small gap that develops and persists might suggest slight overfitting. However, this level of overfitting is manageable and does not detract significantly from the model's performance. The confusion matrix, shown in Figure 42 b), displays overall strong performance, with several classes achieving very high correct classification rates, although there are some misclassifications. Furthermore, this learning rate facilitates a balanced training process that is not only effective but also efficient, optimizing resource usage while maintaining high accuracy. This efficiency is vital for practical applications where both performance and computational cost are considered. Therefore, a learning rate of 0.00001 was identified as the most practical and effective choice for this model, optimizing both performance and operational efficiency, making it suitable for extended training and real-world application.

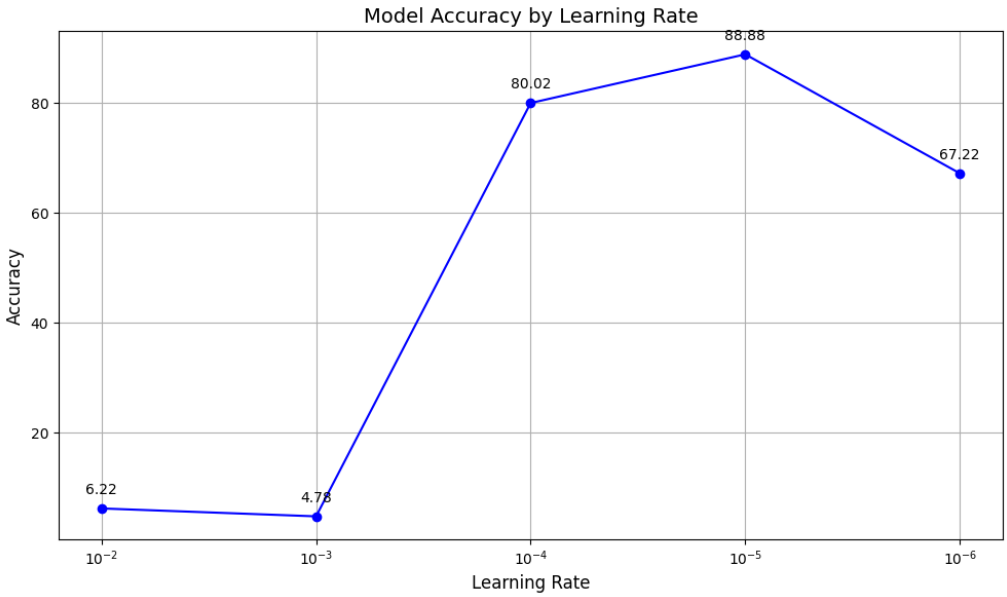


Figure 44: Model accuracy by learning rate plot

With the optimal batch size of 32 and a learning rate of 0.00001 identified, the next phase involved training multiple iterations of the model to establish a robust baseline for general accuracy and to identify any significant deviations in performance. This step was crucial to ensure that the selected hyperparameters consistently supported effective learning across various iterations of the model, providing insights into its stability and reliability. Each model iteration was trained using the same partitioned dataset and employing the AlexNet architecture, allowing for a controlled comparison of outcomes. The systematic training of seven different models under uniform conditions allowed for a detailed evaluation of the CNN's robustness across diverse

sEMG data scenarios. Performance metrics such as accuracy, precision, recall, and F1-score were recorded, revealing variations in these metrics across the models, as represented in Table 5. For instance, accuracies ranged from 0.8468 to 0.9049, with an average of 0.8876, indicating consistent performance with minor deviations of 0.0184. Specifically, the standard deviation in model accuracies was calculated to assess the spread of results, which provides a statistical measure of the variability or consistency across different training cycles. This standard deviation, although small, highlights the small but notable fluctuations that could arise from factors such as model initialization and data shuffling between epochs. By quantifying these deviations, we can better understand the reliability of the model under different conditions and affirm the effectiveness of the chosen hyperparameters. Such findings underscore the generalizability of the model and the predictability of its performance in real-world settings, thereby affirming the stability of the learning process and providing a reliable gauge of the model's performance across different datasets.

Table 5: Multiple training iterations results

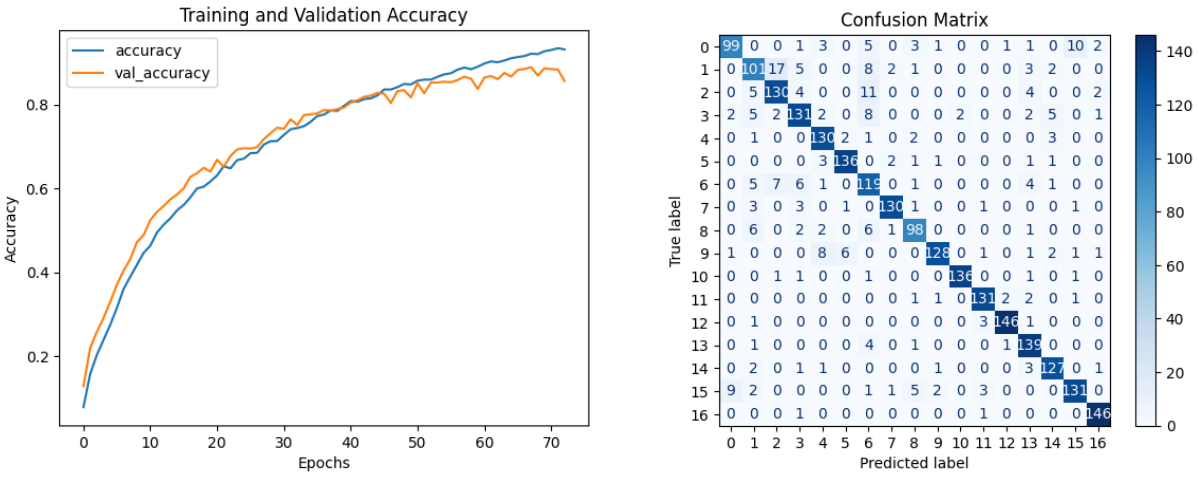
	Accuracy	Precision	Recall	F1 Score	Predictions
Model 1	0.8468	0.8487	0.8452	0.8452	2060
Model 2	0.9007	0.9007	0.9006	0.9000	2146
Model 3	0.8942	0.8957	0.8929	0.8931	2135
Model 4	0.8888	0.8904	0.8882	0.8884	2124
Model 5	0.9049	0.9054	0.9044	0.9043	2113
Model 6	0.8987	0.9003	0.8983	0.8983	2160
Model 7	0.8793	0.8797	0.8790	0.8787	2121
Average	0.8876	0.8887	0.8869	0.8868	2122
Deviation	0.0184	0.0181	0.0188	0.0187	29

From the Table 5, Model 4, which exhibited an accuracy of 0.8888, was selected for further use and analysis. This model not only represents the median performance level but also aligns closely with the average accuracy observed across all iterations, which was 0.8876. This decision was strategically made to minimize any deviations in performance, ensuring that the chosen model was both accurate and representative of typical results. By selecting a model that mirrors the general accuracy trend, it can be asserted that its performance is reliable and

reproducible, which is crucial for real-world.

Through Figure 45 a), the plot demonstrates a consistent improvement in both training and validation accuracy over the epochs, which indicates effective learning and good generalization capabilities. The training accuracy steadily increases and begins to plateau around the 56-epoch mark. Meanwhile, the validation accuracy follows a similar trajectory but starts to plateau slightly earlier, which is typical as the model begins to fit closely to the training data while still maintaining good performance on unseen validation data. The close convergence of the training and validation accuracy lines with a minor divergence towards the end suggests that the model is well-calibrated with minimal overfitting. The slight gap indicates some degree of overfitting but it's not significant, since the early stop method was employed, which is positive as it implies that the model is robust and generalizes well to new data.

The confusion matrix shown in Figure 45 b) a strong diagonal, indicating that most classes are predicted correctly with high accuracy. The strong performance in the diagonal of the matrix confirms that the model achieves high accuracy across most classes.



(a) Training and validation accuracy plot

(b) Confusion matrix

Figure 45: Accuracy and validation plot (a) and confiusion matrix (b) of model 4 from Table 5.

Figure 46 presents the number of correct predictions for each of 17 gestures, with each gesture represented by 189 instances, totaling 3213 gestures. This provides a detailed view of how well the model performs on a per-gesture basis. The model exhibits significant variability in performance across different gestures. Some gestures, such as Gestures 11 (WE), 12 (FP), 14 (HO), and 17 (rest), show notably high correct predictions, suggesting that the model is particularly effective at recognizing these gestures. On the other hand, gestures like Gesture 3 (TLFO), and Gesture 4 (TIFO) are recognized with less accuracy, indicating potential challenges in distinguishing features or characteristics specific to these gestures. Recognizing the strengths and weaknesses on a per-gesture basis helps in setting realistic expectations for the model's application in real-world scenarios. The model's ability to perform exceptionally well for certain gestures while struggling with others totaling 2124 correct prediction out of 3213 total

images. Additionally, the model demonstrated a high efficiency in processing, with an average classification time of 43 ms per image after predicting 3213 images, underscoring its suitability for real-time applications.

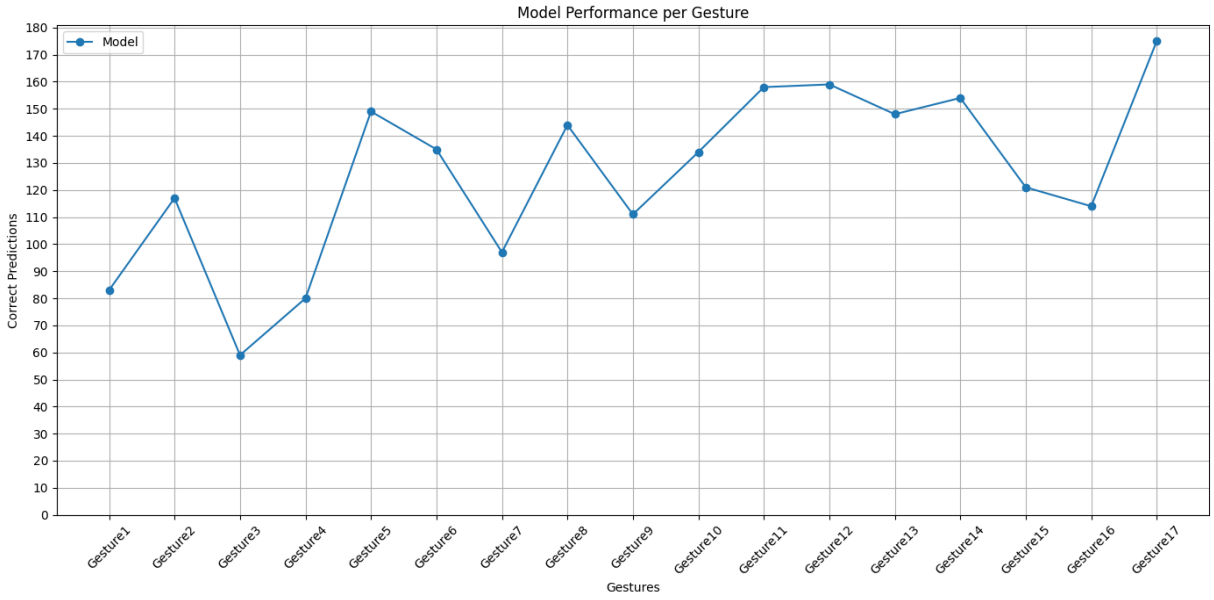


Figure 46: Model performance per gesture for the chosen model

4.4 Ablation

The ablation study is designed to systematically assess the influence of features on model performance by selectively disabling filters within the convolutional layers. This approach enables to pinpoint critical features and components that significantly contribute to the model's ability to recognize and classify EMG-based gestures effectively. Understanding these components is essential, as it allows optimization of the network by focusing on the most impactful elements, thereby enhancing efficiency and potentially improving gesture recognition accuracy. To facilitate a deeper understanding of how the model predicts various gestures, the first step involved generating and analyzing the feature maps from the model. Feature maps in the context of CNNs are the outputs of the convolutional layers within the network. Each feature map is essentially a transformed representation of the input data, processed through convolutional filters. These maps highlight different aspects or features of the input data, depending on the specific filters applied in each layer. These feature maps reveal the intermediate outputs of the convolutional layers, showcasing which features and patterns are emphasized at each stage of the network. This visualization provides valuable insights into the model's operational mechanics, highlighting the aspects of the input data that are most influential for accurate gesture classification. By visualizing these feature maps, it can be seen what each layer of the network is focusing on. This is not only crucial for understanding how the network processes data but also for diagnosing whether the network is paying attention to the relevant features for accurate gesture classification.

In the initial layers of a CNN, the filters capture basic visual features such as edges, colors, and simple textures. These maps often appear less pixelated and more reminiscent of the original input because they are closer to the actual visual inputs, as seen in Figure 47. The feature maps in these early stages might seem blurrier depending on the simplicity of the features they're picking up, which include basic details without complex transformations.

As it progresses deeper into the network, the convolutional layers start combining these basic features to detect more complex patterns. These might include combinations of edges to form shapes or textures that are specific to certain gestures, as illustrated in Figures 48 and 49. The maps may start to look more abstract because the features being represented are less directly visual and more conceptual.

In the deepest layers, the feature maps become highly pixelated and abstract. This pixelation occurs because the layers now represent high-level concepts far removed from the visual input. Instead of showing direct visual data, these layers encode information necessary for making classification decisions, as demonstrated in Figures 50 and 51. The increasing pixelation and variation in activation, with some filters capturing more activity than others, indicate that different filters have specialized to recognize very specific features crucial for the final classification tasks. Some filters might be less active because the features they are designed to detect are not as prevalent or critical in the specific inputs being analyzed.

Once the feature maps of the convolutional layers were visualized, the next step involved segmenting the convolved filters into distinct clusters. This division aimed to enhance the understanding of which clusters more significantly impact gesture recognition. Utilizing the outcomes of the feature maps, each filter from the convolutional layers was categorized into one of two clusters. To achieve a detailed understanding of the filter behaviors, Principal Component Analysis (PCA) followed by K-means Clustering was employed.

For each convolutional layer, ranging from conv2d (first convolutional layer) to conv2d_4 (fifth convolutional layer), the filters weights were first extracted and reshaped, ensuring they were amenable for analysis. To facilitate a meaningful comparison and normalization of these weights, each set was normalized by its Euclidean norm. This normalization is crucial as it prevents any single weight from disproportionately influencing the results due to its scale.

Subsequently, PCA was applied to these normalized weights, reducing the high-dimensional weight data to two principal components. This reduction into a two-dimensional space was essential for visualizing and interpreting the complex relationships between the filters' weights in a form that is comprehensible yet retains the most critical variance within the data.

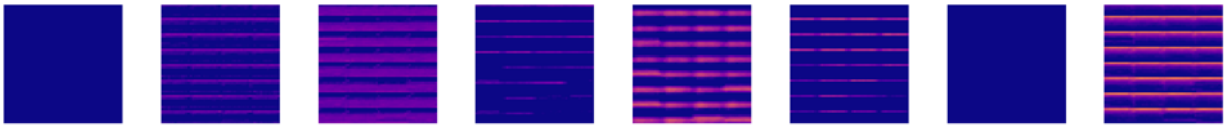


Figure 47: feature maps from the first convolutional layer.

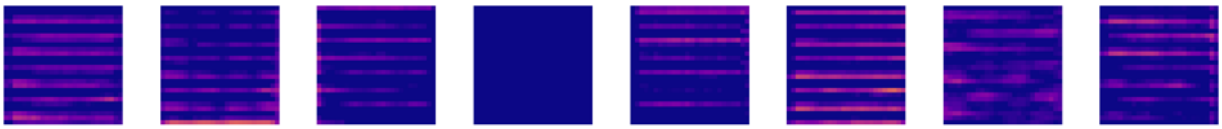


Figure 48: First 8 feature maps from the second convolutional layer.

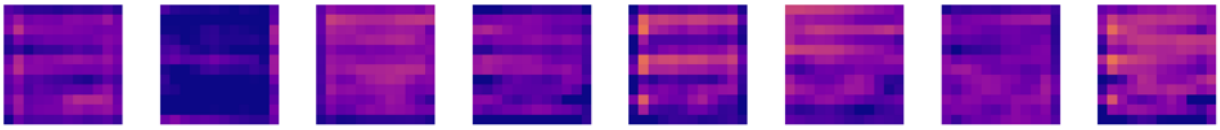


Figure 49: First 8 feature maps from the third convolutional layer.

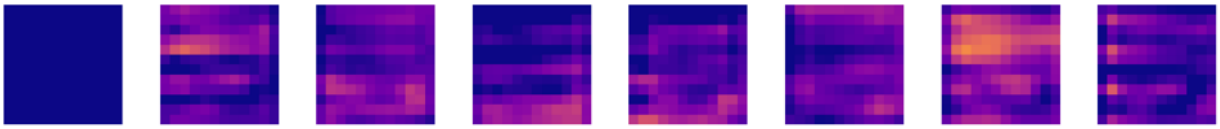


Figure 50: First 8 feature maps from the fourth convolutional layer.

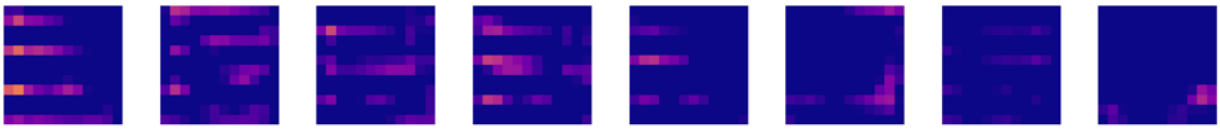


Figure 51: First 8 feature maps from the fifth convolutional layer.

Using K-means Clustering, the filters were systematically grouped into two distinct Clusters. This segmentation was first initialized using the centroids derived from the conv2d layer and subsequently applied to other layers to maintain consistency in Clustering criteria across the network. The Clusters were visualized in a scatter plot, providing a clear depiction of how filters within each layer relate to each other, thereby enhancing our understanding of their contributions to the model's gesture recognition capabilities.

The resultant Clusters are visualized in the scatter plot shown in Figure 52 to 56. These plots display the filters as points in a two-dimensional space defined by the first two principal components (PC1 and PC2). Principal components are new variables that are constructed as linear combinations of the initial variables, in this case, the weights of the filters. These components are designed to capture the maximum amount of variance in the data set with a reduced number of dimensions.

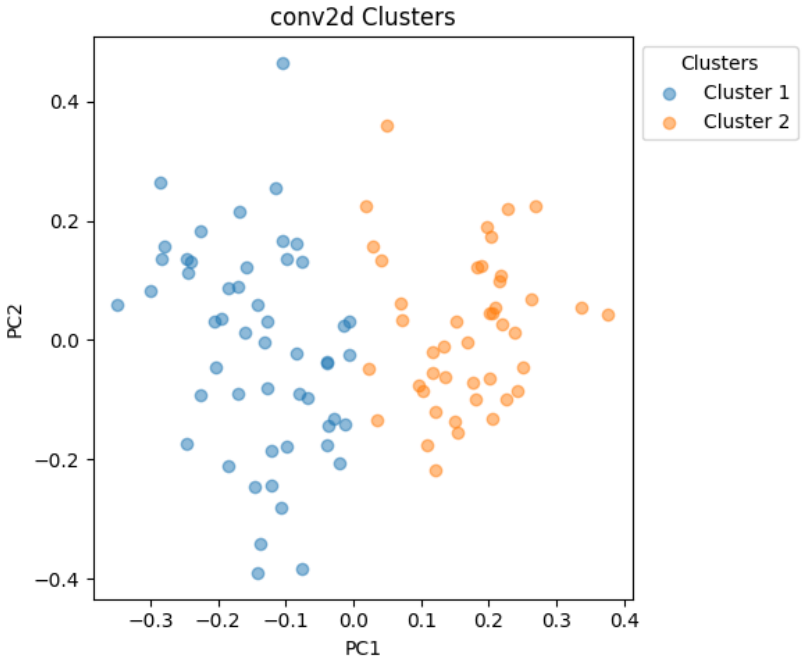


Figure 52: Filter Clusters PCA Scatter Plot for First Convolutional Layer (conv2d).

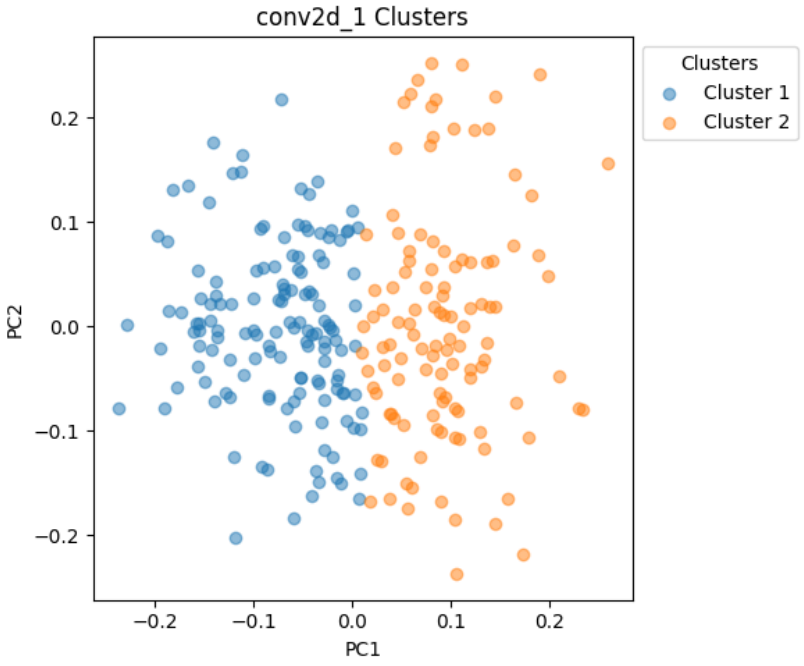


Figure 53: Filter Clusters PCA Scatter Plot for Second Convolutional Layer (conv2d_1).

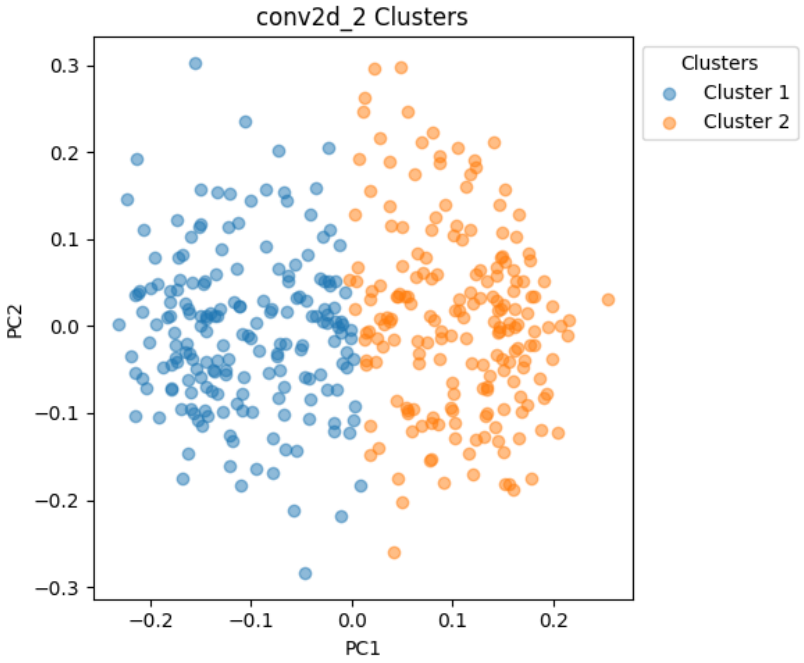


Figure 54: Filter Clusters PCA Scatter Plot for Third Convolutional Layer (conv2d_2).

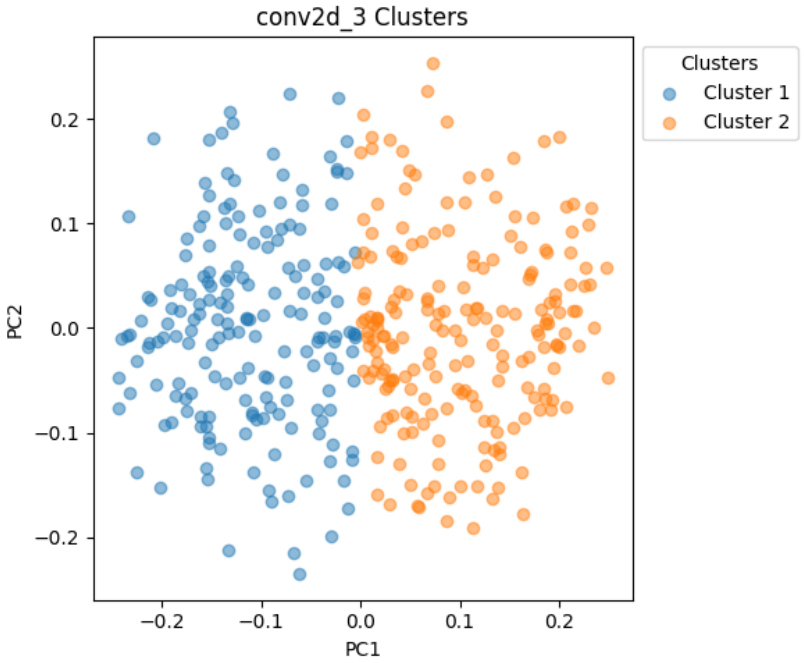


Figure 55: Filter Clusters PCA Scatter Plot for Fourth Convolutional Layer (conv2d_3).

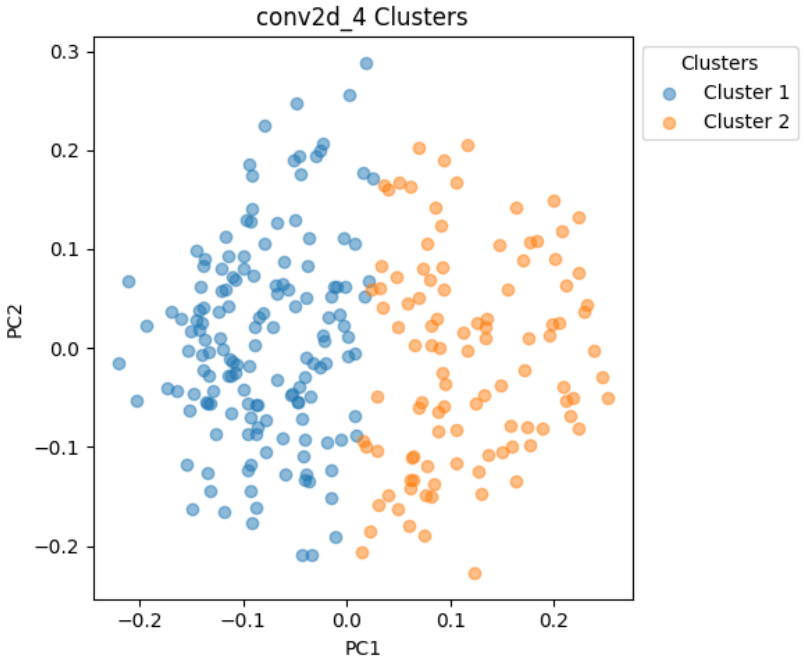


Figure 56: Filter Clusters PCA Scatter Plot for Fifth Convolutional Layer (conv2d_4).

For the ablation study, filters within each Cluster were methodically targeted for deactivation to assess their impact on the model's ability to recognize gestures from EMG spectrograms. This was accomplished by setting the weights and biases of randomly selected filters to zero at increasing proportions of 1%, 5%, 10%, 25%, 50%, and 75%, relative to the total number of filters in each Cluster. The decision to deactivate filters at varying proportions was driven by the need to understand the threshold at which filter deactivation begins to significantly impact model performance. By exploring a wide range of ablation percentages, the study aims to uncover the critical point where the network's performance deteriorates, revealing the redundancy and fault tolerance built into the model. This methodology also helps in identifying which filters are essential for maintaining high accuracy in gesture recognition, thereby pinpointing the most valuable features learned by the network. The intention was to evaluate the influence of each Cluster and layer on gesture recognition systematically.

The study was executed in two primary ways. A general ablation where filters across all convolutional layers were deactivated simultaneously, this approach aims to evaluate the network's overall robustness and its redundancy capabilities. The second way was to make a layer-specific ablation, where the filters were deactivated within a single convolutional layer at a time, while other layers remained operational. This strategy facilitated a detailed analysis of each layer's individual contribution to the network's gesture recognition capabilities and helped isolate the distinct roles played by each layer.

To implement the ablation, detailed logging and tracking mechanisms were established. For each scenario, metadata including the percentage of deactivation, the specific layers affected, and the unique identifiers for the Clusters involved were meticulously recorded. This metadata not only facilitated precise replication of the experiment but also ensured that each step of the

process could be audited and reviewed for consistency and accuracy.

Special attention was given to the proportionate scaling of filter deactivation across layers of varying sizes. Recognizing that each convolutional layer contributes differently to the network based on its dimensionality and filter count, adjustments were made to ensure that the impact of deactivation is proportionally and appropriately evaluated across all layers. This approach ensures that larger layers do not disproportionately influence the overall findings of the study.

The ablation began by loading a pre-trained AlexNet model. For each ablation scenario, a dictionary was created to detail the indices of the filters designated for ablation in each convolutional layer, organized by: percentage-layer-Cluster. An iterative ablation process was employed, where each ablation was applied to the reloaded base model to preserve the integrity of the experiment. Each modified model was then saved in a designated folder, allowing systematic cataloging for subsequent analysis.

The test dataset, consisting of 3213 images labeled with specific gestures, was used to evaluate the performance of each ablated model. A script automatically classified the gestures in each image, recording the outcomes as either correct or incorrect predictions. Results were compiled into a comprehensive table for detailed examination. Post-ablation, the performance of each model variant was analyzed using a suite of statistical tools and performance metrics. The primary metrics included accuracy, precision, recall, and the F1 score, which collectively provide a comprehensive view of the model's efficacy under various states of impairment.

Upon concluding the detailed ablation studies, a critical next step involves the recovery of one of the ablated models. This phase of the research is essential for understanding the resilience and adaptability of neural networks. By attempting to restore the model's original performance levels, it was explored a recovery strategy which retraining the network with the remaining active filters and applying nuanced adjustments to the model's learning parameters to regenerate lost features effectively. The recovery process utilized detailed logging and meticulous tracking of changes to ensure that each intervention could be precisely evaluated. The model, originally impaired by setting a substantial portion of its filters to zero, was reloaded and retrained. This method involved using a low learning rate of 0.00001 with the Adam optimizer, which facilitated fine-grained adjustments to the model's weights, thus restoring its functionality without causing drastic changes that might lead to overfitting. Early stopping was implemented to monitor validation loss and prevent overtraining, ensuring the model's generalizability and robustness.

Recovering an ablated model provides key insights into the network's built-in redundancy and its role in enhancing fault tolerance. Observing how well a model recovers, or fails, can indicate which aspects of the network's design contribute to its robustness. Successful recovery suggests that the network has inherent capabilities to compensate for the loss of certain filters, or that the ablated features, while important, can be regenerated through targeted training strategies. Conversely, limited or unsuccessful recovery efforts highlight the critical nature of the ablated components, underscoring their essential role in maintaining overall network functionality.

This recovery effort is not merely about regaining lost performance but also about deepening

the understanding of which features, and layers are most critical to the network's operation. By selecting a model based on the extent of performance degradation and its potential for recovery, and then rigorously testing the effectiveness of the recovery strategies using the test dataset composed by 3213 images, ensures that the functional capabilities of the model are restored to an acceptable standard. Through quantitative measures of recovery using accuracy, precision, recall, and F1 score, and qualitative assessments of how the model's performance on specific gestures has evolved post-recovery, we gain a comprehensive view of the efficacy of the recovery efforts

Chapter 5

Results and Discussion

After establishing a detailed methodology for the ablation study, this chapter shifts focus to the empirical evaluation, which plays a pivotal role in validating our theoretical predictions about the resilience and adaptability of the neural network in gesture recognition. The findings from this extensive analysis not only reinforce our hypotheses but also uncover critical dependencies within the network's architecture. Supported by a suite of statistical tools, our evaluation offers clear insights into how the network behaves under various stress conditions, providing a robust assessment of its performance and recovery capabilities. Following detailed ablation studies, the recovery phase aimed to explore the network's ability to restore original performance levels. This essential part of the research underscores the network's built-in redundancy and its capability to compensate for lost functionalities through targeted training strategies, highlighting both the resilience and the adaptability of the neural network

5.1 General Ablation Results

This section examines the effects of general ablation on the neural network's performance across two distinct Clusters. By disabling filters at varying levels across all convolutional layers simultaneously, we assess the resilience of each Cluster under uniform stress conditions, providing insight into the network's overall robustness and redundancy capabilities.

General ablation was conducted at varying levels of severity, from as low as 1% to as high as 75% filter deactivation, to ascertain the threshold at which the network's performance significantly declines.

The results from Table 6 indicate that at a low ablation percentage (1%), Cluster 1 demonstrates relative stability in performance, with a slight decrease in accuracy from 88.88% to 83.81%. This suggests that the model retains most of its predictive capability even when a small percentage of filters are deactivated. Notably, a 1% ablation falls within the model's typical deviation (1.84 for accuracy and 29 gestures for prediction), indicating minimal impact at this level. The results from Table 7 reveal that Cluster 2 exhibits a minor decline in accuracy with a

small increase in loss, similar to Cluster 1.

Both Clusters demonstrated remarkable stability in gesture recognition, with correct predictions remaining close to the baseline model's performance, as detailed in Figure 57. This initial resilience suggests that the network contains inherent redundancies that enable it to sustain high performance even when a small proportion of its computational resources are compromised.

Table 6: Total correct predictions out of the test dataset, accuracy and loss from general ablation cluster 1

Percentage	Predictions	Accuracy	Loss
0%	2138	88.88%	0.3478
1%	2112	88.55%	0.3609
5%	2009	83.81%	0.4903
10%	1830	77.10%	0.6879
25%	1601	61.12%	1.1401
50%	1379	46.99%	1.7968
75%	579	18.95%	2.6079

Table 7: Total correct predictions out of the test dataset, accuracy and loss from general ablation cluster 2

Percentage	Predictions	Accuracy	Loss
0%	2138	88.88%	0.3478
1%	2132	88.26%	0.3548
5%	2116	84.47%	0.4557
10%	2095	83.48%	0.4813
25%	1131	40.94%	2.0640
50%	189	6.10%	5.7445
75%	189	6.10%	3.5965

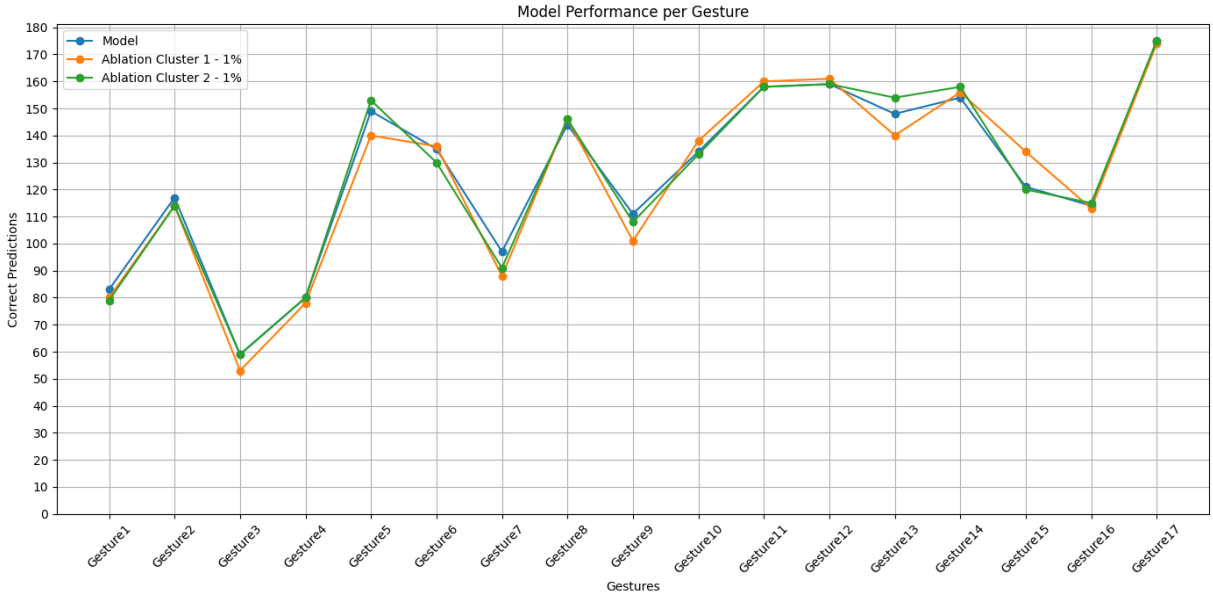


Figure 57: Model Performance per gesture, comparison between original model, and the general ablation of 1% for both Clusters.

However, as the ablation increases to 5%, subtle yet significant declines in accuracy become apparent, particularly within Cluster 1 for gestures such as 3 (TLFO), 5 (TLFE), 7 (IMFE), 9 (IFE), 13 (FP), and 16 (HC), as shown in Figure 58. This targeted decline suggests that these gestures depend more heavily on the specialized features processed by Cluster 1. While the overall accuracy at this ablation level remains relatively similar, 83.81% for Cluster 1 and 84.47% for Cluster 2, the difference in correct predictions is noticeable, with 2009 for Cluster 1 and 2116 for Cluster 2, as detailed in Tables 6 and 7. Despite these variations, the minimal overall impact on network performance underscores the robustness inherent in the system.

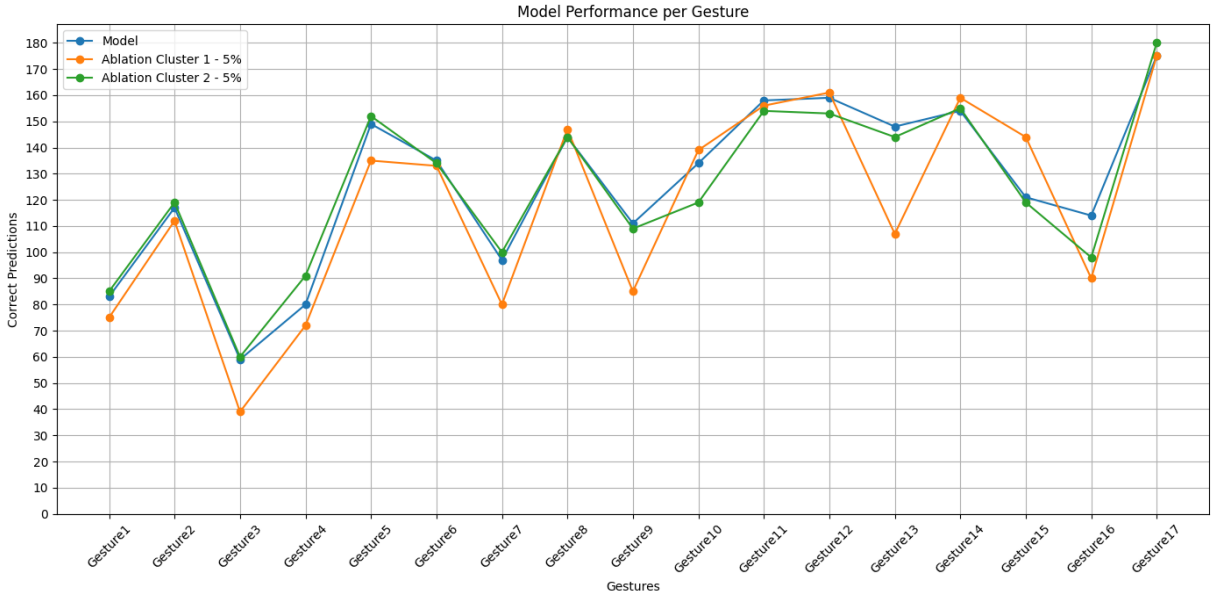


Figure 58: Model Performance per gesture, comparison between original model, and the general ablation of 5% for both Clusters.

As the percentage of deactivated filters increases to 10%, a more pronounced decline in accuracy is observable in Cluster 1, where accuracy sharply drops to 77.10%, coupled with a reduction to 1830 correct gesture predictions, as detailed in Table 6. This decline is more pronounced compared to Cluster 2, which maintains a relatively stable accuracy of 83.48% and 2095 correct predictions, as shown in Table 7. Notably, Figure 59 highlights that gestures such as 3 (TLFO), 5 (TLFE), 7 (IMFE), 9 (IFE), 13 (FP), and 16 (HC), previously identified as significantly impacted in Cluster 1 at the 5% ablation level, continue to be adversely affected at the 10% level. Additionally, new gestures including 1 (LP), 2 (TA), 4 (TIFO), and 8 (LFE) also experience notable declines in Cluster 1. Conversely, in Cluster 2, only a few gestures such as 2 (TA) and 15 (HO) exhibit a significant reduction in prediction accuracy.

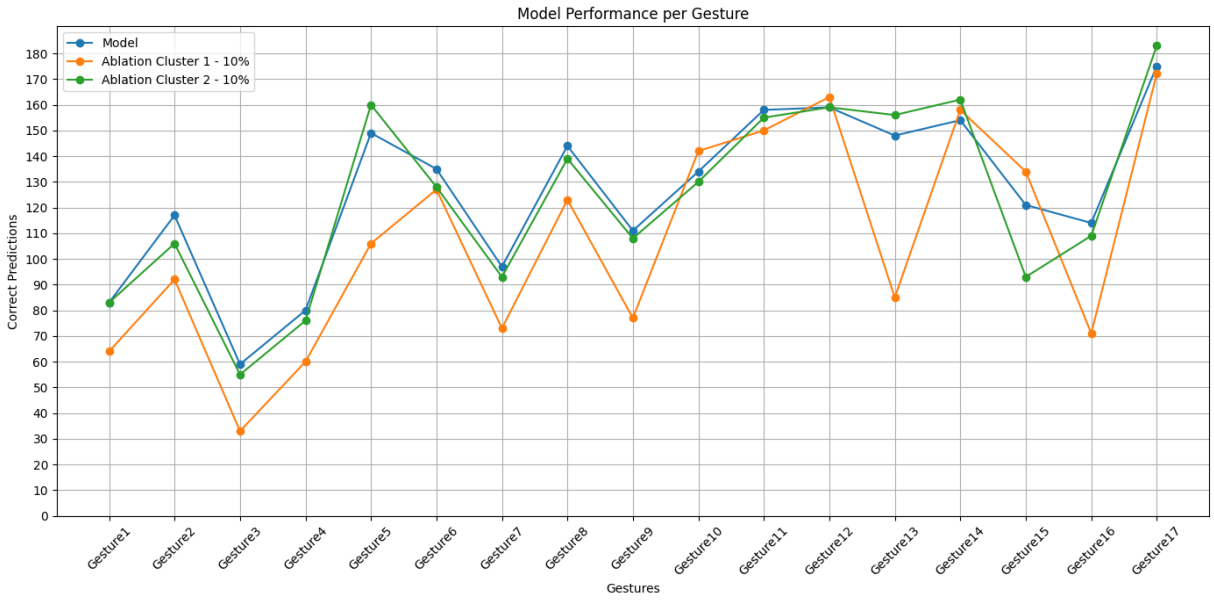


Figure 59: Model Performance per gesture, comparison between original model, and the general ablation of 10% for both Clusters.

At the 25% ablation level, Cluster 2 experiences a significant decline, dropping from 83.48accuracy at 10% ablation to 46.99% as detailed in Table 7. Figure 60 illustrates that at this ablation level, Cluster 2 approaches a critical threshold, with most gestures exhibiting substantial reductions in correct predictions. Notably, only Gestures 9 (IFE) and 17 (rest) maintain performance levels close to those of the unaltered model, as shown in Figure 60, suggesting that certain features critical for these gestures are less impacted or adequately compensated by the remaining active filters. However, the confusion matrix in Figure 61 b) indicates emerging biases in the model, as many incorrect predictions are misclassified as Gesture 17 (rest), which underscores the model's reduced ability to differentiate between gestures effectively. On the other hand, Cluster 1 demonstrates a relatively lesser impact at the same 25% ablation level. Although its accuracy decreases to 61.12%, as shown in Table 6, it retains a significantly higher prediction capacity compared to Cluster 2. According to Figure 60, gestures such as 12 (WE) and 17 (rest) in Cluster 1 experience notable declines, in addition to those identified at lower ablation

percentages. Notably, Gesture 3 (TLFO), which had previously declined, rebounds to its original prediction level. The confusion matrix presented in Figure 61 a) shows that while some misclassifications are evident, they do not reveal any systematic bias within the model. This suggests a level of robustness in the remaining filters of Cluster 1, maintaining a more diverse and effective gesture recognition capability even as a quarter of its filters are disabled.

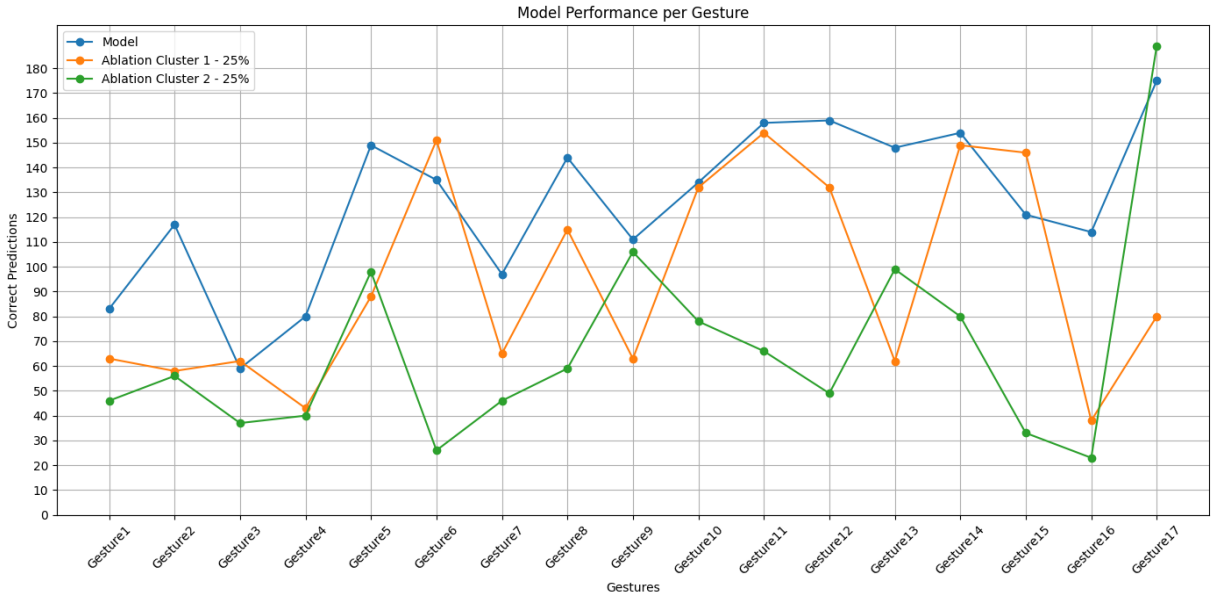
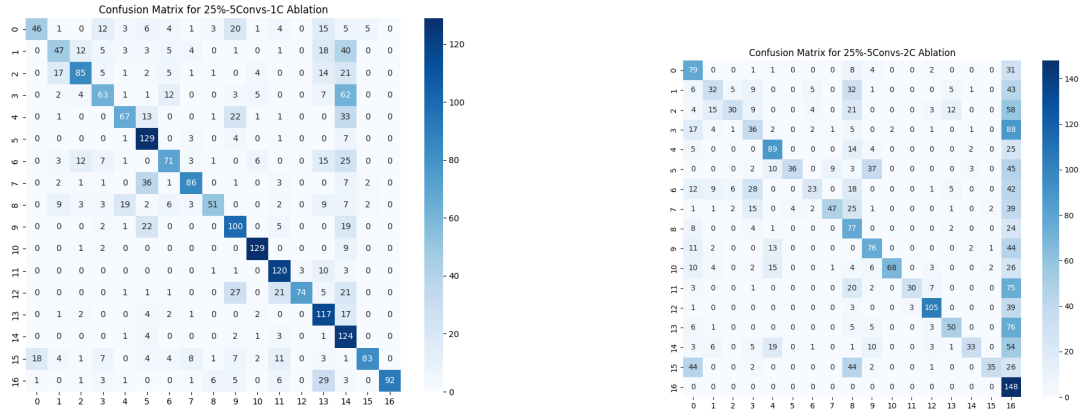


Figure 60: Model Performance per gesture, comparison between original model, and the general ablation of 25% for both Clusters.



(a) Cluster 1 **(b)** Cluster 2

Figure 61: Confusion matrix of general ablation 25% for Cluster 1 (a) and Cluster 2 (b).

During the 50% general ablation, the two Clusters exhibit markedly different results. Cluster 1 maintains an accuracy of 46.99%, whereas Cluster 2's accuracy plummets to 6.10%, as shown in Tables 6 and 7. This stark contrast highlights the varying impacts of filter deactivation across the network. Cluster 1 has demonstrated a consistent linear decline in performance since the

10% ablation level. Notably, Gestures 6 (TIFE) and 14 (FS) continue to achieve a relatively high number of correct predictions, standing out as consistently resilient throughout the ablation study. This suggests that these gestures may not rely as heavily on the features processed by the now compromised filters of Cluster 1. The confusion matrix for Cluster 1, Figure 63 a), supports this observation, showing no evidence of bias and a more random distribution of predictions across various gestures. This indicates potential redundancies or simpler feature mappings within Cluster 1 that facilitate these gestures under significant network impairment. Conversely, at this level of ablation, Cluster 2 appears to have reached its functional threshold and becomes completely ineffective. As depicted in Figure 62, the model fails to distinguish between different gestures effectively. An exception is Gesture 17 (rest), which anomalously maintains accurate predictions. However, this is likely not indicative of the model's effectiveness but rather points to a bias where the model defaults to predicting Gesture 17 (rest) across various inputs. This significant dysfunction in classification capabilities is further evidenced in its confusion matrix, Figure 63 b), where all predicted gestures are erroneously classified as Gesture 17 (rest), rendering the model practically unusable. This divergence in performance between the two Clusters at 50% ablation underscores the critical nature of the filters in Cluster 2 for maintaining overall model accuracy and functionality.

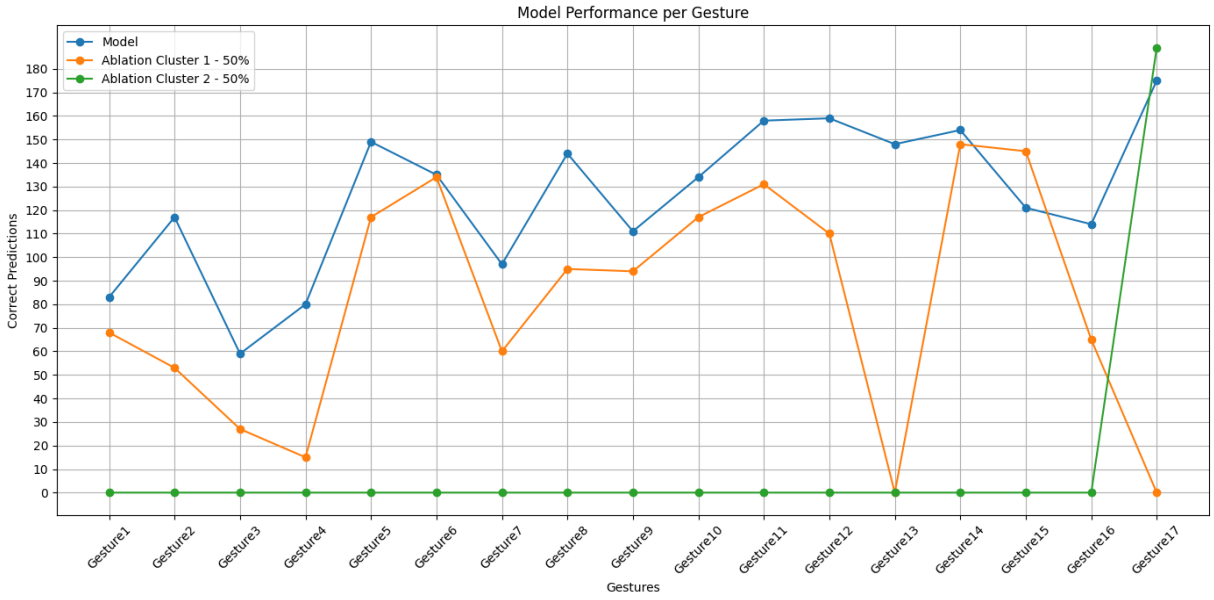
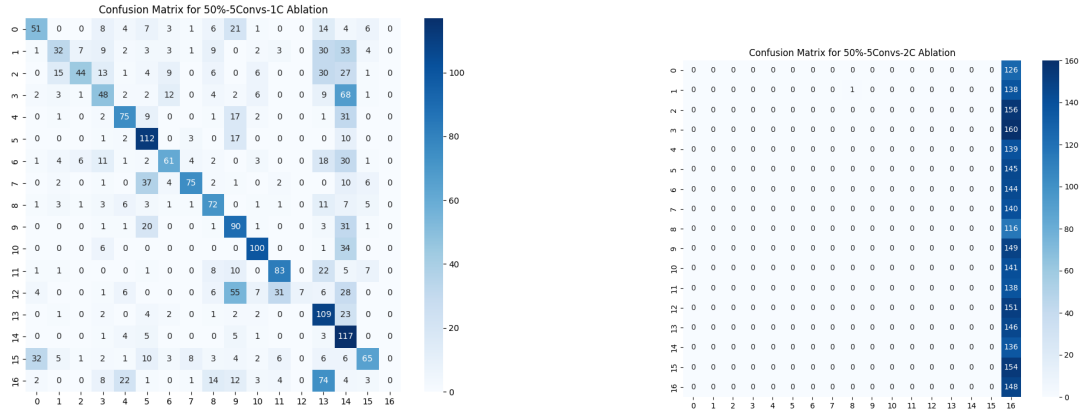


Figure 62: Model Performance per gesture, comparison between original model, and the general ablation of 50% for both Clusters.



(a) Cluster 1 **(b)** Cluster 2

Figure 63: Confusion matrix of general ablation 50% for Cluster 1 (a) and Cluster 2 (b).

At the 75% general ablation level, it becomes evident that the model is largely unusable for both Cluster 1 and Cluster 2, as demonstrated in Tables 6 and 7. Accuracy plummets to 18.95% for Cluster 1 and remains at 6.10% for Cluster 2, indicating severe impairments across the network. Figure 64 extends observations from previous ablation levels, confirming that Cluster 2 continues to exhibit no predictive capability beyond the 50% ablation mark. This consistent lack of performance at a 75% ablation rate underscores that any further reduction in active filters does not impact its already minimal functionality. In contrast, Cluster 1, while still following its trend of linear performance decline, shows a significant change at this stage. Notably, Gesture 14 (FS), which previously managed to maintain a semblance of correct predictions, now registers zero correct predictions. This indicates that although Cluster 1 has been more resilient than Cluster 2, its capacity to sustain gesture recognition under severe network impairment is inherently limited. The inability to accurately predict Gesture 14 (FS), at this ablation rate suggests that certain gestures, which may have less complex feature requirements, were still being processed to some degree by the remaining active filters in Cluster 1. However, this residual functionality is finally depleted as the ablation percentage increases. Furthermore, the confusion matrix in Figure 65 b) reveals that the model begins to show some bias towards predicting Gestures 6 (TIFE), 7 (IMFE), and 15 (HO), while other gestures receive zero predictions. This emerging pattern of bias indicates that the network, while overwhelmed, defaults to recognizing only a few specific gestures, reflecting a critical breakdown in its generalization capability.

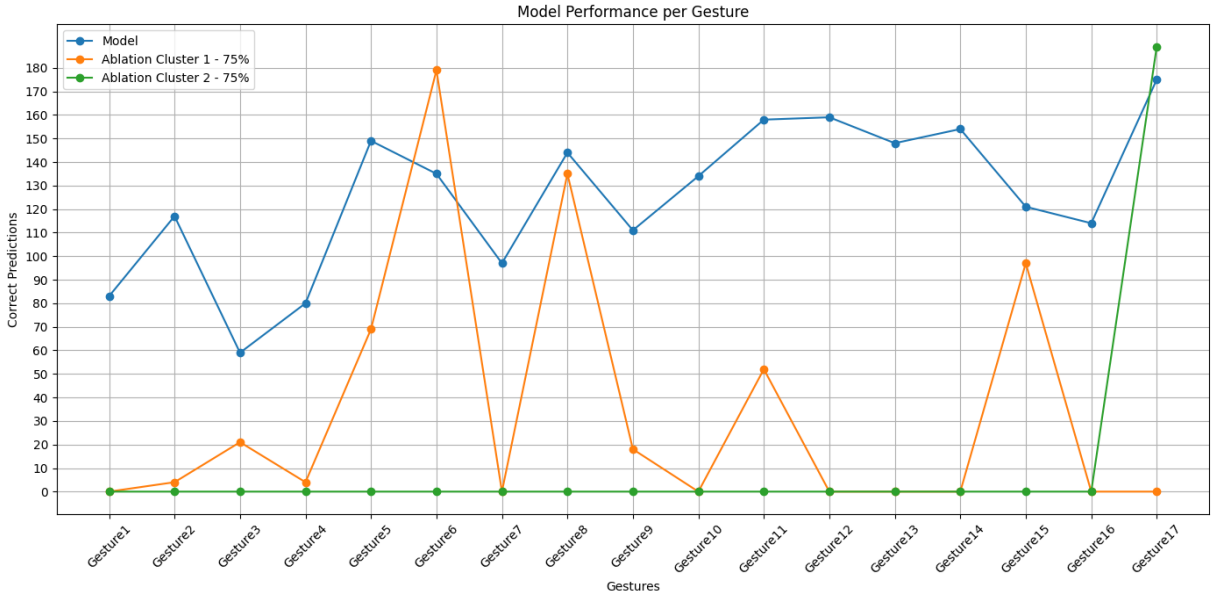
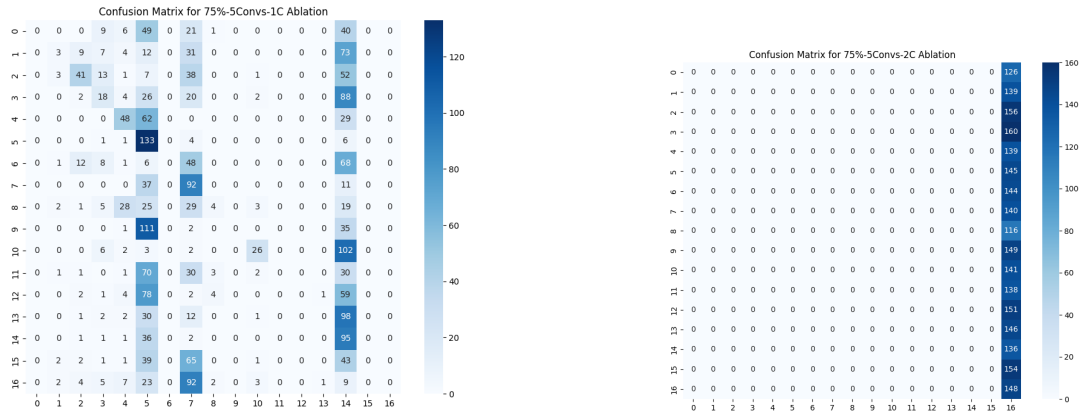


Figure 64: Model Performance per gesture, comparison between original model, and the general ablation of 75% for both Clusters.



(a) Cluster 1 **(b)** Cluster 2

Figure 65: Confusion matrix of general ablation 75% for Cluster 1 (a) and Cluster 2 (b).

The general ablation study has elucidated significant insights into the network's resilience and the distinct functional roles of each Cluster in gesture recognition. Through comprehensive analysis at various ablation percentages (1%, 5%, 10%, 25%, 50%, and 75%), it is evident that the network's Clusters exhibit different thresholds of fault tolerance, each critical to the overall functionality and robustness of the model.

Cluster 1 demonstrated a gradual and consistent decline in performance as ablation levels increased, suggesting that its filters contribute incrementally to gesture recognition. This resilience, particularly for gestures like 6 (TIFE) and 13 (FP), indicates that these gestures may rely on less complex features or benefit from greater redundancy within this Cluster.

Cluster 2, in stark contrast, showed a threshold effect, with performance dropping sharply and significantly beyond certain ablation levels, particularly starting at 25%. This sudden loss of functionality underscores the Cluster's role in processing complex gesture features critical to the model's performance, highlighting a vulnerability due to a lack of redundancy.

These findings illustrate a delicate balance between redundancy and essential functionality within the neural network. While some redundancy allows the network to withstand minor ablations without significant performance loss, the exposure of critical thresholds reveals the indispensable nature of certain filters.

5.2 Single Ablation Results

With a comprehensive understanding of the general ablation impacts, we now advance to dissecting the effects of layer-specific ablations. This subsequent phase of analysis will concentrate on evaluating how individual layers contribute to the recognition of specific gestures under similar ablation pressures. By isolating the effects within single layers, we aim to delineate the unique contributions and resilience of each layer within the network's architecture.

This layer-specific exploration will enhance our insights into optimal layer configurations, potentially leading to improved network designs that bolster fault tolerance and operational efficiency. We will scrutinize the performance of each gesture under layer-specific ablations to pinpoint which layers are most pivotal for their recognition and how these layers interplay to sustain network functionality amidst partial impairments.

The forthcoming detailed analysis will leverage both quantitative data and visual representations to offer a granular view of the network's operational dynamics, further enriching our understanding of the critical nature of network components and their interactions within the model's architecture.

Table 8: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 1% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2144	0.8830	0.3664
Layer 2	2141	0.8847	0.3628
Layer 3	2133	0.8909	0.3462
Layer 4	2144	0.8892	0.3477
Layer 5	2139	0.8888	0.3474

Table 9: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 1% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2138	0.8888	0.3478
Layer 2	2124	0.8855	0.3612
Layer 3	2132	0.8880	0.3447
Layer 4	2138	0.8880	0.3512
Layer 5	2134	0.8851	0.3490

The results from Tables 8 and 9 demonstrate that low percentages of layer-specific ablation, such as the 1% shown, do not significantly impact the model, similar to findings from the general ablation study. This lack of impact holds true across all convolutional layers, affirming the robustness of the network. The performance metrics derived from the ablation of individual layers, as shown in the tables 8 and 9 and supported by graphical representations in Appendix C (Figures C.117 to C.121), confirm that even with a 1% ablation, the variations in model performance remain within the typical deviation observed in the general ablation analysis. This consistency suggests that the isolated layer ablation, even at this low level, does not exceed the model's inherent performance variability, aligning with the earlier findings where multiple

layers were ablated simultaneously.

Table 10: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 5% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2004	0.8703	0.3981
Layer 2	2134	0.8641	0.4054
Layer 3	2126	0.8797	0.3684
Layer 4	2134	0.8863	0.3519
Layer 5	2147	0.8847	0.3583

Table 11: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 5% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2095	0.8616	0.4397
Layer 2	2108	0.8756	0.3904
Layer 3	2108	0.8892	0.3465
Layer 4	2137	0.8847	0.3528
Layer 5	2124	0.8843	0.3472

The results from Tables 10 and 11 indicate that, similar to the 1% ablation, a 5% layer-specific ablation does not significantly affect the majority of convolutional layers. However, the effects begin to emerge, particularly noticeable in the first convolutional layer of both Clusters, where deviations exceed the standard model's performance variability.

In Cluster 1, Table 10 details that the first convolutional layer achieved 2004 correct predictions. From Figure 66, it is evident which gestures are beginning to show impacts due to this level of ablation. Notably, gestures 1 (LP), 3 (TLFO), 4 (TIFO), 5 (TLFE), 7 (IMFE), 8 (LFE),

9 (IFE), 11 (WF), and 16 (HC) exhibit some performance degradation, with gestures 1 (LP), 5(TLFE), and 16 (HC) experiencing the most significant declines. Interestingly, although the first convolutional layer shows higher accuracy than the second, 0.8703 and 0.8641 correspondently, Table 10 show more correct predictions for the second convolutional layer. This discrepancy is highlighted in Appendix C (Figure C.122), where predictions for the second convolutional layer are consistently close to those of the original model, except for gesture 9 (IFE).

For Cluster 2, as shown in Table 11, the first convolutional layer is the only one exhibiting deviations outside the expected model performance, though the impacts are not as pronounced as in Cluster 1. Figure 66 illustrates that Cluster 2 first convolutional layer outperforms that of Cluster 1, despite a notable performance drop in gesture 10 (TE). This specific decline in Cluster 2, contrasted with the stable performance of the same gesture in Cluster 1, suggests a reliance on filters within Cluster 2 for recognizing this particular gesture.

Tables 10 and 11 demonstrate that subsequent convolutional layers, beyond the first, are less affected by the 5% ablation, maintaining performance within the model's standard deviation. This observation is corroborated by visual data in Appendix C (Figures C.122 to C.125), which show minimal to no deviation in model predictions compared to the original model across these layers.

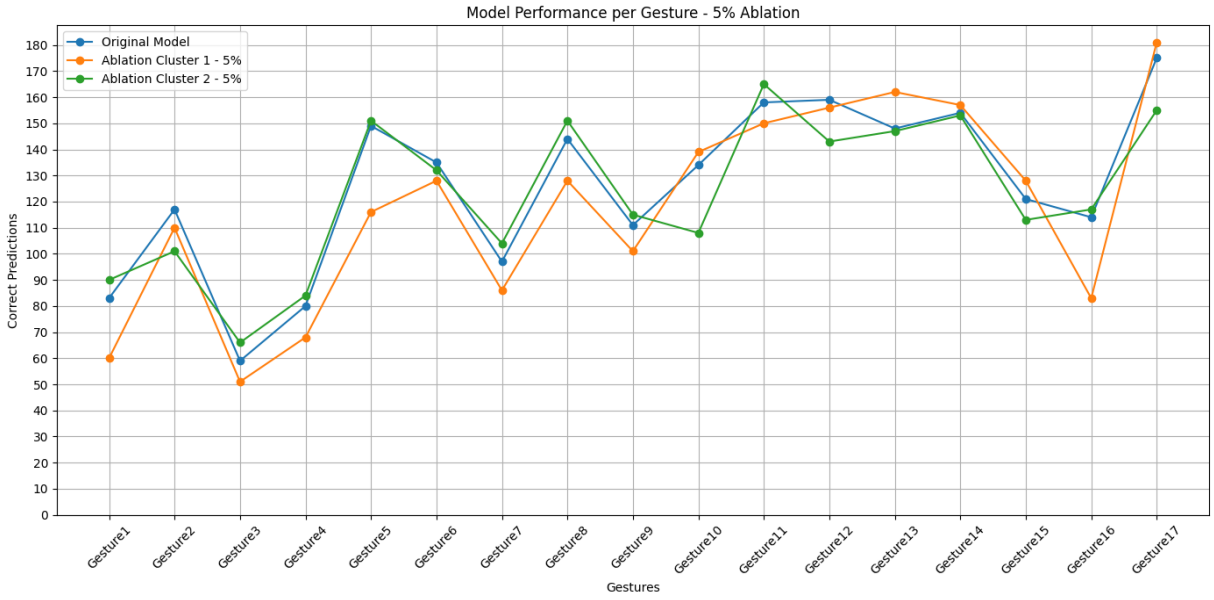


Figure 66: Model Performance per gesture, comparison between original model, and the single layer ablation of 5% for both Clusters of the first convolutional layer.

Table 12: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 10% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2013	0.7834	0.6698
Layer 2	2157	0.8488	0.4731
Layer 3	2108	0.8740	0.3875
Layer 4	2121	0.8843	0.3620
Layer 5	2123	0.8793	0.3683

Table 13: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 10% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	2006	0.8476	0.4514
Layer 2	2114	0.8756	0.3790
Layer 3	2102	0.8756	0.3700
Layer 4	2131	0.8744	0.3749
Layer 5	2116	0.8834	0.3546

At the 10% ablation level, while the model continues to perform adequately, a noticeable decline in accuracy is evident across the first convolutional layers, as demonstrated in Tables 12 and 13. These results deviate from the typical pattern observed in less intensive ablations. In Cluster 1, the first convolutional layer experiences a significant accuracy drop to 78.34%, as detailed in Table 12. Figure 67 illustrates a noticeable decline in predictions for gestures 1 (LP), 4 (TIFO), 7 (IMFE), 10 (TE), and 16 (HC). Not all gestures affected by the 5% ablation exhibit significant drops at this stage, suggesting that certain gestures are particularly sensitive to the filters in this convolutional layer. It is also noteworthy that the second convolutional layer, while showing results outside the typical deviation with an accuracy of 84.88%, maintains superior

prediction results, even outperforming the original model. This is visually corroborated by the graph in Appendix C (Figure C.126), which shows minor fluctuations in some gestures compared to the original model results. Other convolutional layers in Cluster 1, as presented in Appendix C (Figures C.127 to C.129), do not exhibit significant deviations, except for a notable decrease in gesture 13 (FP) prediction capabilities in the fifth convolutional layer.

Detailed in Table 13, Cluster 2 first layer, while sensitive to ablation, maintains an accuracy of 84.76% with 2006 correct gestures predicted. Subsequent layers continue to perform comparably to the original model, showing little variation among them. Figure 67 highlights that, overall, the model largely maintains its performance. However, gestures such as 2 (TA), 8 (LFE), 10 (TE), 12 (WE), 13 (FP), 15 (HO), and 16 (HC) show performance drops.

Notably, gesture 10 (TE), previously identified in the 5% ablation as sensitive, again shows diminished results, reinforcing the possibility that essential filters for accurately predicting this gesture reside in this layer. Contrarily, this specific gesture does not exhibit prediction losses in other convolutional layers, as depicted in in Appendix C (Figures C.126 to C.129.

The results from the 10% ablation reveal that while the model generally upholds a good level of performance, certain gestures and convolutional layers exhibit heightened sensitivity to filter reduction. These layers, particularly the first in both Clusters, contain critical filters whose impairment significantly affects gesture recognition, especially for gestures that consistently show vulnerability across ablation percentages.

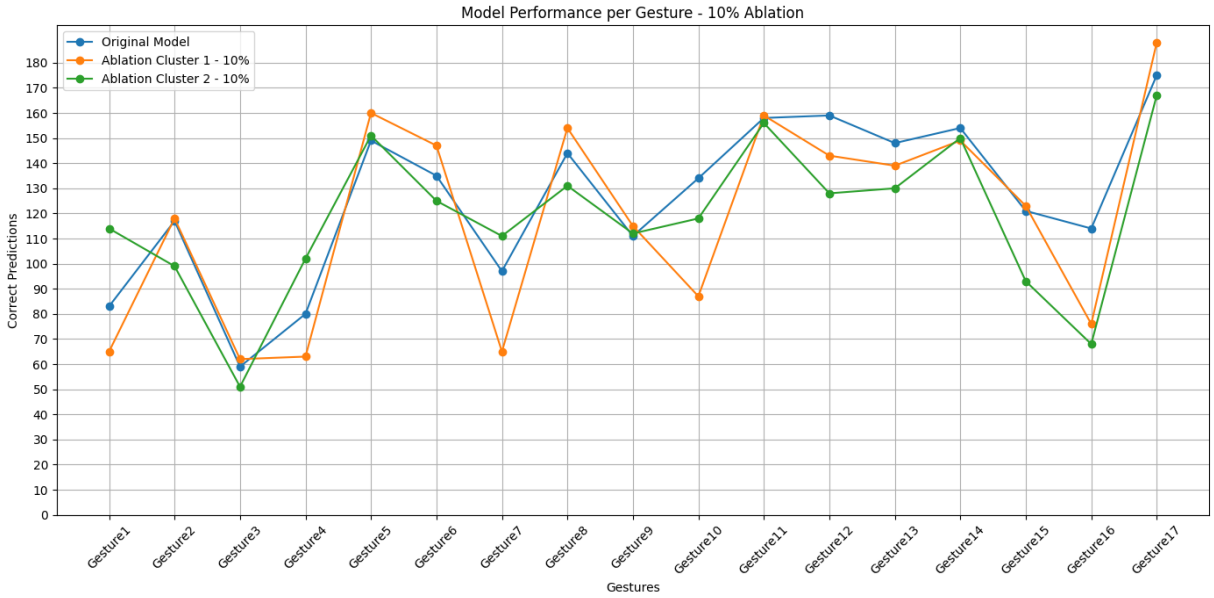


Figure 67: Model Performance per gesture, comparison between original model, and the single layer ablation of 10% for both Clusters of the first convolutional layer.

Table 14: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 25% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1828	0.7657	0.6950
Layer 2	2009	0.7681	0.6951
Layer 3	2038	0.8237	0.5284
Layer 4	2039	0.8369	0.4701
Layer 5	2117	0.8756	0.3851

Table 15: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 25% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1999	0.7681	0.7494
Layer 2	1815	0.6960	1.0421
Layer 3	1971	0.8007	0.5794
Layer 4	2088	0.8488	0.4378
Layer 5	2101	0.8740	0.3792

Following insights from the 10% ablation study, we observed significant impacts at the 25% ablation level as detailed in Tables 14 and 15. Notably, all layers were affected by this level of ablation, except for the fifth convolutional layer in Cluster 1, which performed within the model's typical deviation range, as evidenced in Table 14. This suggests a dependency on all convolutional layers for maintaining model performance, with some layers having a more pronounced impact on gesture prediction.

Across Cluster 1, there was a performance decline in all convolutional layers except the fifth, indicating widespread sensitivity to filter reduction. The first convolutional layer was the most affected, with accuracy dropping to 76.57% and correct predictions at 1828, as shown

in Table 14. Interestingly, Figure 68 reveals that despite the reduction in filters, gestures such as 7 (IMFE), 10 (TE), and 13 (WE) still performed close to baseline levels, highlighting their resilience. However, other gestures including 4 (TIFO) and 16 (HC) experienced significant performance deterioration.

The second convolutional layer, while closely matching the first in accuracy, demonstrated enhanced gesture recognition capabilities. This is illustrated in Figure 69, where gestures 2 (TA), 9 (IFE), 10 (TE), 12 (WE), 13 (WE), 16 (HC), and 17 (rest) showed varying impacts, suggesting the presence of essential filters in this layer crucial for maintaining performance stability.

The third and fourth layers exhibited similar accuracy results but differed in gesture predictions. While the third layer struggled with gestures 4 (TIFO), 10 (TE), 13 (WE), and 16 (HC), as depicted in Figure 70, the fourth layer had challenges with a different set, including gestures 1 (LP), 2 (TA), 3 (TLFO), 5 (TLFE), 6 (TIFE), 8 (LFE), and 15 (HO), as represented in Figure 71. This variance highlights the unique role of filters in these layers.

Despite not deviating outside the typical model pattern, the fifth layer showed a notable performance drop in gesture 13 (FP) as seen in Appendix C (Figure C.130), indicating specific vulnerabilities even within a generally resilient layer.

In Cluster 2, all layers exhibited a decline in performance at the 25% ablation mark, with the second layer showing the most substantial deterioration dropping to 69.90% accuracy and only 1815 correct predictions, as shown in Table 15. Figure 69 shows a significant reduction in performance for gestures 1 (LP), 6 (TIFE), 7 (IMFE), 8 (LFE), 10 (TE), 11 (WF), 12 (WE), 13 (FP), 15 (HO), and 16 (HC) in this layer, underscoring its sensitivity.

Interestingly, the first layer also suffered performance drops in gestures 3 (TPFO), 6 (TIFE), 7 (IMFE), 8 (LFE), 13 (FP), 15 (HO), and 16 (HC), as depicted in Figure 68, which had been previously identified as vulnerable. This indicates the critical role of the first and second layers in recognizing these gestures, supported by similar findings in Cluster 1, suggesting that these gestures are highly susceptible to ablations.

The third and fourth layers exhibited similar accuracy results but differed in gesture predictions. While the third layer struggled with gestures 4 (TIFO), 10 (TE), 13 (FP), and 16 (HC), as depicted in Figure 70, the fourth layer had challenges with a different set, including gestures 1 (LP), 2 (TA), 3 (TLFO), 5 (TLFE), 6 (TIFE), 8 (LFE), and 15 (HO), as represented in Figure 70. This variance highlights the unique role of filters in these layers.

At the 25% ablation mark, the study highlights the critical interdependencies within the network's layers. Each layer's unique filter contributions are essential for accurate gesture recognition, and significant ablation leads to noticeable performance drops across specific gestures, particularly those dependent on the first and second convolutional layers

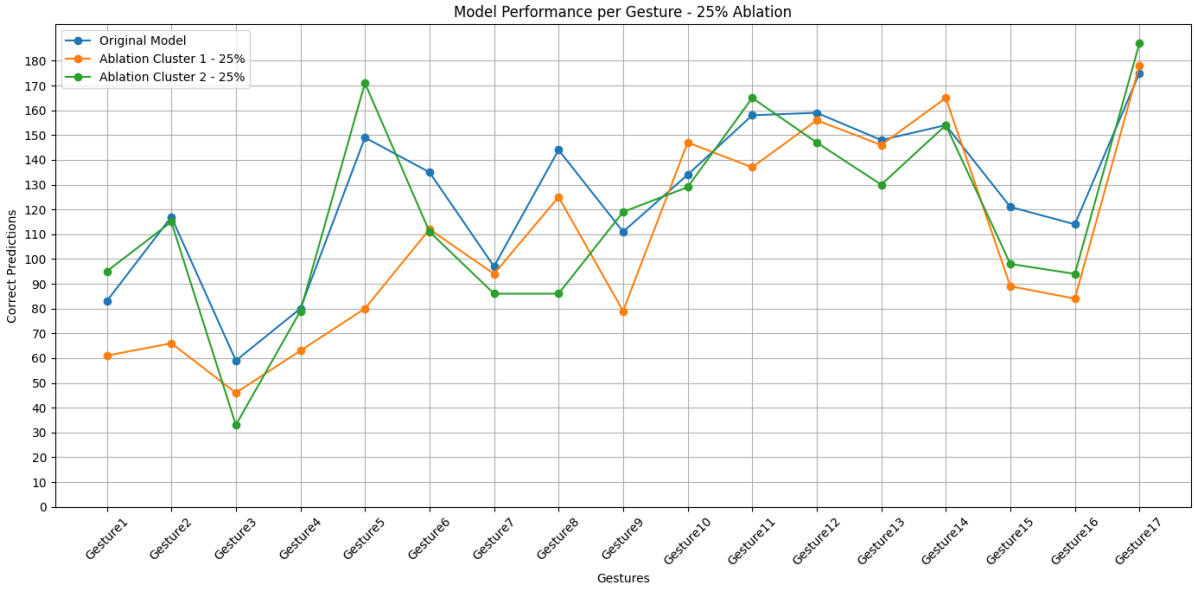


Figure 68: Model Performance per gesture, comparison between original model, and the single layer ablation of 25% for both Clusters of the first convolutional layer.

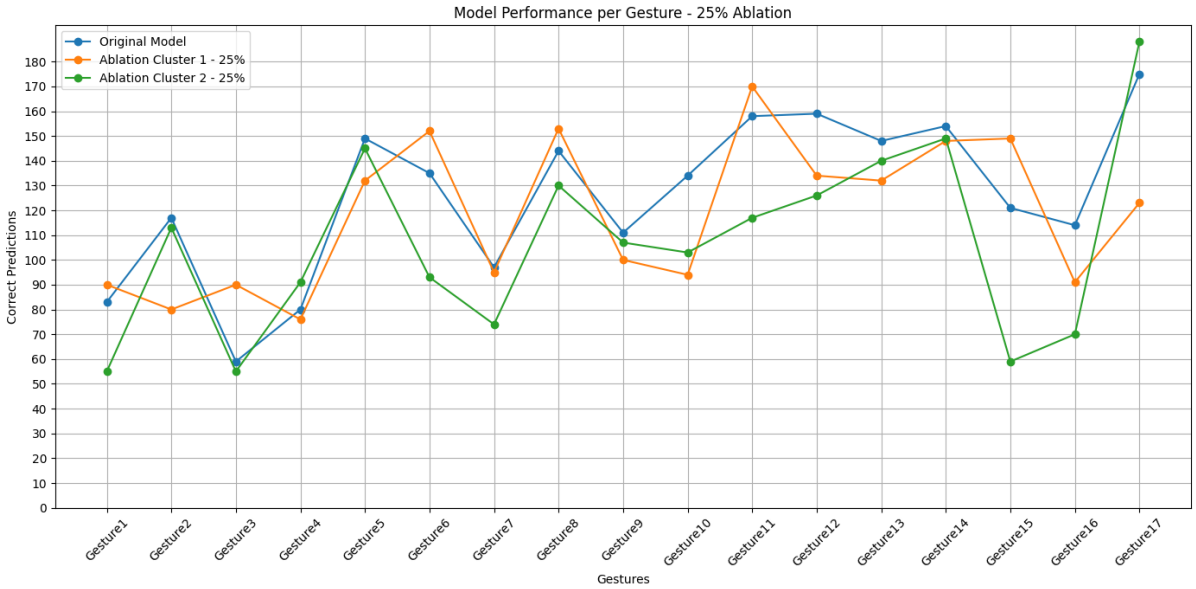


Figure 69: Model Performance per gesture, comparison between original model, and the single layer ablation of 25% for both Clusters of the second convolutional layer.

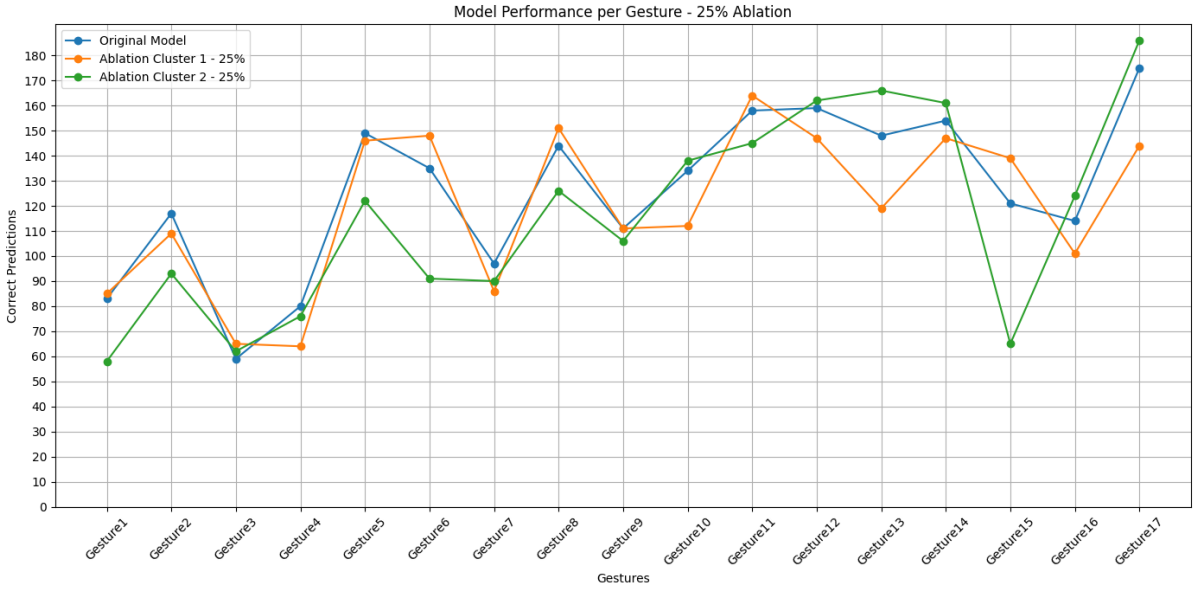


Figure 70: Model Performance per gesture, comparison between original model, and the single layer ablation of 25% for both Clusters of the third convolutional layer.

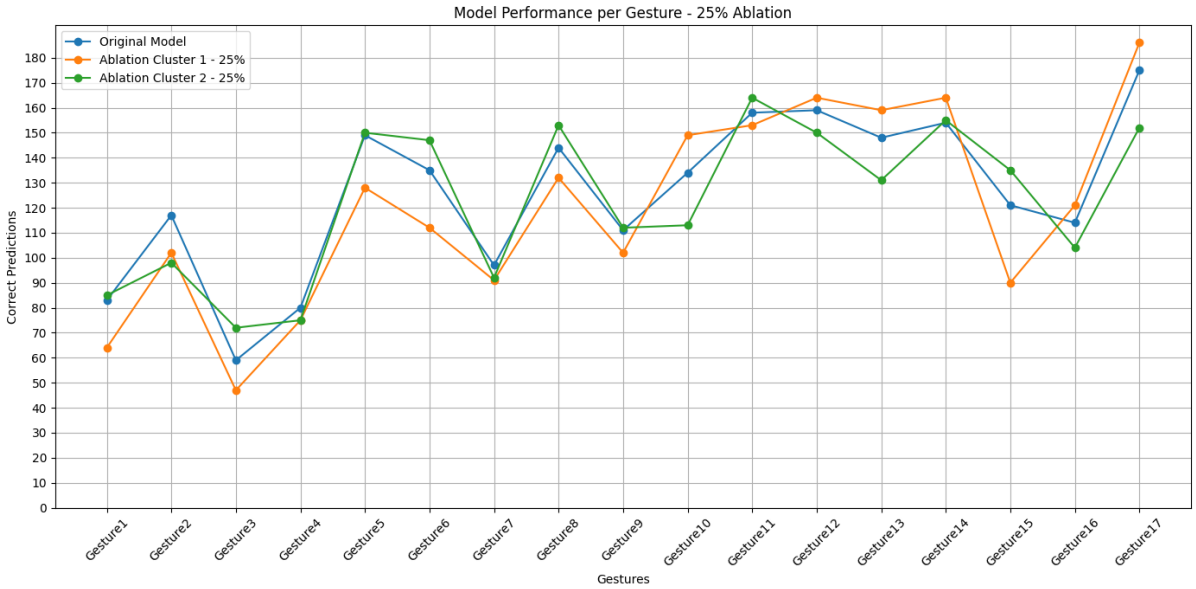


Figure 71: Model Performance per gesture, comparison between original model, and the single layer ablation of 25% for both Clusters of the fourth convolutional layer.

Table 16: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 50% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1234	0.4848	2.0327
Layer 2	1512	0.5379	1.7964
Layer 3	1714	0.6351	1.3181
Layer 4	1803	0.7022	0.9227
Layer 5	2081	0.8381	0.4742

Table 17: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 50% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1454	0.4798	2.0497
Layer 2	538	0.1845	5.4714
Layer 3	1459	0.5605	1.6440
Layer 4	1981	0.7496	0.7405
Layer 5	2007	0.8295	0.4720

In the 50% ablation there was a significant impact of in the results of the model, this can be seen in Table 16 and 17, in which all layers can be seen as being heavily affected by the 50% ablation, while the fifth convolutional layers presents results above 80%, its noticeable that this convolutional layer starts to being affected at this ablation percentage, demonstrating that this layer contains the less important filters to gesture classification in both Cluster 1 and Cluster 2

In Cluster 1, every layer exhibited a decline in performance, but the first convolutional layer was the most severely impacted, with accuracy plummeting to 48.48%. Figure 72 highlights that almost all gestures suffered a substantial drop in prediction accuracy. Interestingly, gestures 1 (LP), 10 (TE), and 14 (FS) remained close to the baseline model predictions. Gesture 10 (TE),

which was also resilient in the 25% ablation, suggests a lower dependence on this particular layer. However, the confusion matrix in Figure 73 shows widespread misclassification, including frequent incorrect predictions as gesture 10 (TE), indicating complex model behavior rather than explicit bias

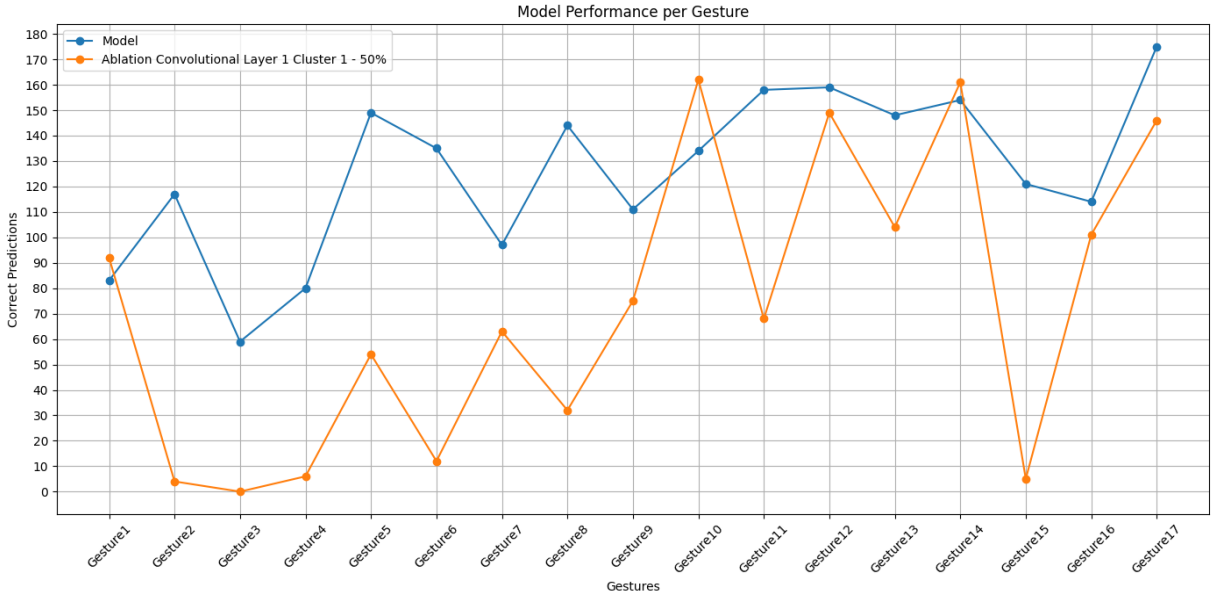


Figure 72: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 1 of the first convolutional layer.

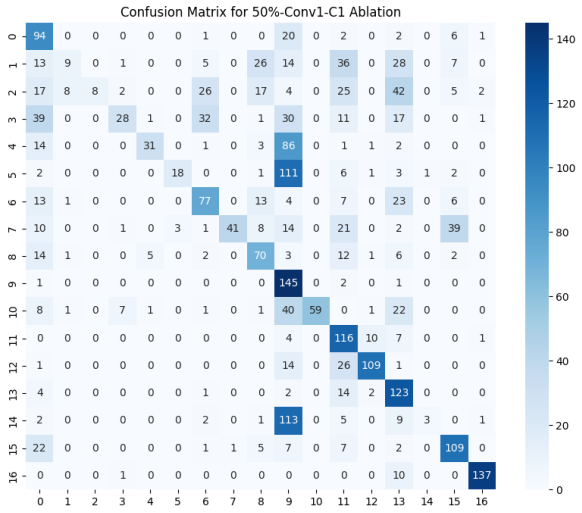


Figure 73: Confusion matrix of single layer ablation of 50% for Cluster 1 of the first convolutional layer.

The second convolutional layer recorded an accuracy of 53.79%, with performance variations evident in Figure 74. Some gestures maintained performance, while others, previously impacted at the 25% ablation, plummeted further, emphasizing the critical nature of this layer's filters

for specific gestures. The confusion matrix in Figure 75 shows dispersed misclassifications, indicating a lack of bias despite the drastic performance drop.

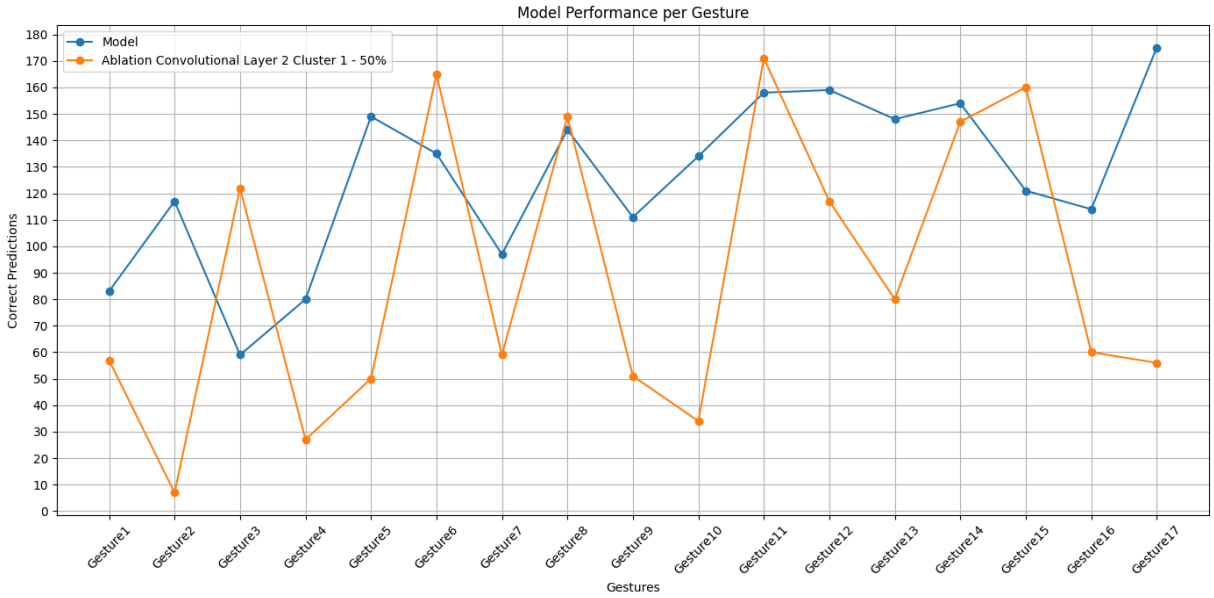


Figure 74: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 1 of the second convolutional layer.

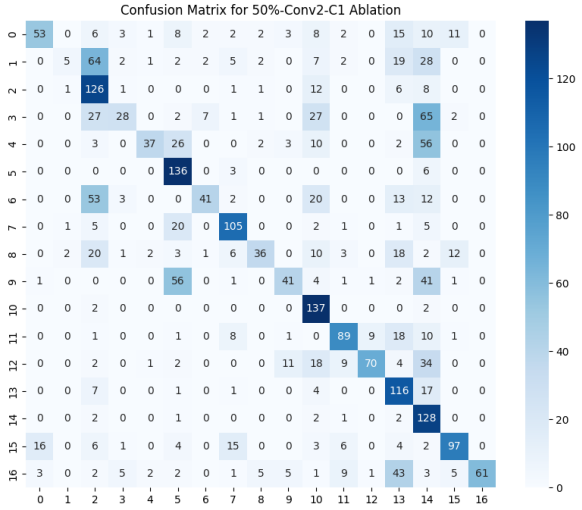


Figure 75: Confusion matrix of single layer ablation of 50% for Cluster 1 of the second convolutional layer.

The third convolutional layer, while exhibiting minor performance drops, as depicted in Figure 76, maintained reasonable accuracy for gestures consistent with the second layers resilience, suggesting these layers share critical filters. Figure 77 shows this layer ability to perform without bias, despite lower accuracy rates shown in Table 16.

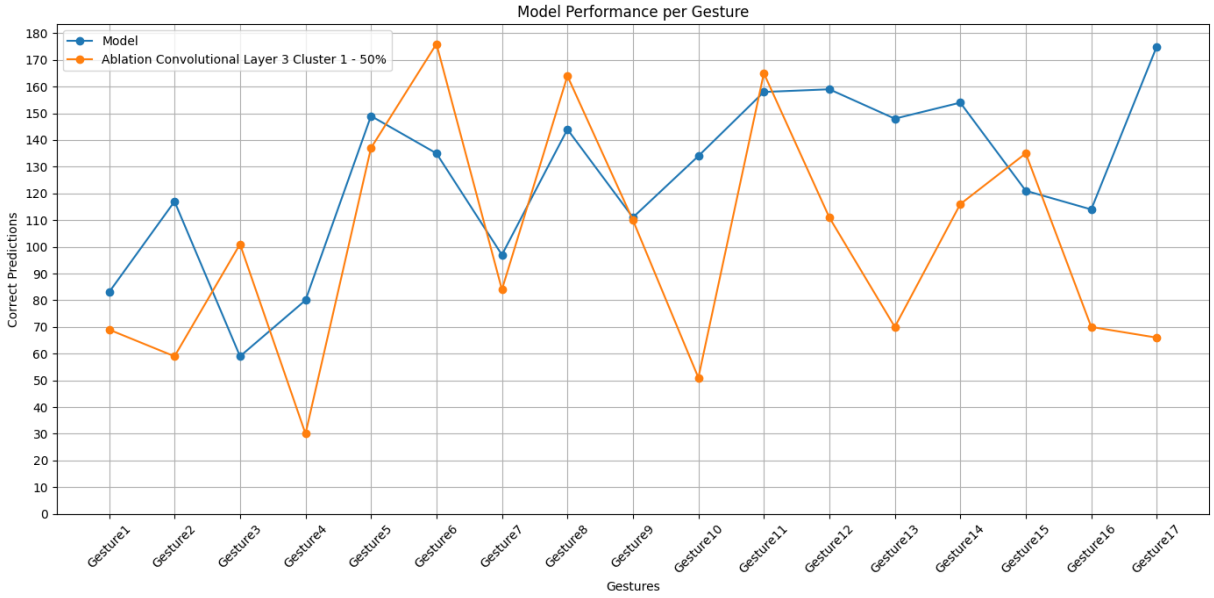


Figure 76: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 1 of the third convolutional layer.

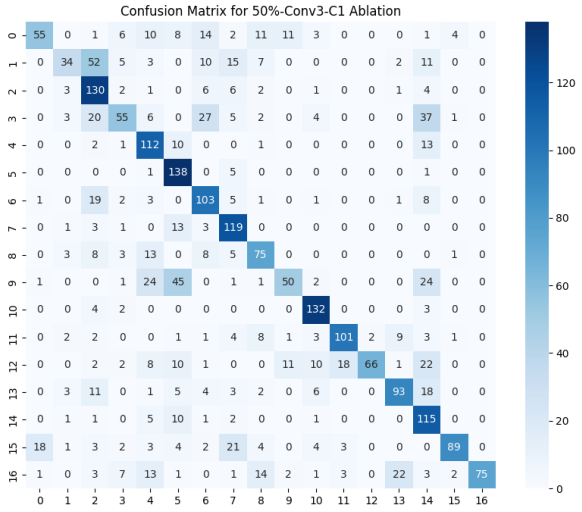


Figure 77: Confusion matrix of single layer ablation of 50% for Cluster 1 of the third convolutional layer.

The fourth and fifth layers, though experiencing performance dips, did not drastically alter the model's insights, as seen in Appendix C (Figures C.135 and C.135). Only gestures 2 (TA) and 15 (HO) in the fourth layer showed a significant decline, hinting at specific dependencies on the filters of this layer.

Cluster 2 displayed different dynamics compared to Cluster 1. Notably, the second convolutional layer suffered dramatically, with accuracy dropping to 18.45%, highlighting its critical role in the network's gesture recognition capabilities, as it can be seen in Table 17 and Figure 78. The corresponding confusion matrix in Figure 79 showed a clear bias, with the model predominantly predicting gesture 17 (rest), indicating a breakdown in network functionality.

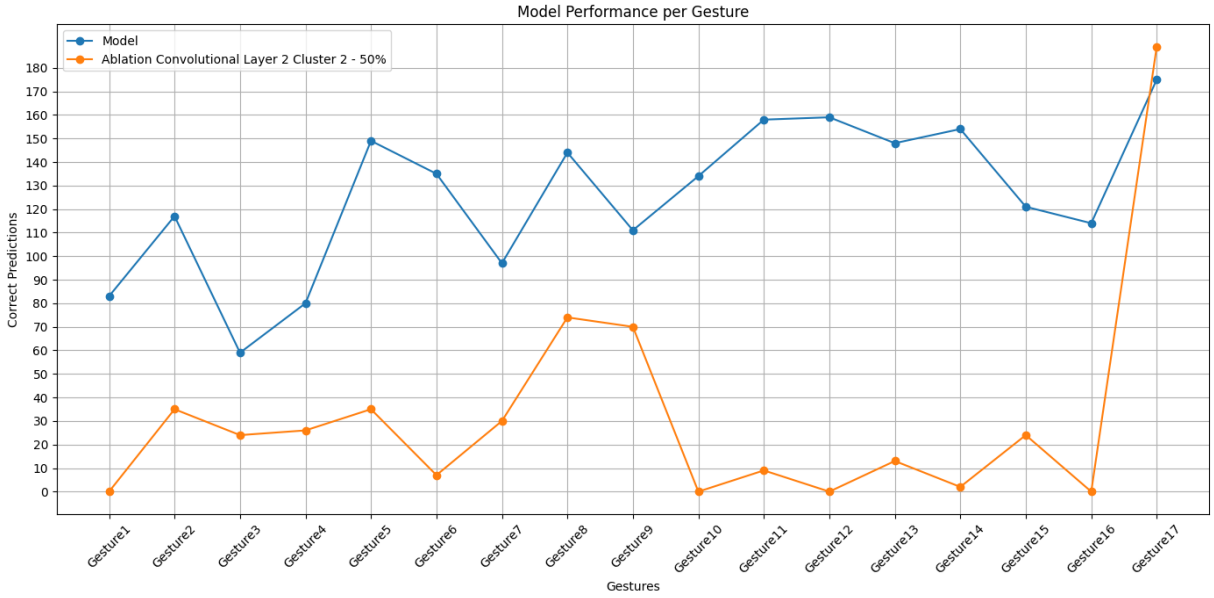


Figure 78: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 2 of the second convolutional layer.

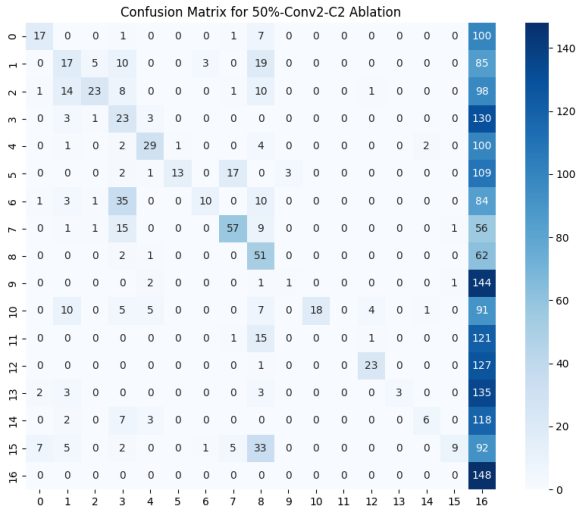


Figure 79: Confusion matrix of single layer ablation of 50% for Cluster 2 of the second convolutional layer.

The first and third convolutional layers in Cluster 2 show closely matched results in terms of gesture prediction accuracy, with the first layer correctly predicting 1454 gestures and the third layer slightly higher at 1459. However, the distribution of these predictions varies significantly between the layers as depicted in Figures 80 and 81. The first convolutional layer mirrors the performance seen in Cluster 1's second and third layers, where gestures such as 6 (TIFE), 9 (IFE), and 11 (WF) closely align with the original model's accuracy.

Conversely, gestures 1 (LP), 5 (TLFE), and 17 (rest) also exhibit high prediction levels, yet no significant bias is evident from the confusion matrix presented in Figure 81. Despite its lower

overall accuracy, this layer maintains sufficient discriminatory power, suggesting its continued utility despite the ablation.

In contrast, the third layer's performance diverges from that of the first. As shown in Figure 82, gestures 1 (LP), 5 (TLFE), 6 (TIFE), 9 (IFE), and 11 (WF), which previously performed well, now fall below their original accuracy levels. However, gestures 2 (TA), 12 (WE), 13 (FP), 14 (FS), 16 (HC), and 17 (rest) maintain performance close to baseline, underscoring the varied impact of ablation on different gestures. This discrepancy highlights the nuanced role of specific filters within these layers, emphasizing their collective importance for model functionality. The corresponding confusion matrix in Figure 83 confirms the absence of bias, with a well-distributed array of misclassifications across the predictions.

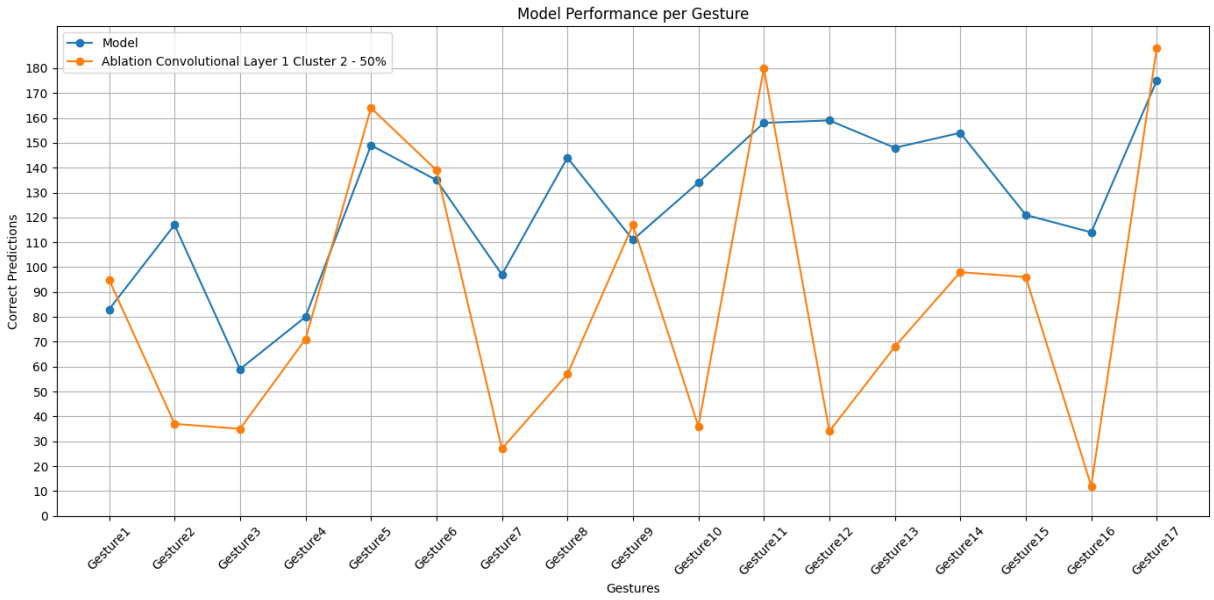


Figure 80: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 2 of the first convolutional layer.

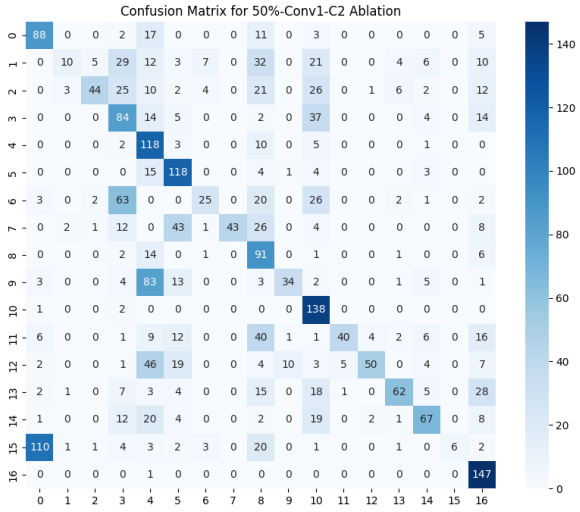


Figure 81: Confusion matrix of single layer ablation of 50% for Cluster 2 of the first convolutional layer.

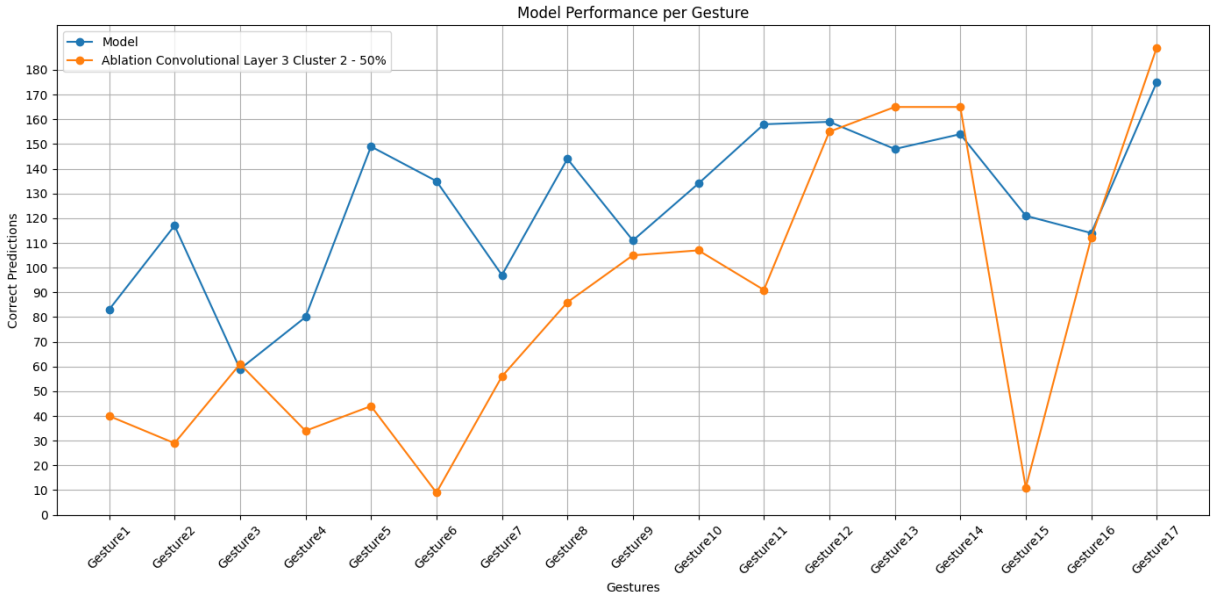


Figure 82: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for Cluster 2 of the third convolutional layer.

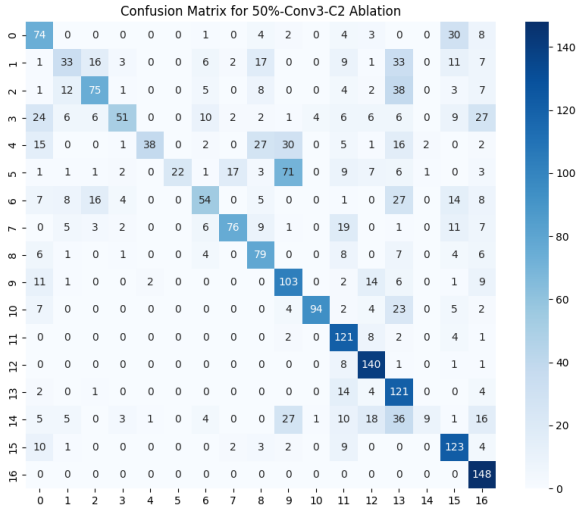


Figure 83: Confusion matrix of single layer ablation of 50% for Cluster 2 of the third convolutional layer.

The fourth and fifth convolutional layer, similar to observations in Cluster 1, show less pronounced impacts on overall model performance. As observed in Figures 80 and 83, these layers appear less critical to recognizing gestures that were most vulnerable in earlier ablation stages. This suggests that with the more crucial layers compromised, the remaining structures in Cluster 2 are capable of compensating effectively, thereby mitigating some of the performance degradation.

Comprehensive comparisons of layer effects across both Clusters for each corresponding layer are detailed in Appendix C (Figures C.131 to C.135).

Table 18: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 75% ablation from Cluster 1

Cluster 1	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1143	0.4049	2.0476
Layer 2	957	0.3122	3.6348
Layer 3	1045	0.3933	2.7517
Layer 4	1341	0.4930	1.8673
Layer 5	1999	0.7685	0.6597

Table 19: Total correct predictions out of the test dataset, accuracy and loss from single layer abaltion with 75% ablation from Cluster 2

Cluster 2	Prediction	Accuracy	Loss
Original	2138	0.8888	0.3478
Layer 1	1008	0.3369	2.8795
Layer 2	189	0.0610	14.4978
Layer 3	895	0.3402	3.2279
Layer 4	1564	0.5758	1.5277
Layer 5	1839	0.7615	0.6737

At the 75% ablation level, the performance of both Clusters is markedly diminished, as evident from the significantly low accuracy and gesture predictability shown in Tables 18 and 19. Most models are rendered nearly unusable due to the substantial reduction in filters. Despite this, the fifth convolutional layer in both Clusters shows a smaller drop in performance (about 10 percentage points in accuracy), suggesting its filters have a lesser impact on gesture predictability compared to other layers.

Across Cluster 1, all layers except the fifth experienced a significant decline, as shown in Table 18. The second convolutional layer was particularly affected, with its accuracy dropping to 31.22% and only 957 gestures correctly predicted, making it the worst- performing layer at this ablation level. Figure 84 highlights that only gestures 3 (TLFO), 6 (TIFE), 11 (WF), and 15 (HO) maintained prediction accuracies close to the original model, similar to the 50% ablation patterns observed in Figure 74. Despite the severe performance reduction, the confusion matrix (Figure 85) indicates a random distribution of misclassifications, suggesting no specific bias in predictions.

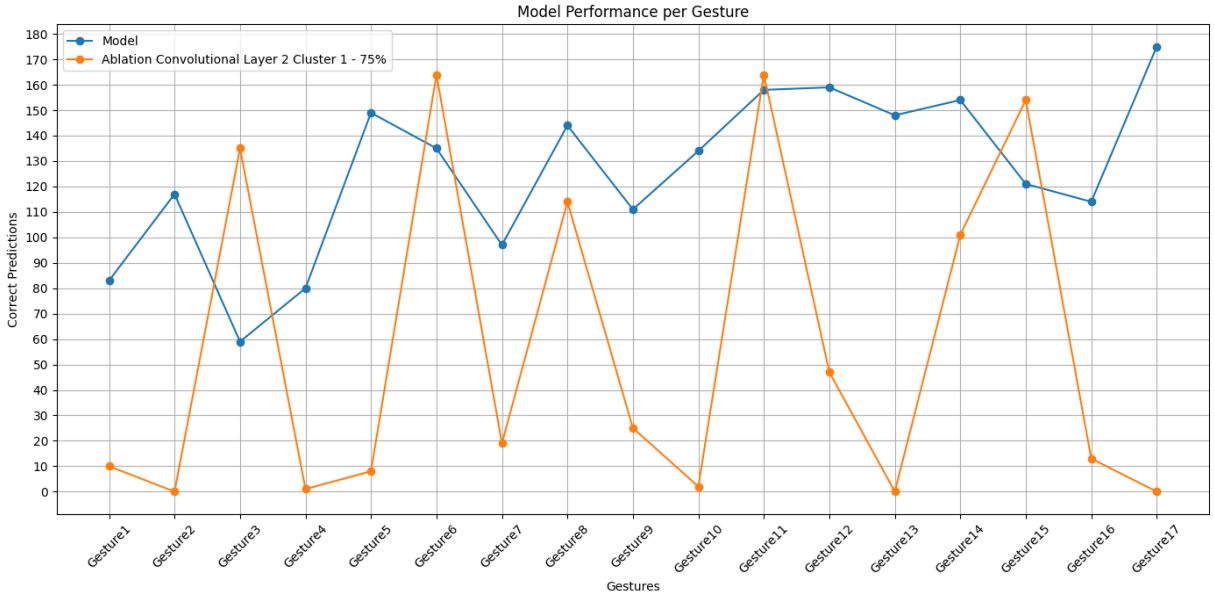


Figure 84: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 1 of the second convolutional layer.

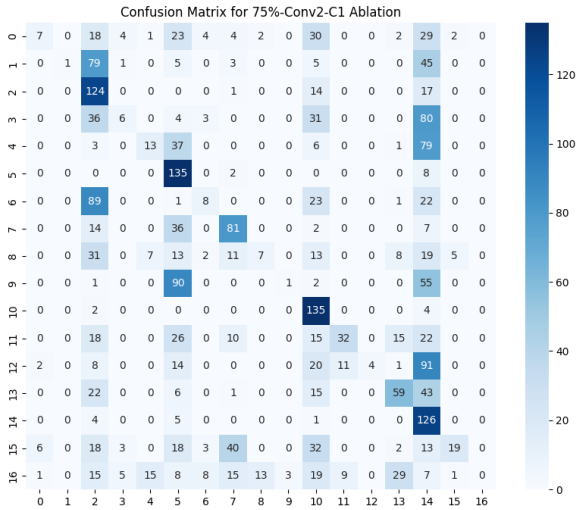


Figure 85: Confusion matrix of single layer ablation of 75% for Cluster 1 of the second convolutional layer.

The first and third layers displayed similar accuracy and number of correct predictions, yet their performance per gesture varied markedly, as depicted in Figures 86 and 88. The first layer managed to sustain classification rates close to the original for gestures 1 (LP), 9 (IFE), and 10 (TE) (Figure 86), consistent with its 50% ablation resilience. Conversely, the third layer showed differentially affected gestures, with 2 (TA), 6 (TIFE), 8 (LFE), and 11 (WF) maintaining closer to baseline performance (Figure 88), pointing to the nuanced roles these layers play in the model's architecture. Both layers confusion matrices (Figures 87 and 89) show dispersed misclassifications, affirming the lack of bias.

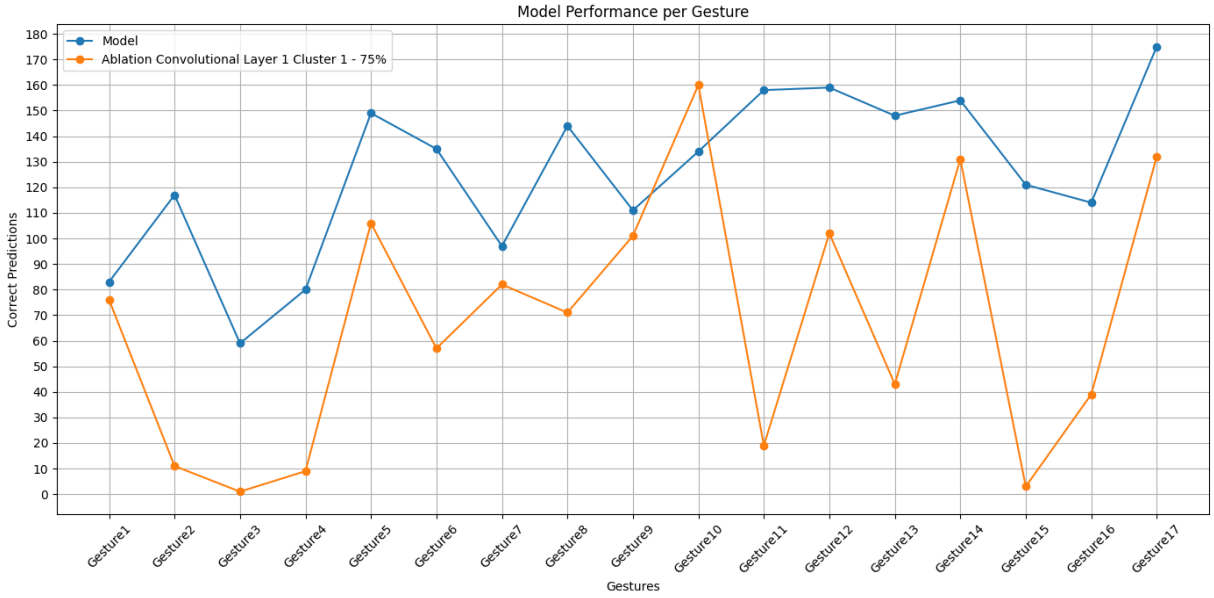


Figure 86: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 1 of the first convolutional layer.

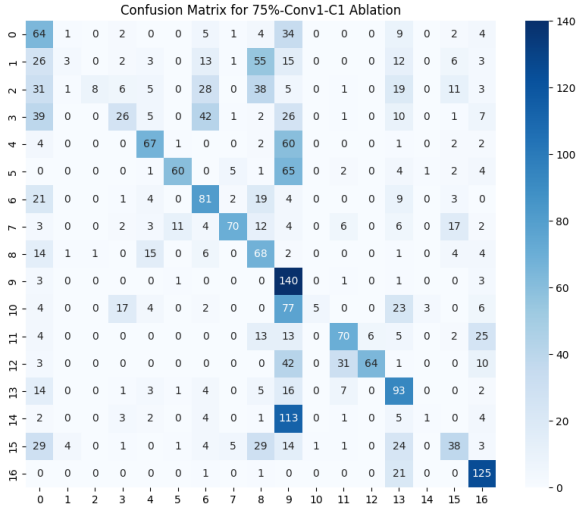


Figure 87: Confusion matrix of single layer ablation of 75% for Cluster 1 of the first convolutional layer.

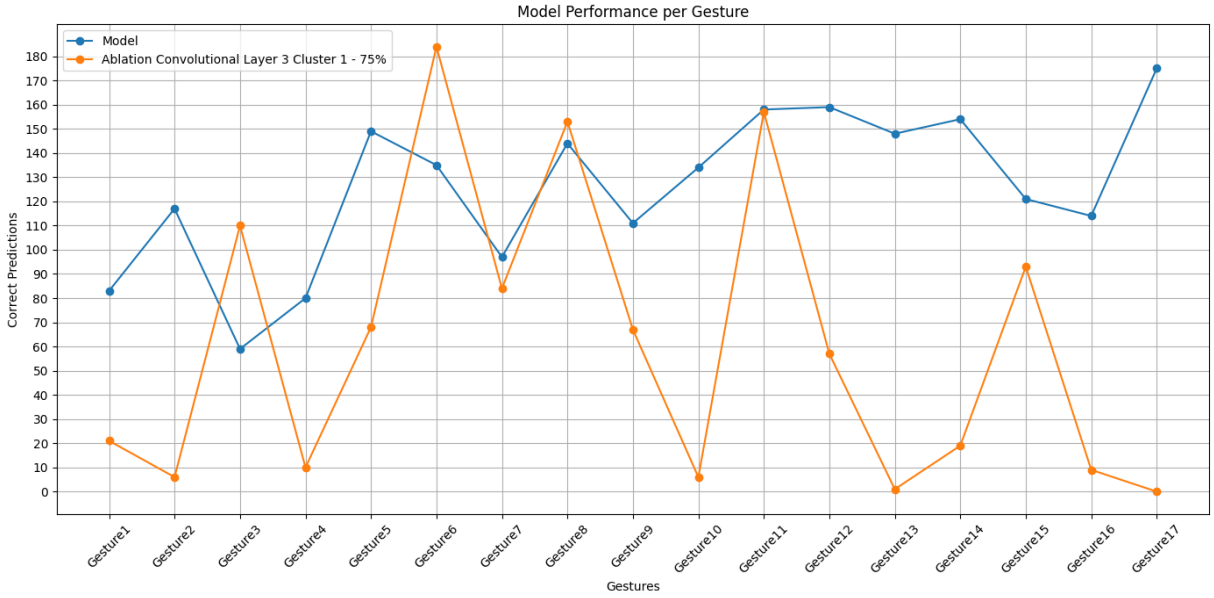


Figure 88: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 1 of the third convolutional layer.

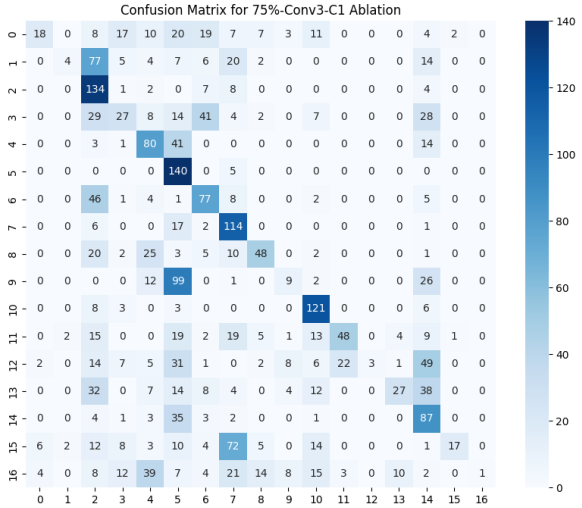


Figure 89: Confusion matrix of single layer ablation of 75% for Cluster 1 of the third convolutional layer.

The fourth layer at this ablation level showed a notable deterioration in performance, with a drop in accuracy to 49.3% as seen in Table 14. Gesture performance analysis (Figure 90) revealed that only gestures 9 (IFE), 10 (TE), 12 (WE), 13 (FP), 14 (FS), and 17 (rest) remained close to baseline levels, while gesture 15 (HO) plummeted significantly, highlighting its sensitivity to this layer's filter reduction. The corresponding confusion matrix (Figure 91) reflects the model's classification struggles without showing signs of bias.

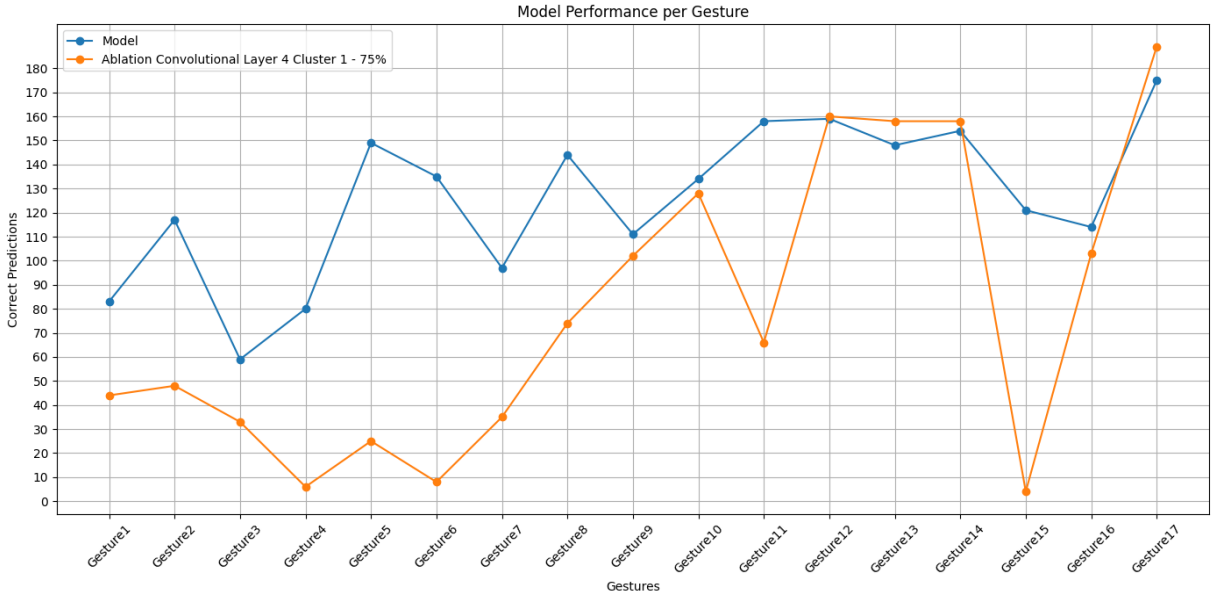


Figure 90: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 1 of the fourth convolutional layer.

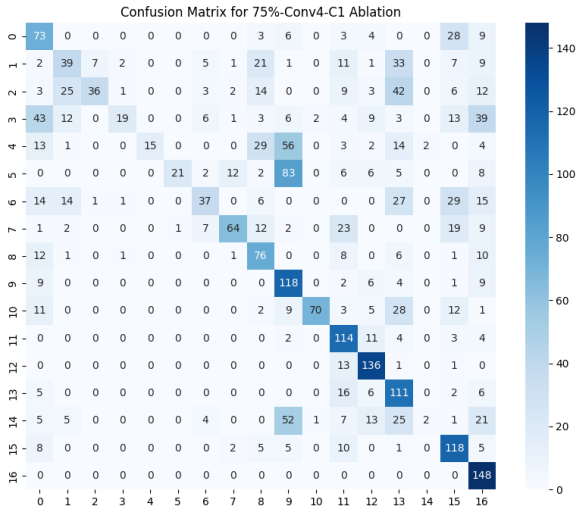


Figure 91: Confusion matrix of single layer ablation of 75% for Cluster 1 of the fourth convolutional layer.

The fifth convolutional layer while presenting some drops in performance at the 75% ablation mark its noticeable its differences comparing to other layers affected by this ablation percentage, as depicted in Table 18 while other models have accuracies bellow the 50% mark the fifth convolutional layer presents results of 76,85%. In Figure 92 its observable that the majority of the gestures were not widely affected by this ablation with only gestures 13 (FP) and 16 (HC) presenting major drops in performance. The confusion matrix illustrated in Figure 93 demonstrates a well distributed prediction classification with some misclassifications, but it does not present any signs of the model being bias.

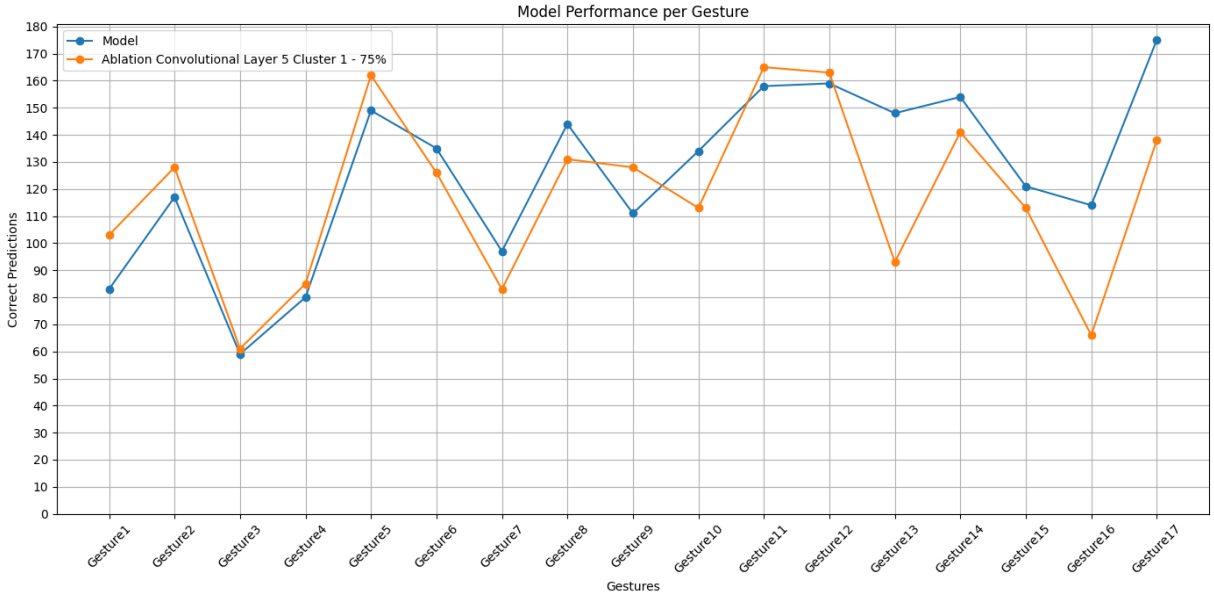


Figure 92: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 1 of the fifth convolutional layer.

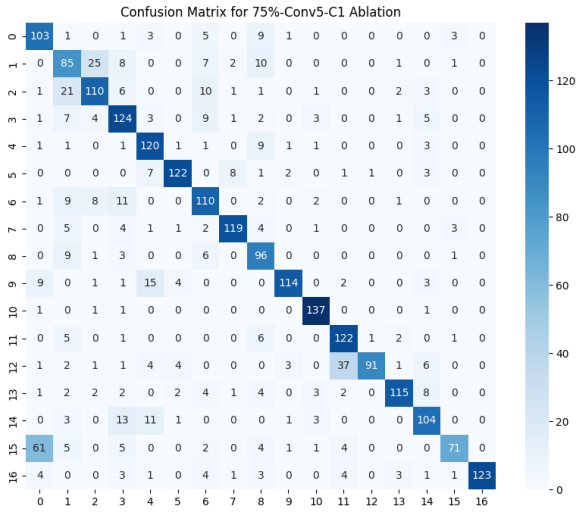


Figure 93: Confusion matrix of single layer ablation of 75% for Cluster 1 of the fifth convolutional layer.

In Cluster 2, the data presents crucial insights that are essential for understanding the limitations and breakdown of the model under severe filter reduction. Table 19 highlights a critical observation in the second convolutional layer where each gesture's total correct prediction is exactly 189, matching the number of images per gesture. This stark result suggests the model has reached a breaking point, likely defaulting to predicting only a single gesture. This pattern of decline continues from the 50% ablation level, with the second convolutional layer suffering the most, followed by the first and third layers, which show similar deterioration. Contrary to Cluster 1, the fourth layer in Cluster 2 maintains better resilience, with accuracy still above

50%, and the fifth layer, despite a significant drop, maintains an accuracy above 75%. The performance per gesture in the second convolutional layer, illustrated in Figure 94, is telling. The layer almost exclusively predicts gesture 17 (rest), as confirmed by the confusion matrix in Figure 95, which shows all images classified as this gesture, marking the model as essentially dysfunctional and biased at this stage of ablation.

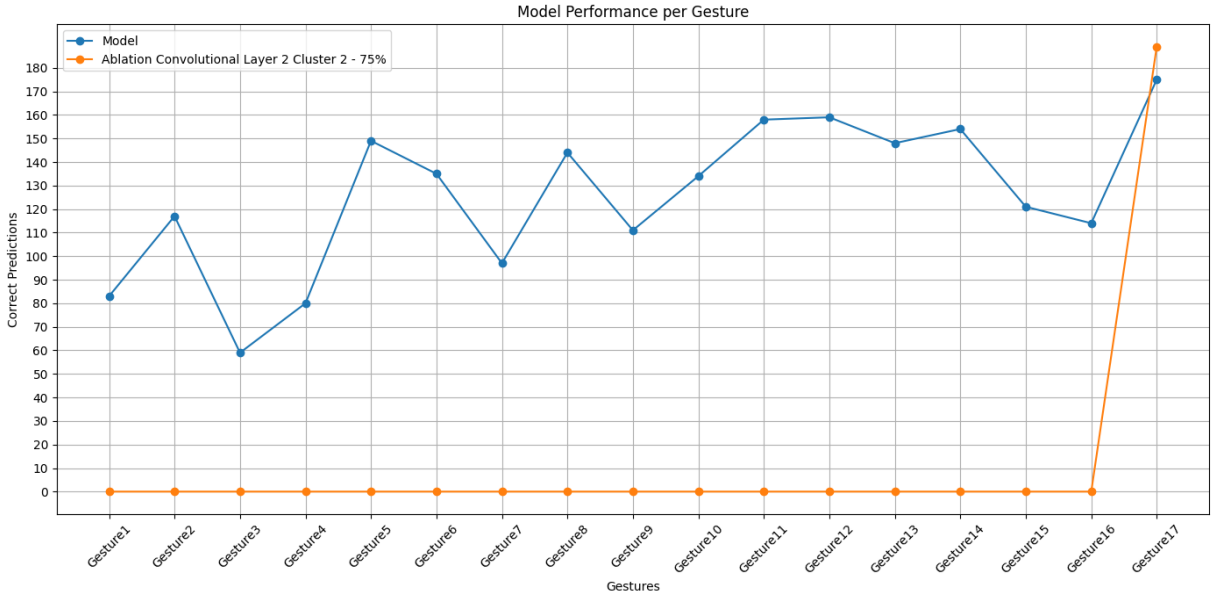


Figure 94: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 2 of the second convolutional layer.

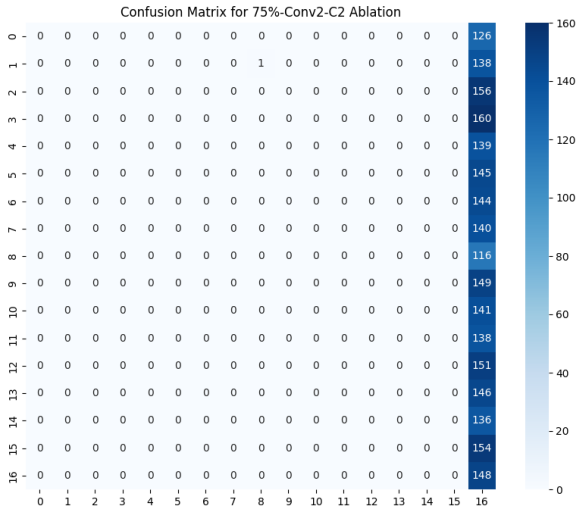


Figure 95: Confusion matrix of single layer ablation of 75% for Cluster 2 of the second convolutional layer.

For the first convolutional layer, the performance per gesture, also documented in Figure 96, exhibits reduced effectiveness compared to the 50% ablation results. It retains some resilience in previously robust gestures but shows a marked decline in others, indicating a deepening impact

of the ablation. The confusion matrix in Figure 97 reveals new trends: a significant number of misclassifications are wrongly attributed to gestures 10 (TE) and 17 (rest), the ones with the best performance in this layer, suggesting the onset of bias.

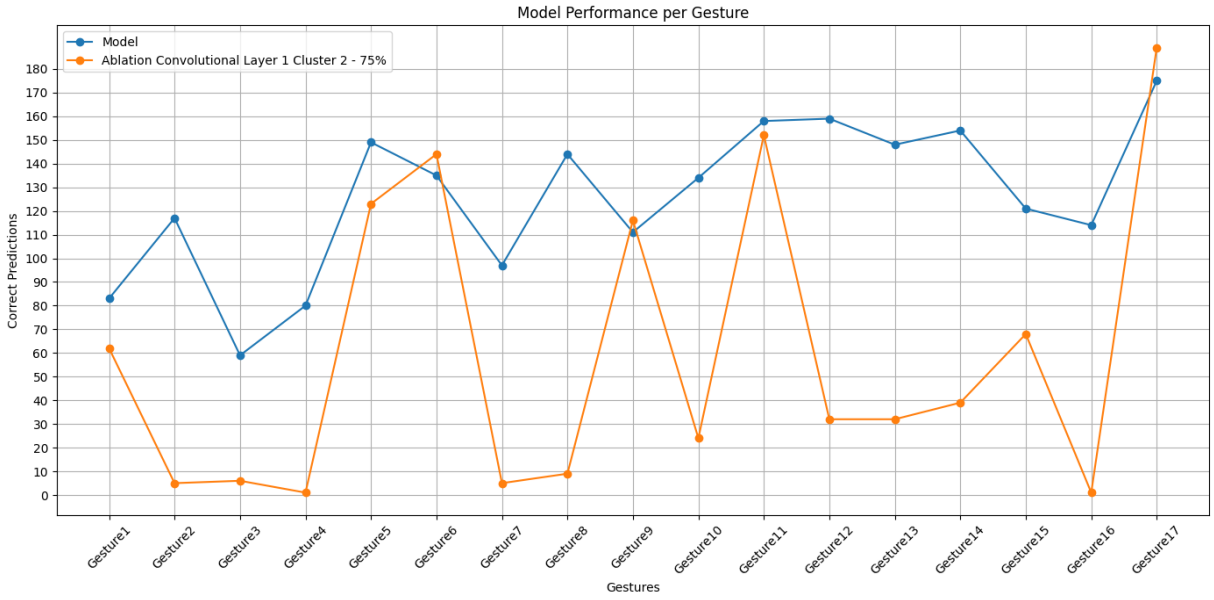


Figure 96: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 2 of the first convolutional layer.

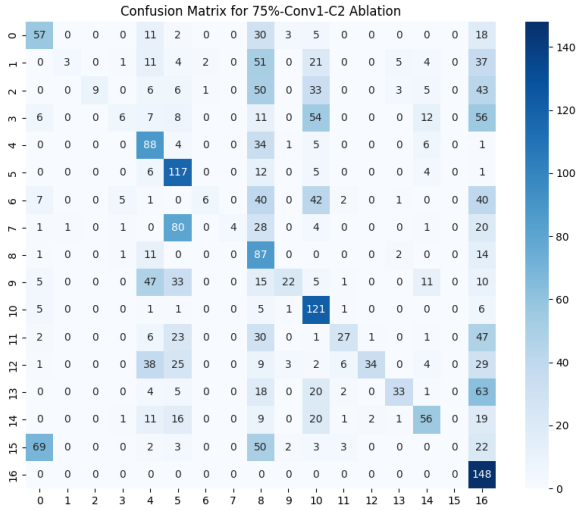


Figure 97: Confusion matrix of single layer ablation of 75% for Cluster 2 of the first convolutional layer.

The third convolutional layer, depicted in Figure 98, shows a significant impact on gestures 1 to 9, mirroring the decline seen at the 50% ablation level but with an enhanced effect. While gestures 9 (IFE) and 17 (rest) still perform relatively well, the overall capability of this layer to classify gestures correctly has deteriorated significantly. The confusion matrix, shown in

Figure 99, indicates that while there is no clear bias, the layer struggles to fulfill its classification role effectively, with a notable number of gestures almost completely misclassified.

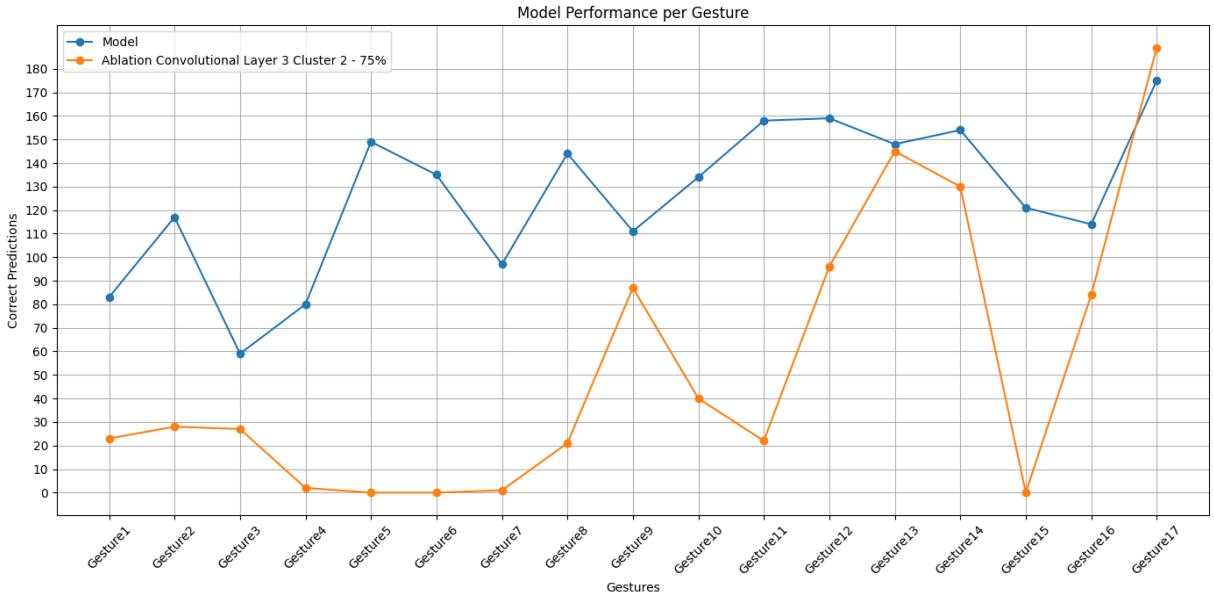


Figure 98: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 2 of the third convolutional layer.

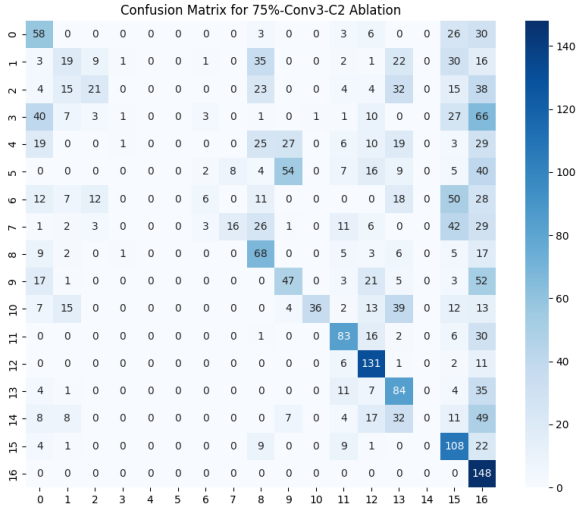


Figure 99: Confusion matrix of single layer ablation of 75% for Cluster 2 of the third convolutional layer.

At 75% ablation, the fourth convolutional layer's performance significantly worsens from previous levels, as seen in Table 19. Gesture prediction performance, detailed in Figure 100, highlights severe drops across most gestures except for 3 (TLFO), 6 (TIFE), 9 (IFE), 11 (WF), and 15 (HO), which remain close to original model levels. The confusion matrix (Figure 101) does not indicate bias, despite the layer's reduced effectiveness.

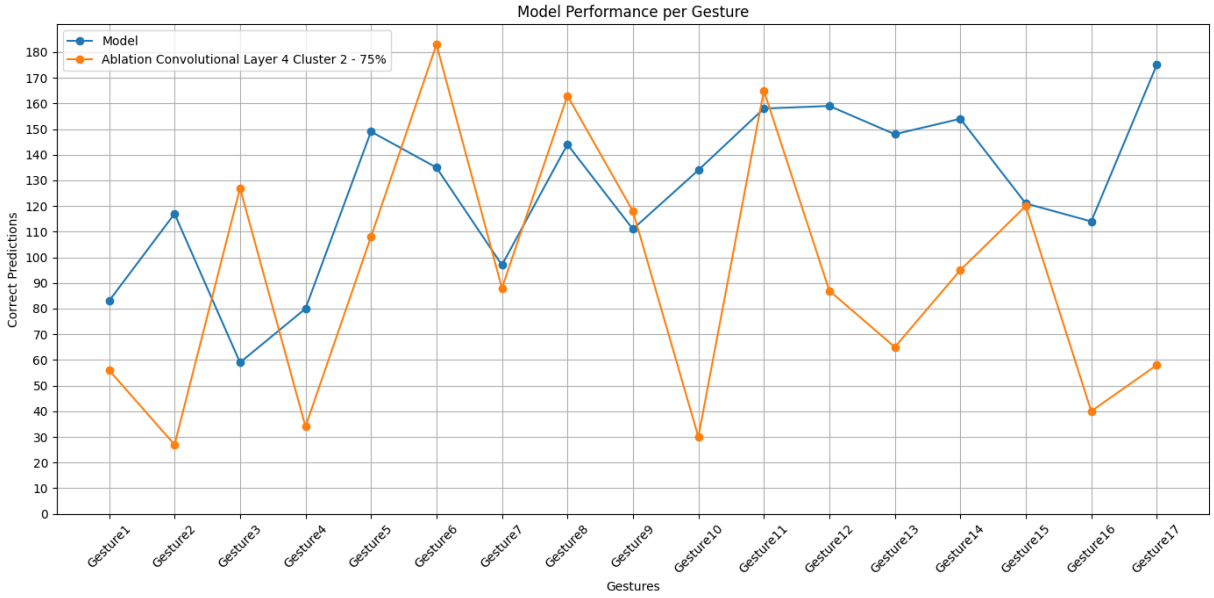


Figure 100: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 2 of the fourth convolutional layer.

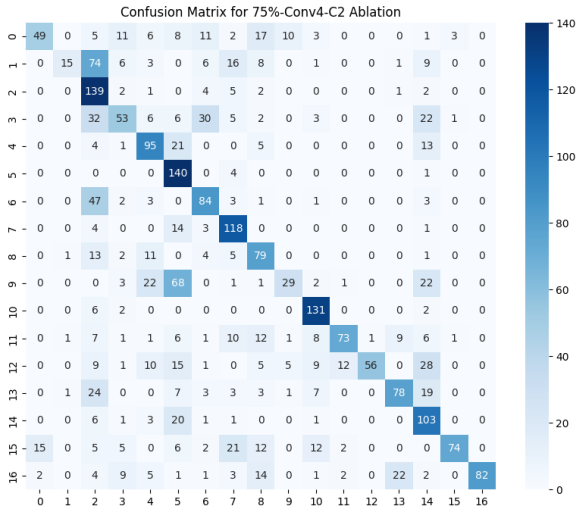


Figure 101: Confusion matrix of single layer ablation of 75% for Cluster 2 of the fourth convolutional layer.

Lastly, the fifth convolutional layer's resilience is notable, despite a drop in performance, its impact is less severe compared to other layers, as detailed in Figure 102. While gesture prediction accuracy falls below the original model, the layer does not exhibit major disruption to the overall model's functionality, maintaining essential accuracy for the network.

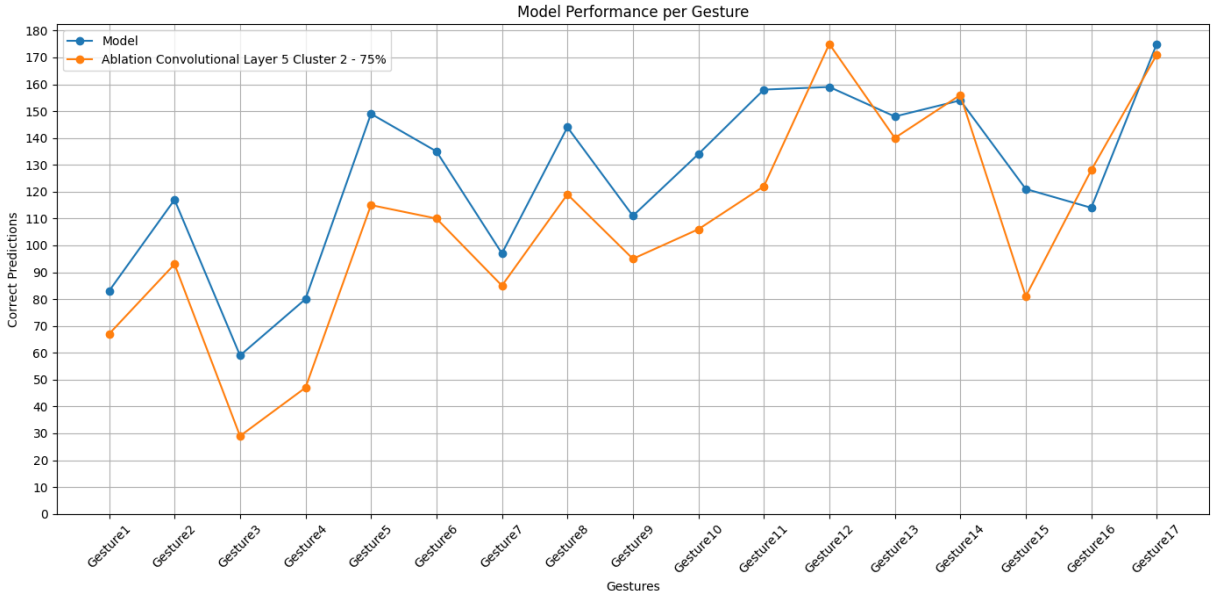


Figure 102: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for Cluster 2 of the fifth convolutional layer.

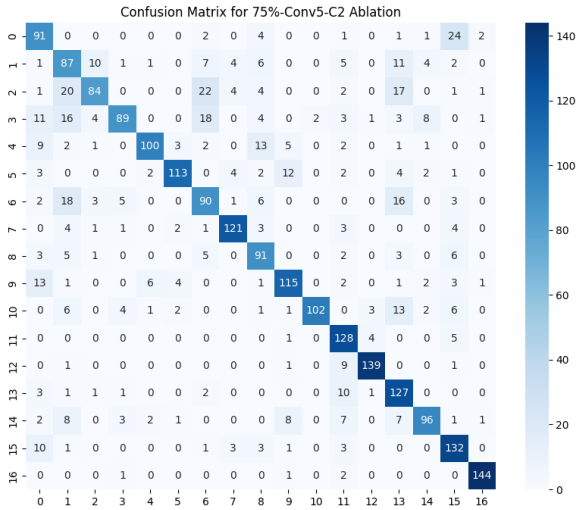


Figure 103: Confusion matrix of single layer ablation of 75% for Cluster 2 of the fifth convolutional layer.

The layer-specific ablation study provided deeper insights into the contributions of individual layers to the overall performance of the neural network. While some layers displayed considerable resilience, maintaining gesture recognition capabilities even at higher ablation levels, others were more vulnerable, showing significant performance drops. This variance among layers highlights the specialized roles they play in gesture recognition. The study illustrates that while some layers are critical for capturing complex features essential for accurate predictions, others contribute more to the network's robustness and can compensate for losses in their counterparts.

The ablation studies conducted, both general and layer-specific, clearly delineate the differential impact of layer ablation on gesture recognition capabilities within a CNN utilizing the AlexNet architecture. Early convolutional layers, particularly Layers 1 and 2, are identified as crucial for accurately classifying gestures due to their role in processing primary visual features such as edges, contours, and basic shapes. These foundational features set the stage for all subsequent gesture recognition processes.

When minor percentages of filters across all layers are deactivated (1% to 10%), the network exhibits inherent resilience, suggesting redundancy within these layers that mitigates minor losses. However, as filter deactivation extends to greater percentages (50% to 75% observed in general ablations), the compounded loss of functionality in these crucial layers results in a sharp decline in overall network performance.

This is particularly notable when Layers 1 and 2 are specifically targeted in single-layer ablations, underscoring their significant impact. The specific ablation of these layers highlights their indispensable role in maintaining a diverse gesture recognition capability.

For Cluster 1, Table 20 to 22 provides an insightful look at how gesture performance deteriorates under varying degrees of layer-specific ablations, particularly highlighting the effects of 25% to 75% ablation. It is evident from the table that low-level features are impacted by these ablations. Layer 1 consistently shows the most significant decline in performance for the 25% and 50% ablation level, illustrating that many gestures suffer greater losses relative to the original model's outcomes when this layer's functionality is reduced. However, for the 75% ablation level its observable that layer 2 is the most affected layer which once again shows the importance of the low-level features since layers 4 and 5 are very small or non-affected by the ablations.

Table 20: 25% Single-Layer Ablation results for all convolutional layers for cluster 1. Marked as red are the gestures which results were at least suffer a 25% loss compared to the original model results.

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	61	90	85	64	89
Gesture 2	117	66	80	109	102	132
Gesture 3	59	46	90	65	47	55
Gesture 4	80	63	76	64	75	82
Gesture 5	149	80	132	146	128	153
Gesture 6	135	112	152	148	112	127
Gesture 7	97	94	95	86	91	97
Gesture 8	144	125	153	151	132	147
Gesture 9	111	79	100	111	102	111
Gesture 10	134	147	94	112	149	125
Gesture 11	158	137	170	164	153	159
Gesture 12	159	156	134	147	164	170
Gesture 13	148	146	132	119	159	120
Gesture 14	154	165	148	147	164	147
Gesture 15	121	89	149	139	90	120
Gesture 16	114	84	91	101	121	106
Gesture 17	175	178	123	144	186	177

Table 21: 50% Single-Layer Ablation results for all convolutional layers for cluster 1. Marked as red are the gestures which results were at least suffer a 50% loss compared to the original model results.

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	92	57	69	52	101
Gesture 2	117	4	7	59	64	143
Gesture 3	59	0	122	101	53	53
Gesture 4	80	6	27	30	49	83
Gesture 5	149	54	50	137	97	156
Gesture 6	135	12	165	176	64	129
Gesture 7	97	63	59	84	83	82
Gesture 8	144	32	149	164	119	135
Gesture 9	111	75	51	110	112	117
Gesture 10	134	162	34	51	144	122
Gesture 11	158	68	171	165	120	165
Gesture 12	159	149	117	111	151	156
Gesture 13	148	104	80	70	172	119
Gesture 14	154	161	147	116	165	151
Gesture 15	121	5	160	135	46	109
Gesture 16	114	101	60	70	124	99
Gesture 17	175	146	56	66	188	161

Table 22: 75% Single Layer Ablation Results for All Convolutional Layers for Cluster 1

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	76	10	21	44	103
Gesture 2	117	11	0	6	48	128
Gesture 3	59	1	135	110	33	61
Gesture 4	80	9	1	10	6	85
Gesture 5	149	106	8	68	25	162
Gesture 6	135	57	164	184	8	126
Gesture 7	97	82	19	84	35	83
Gesture 8	144	71	114	153	74	131
Gesture 9	111	101	25	67	102	128
Gesture 10	134	160	2	6	144	122
Gesture 11	158	19	164	157	66	165
Gesture 12	159	102	47	57	160	163
Gesture 13	148	43	0	1	158	93
Gesture 14	154	131	101	19	158	141
Gesture 15	121	3	154	93	4	113
Gesture 16	114	39	13	9	103	66
Gesture 17	175	132	0	0	189	138

For Cluster 2, Table 23 to 25 presents a detailed analysis of how gesture performance is affected under various degrees of layer-specific ablations, with a focus on the impact from 25% to 75% ablation. Similar to Cluster 1, low-level features in Cluster 2 also show significant vulnerability to these ablations. Notably, Layer 2 is the most impacted layer in this cluster, particularly at the 75% ablation level, which highlights its critical role in processing foundational features.

Unlike Cluster 1, where Layer 1 exhibited the most pronounced declines at lower ablation

levels, Cluster 2 demonstrates that Layer 2 not only is the most impacted layer but also reveals a significant model bias when heavily ablated. This is evidenced by the dramatic performance drop at the 75% ablation level, indicating that this layer has reached a critical threshold where the model's ability to compensate is severely compromised.

This differential impact on Layer 2 in Cluster 2 compared to Cluster 1 suggests that while both clusters rely on early convolutional layers for feature extraction, the dependency patterns and their robustness against filter loss vary significantly.

Table 23: 25% Single-Layer Ablation results for all convolutional layers for cluster 2. Marked as red are the gestures which results were at least suffer a 25% loss compared to the original model results.

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	95	55	58	85	80
Gesture 2	117	115	113	93	98	106
Gesture 3	59	33	55	62	72	51
Gesture 4	80	79	91	76	75	75
Gesture 5	149	171	145	122	150	138
Gesture 6	135	111	93	91	147	132
Gesture 7	97	86	74	90	92	95
Gesture 8	144	86	130	126	153	145
Gesture 9	111	119	107	106	112	109
Gesture 10	134	129	103	138	113	136
Gesture 11	158	165	117	145	164	155
Gesture 12	159	147	126	162	150	163
Gesture 13	148	130	140	166	131	153
Gesture 14	154	154	149	161	158	161
Gesture 15	121	98	59	65	135	112
Gesture 16	114	94	70	124	104	115
Gesture 17	175	187	188	186	189	175

The resilience of certain gestures under high-level general ablations often relates back to their lesser dependency on the features processed in the initial layers or to compensatory mechanisms within these layers. Conversely, gestures heavily impacted by ablations in these initial layers during single-layer studies tend to show significant performance drops in general ablations once a critical threshold of filter loss is reached. This demonstrates the network's capacity for compensatory processing, where other parts of the network can enhance their processing

Table 24: 50% Single-Layer Ablation results for all convolutional layers for cluster 2. Marked as red are the gestures which results were at least suffer a 50% loss compared to the original model results.

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	95	0	40	94	68
Gesture 2	117	37	35	29	88	100
Gesture 3	59	35	24	61	85	42
Gesture 4	80	71	26	34	65	61
Gesture 5	149	164	35	44	153	137
Gesture 6	135	139	7	9	167	121
Gesture 7	97	27	30	56	83	93
Gesture 8	144	57	74	86	152	131
Gesture 9	111	117	70	105	123	100
Gesture 10	134	36	0	107	81	117
Gesture 11	158	180	9	91	164	140
Gesture 12	159	34	0	155	124	172
Gesture 13	148	68	13	165	111	151
Gesture 14	154	98	2	19	134	165
Gesture 15	121	96	24	11	133	106
Gesture 16	114	12	0	112	75	129
Gesture 17	175	188	189	189	149	174

Table 25: 75% Single-Layer Ablation results for all convolutional layers for cluster 2. Marked as red are the gestures which results were at least suffer a 75% loss compared to the original model results.

Gesture	Original Model	Layer 1	Layer 2	Layer 3	Layer 4	Layer 5
Gesture 1	83	62	0	23	56	67
Gesture 2	117	5	0	28	27	93
Gesture 3	59	6	0	27	127	29
Gesture 4	80	1	0	2	34	47
Gesture 5	149	123	0	0	108	115
Gesture 6	135	144	0	0	183	110
Gesture 7	97	5	0	1	88	85
Gesture 8	144	9	0	21	163	119
Gesture 9	111	116	0	87	118	95
Gesture 10	134	24	0	40	30	106
Gesture 11	158	152	0	22	165	122
Gesture 12	159	32	0	96	87	175
Gesture 13	148	32	0	145	65	140
Gesture 14	154	39	0	130	95	156
Gesture 15	121	68	0	0	120	81
Gesture 16	114	1	0	84	40	128
Gesture 17	175	189	189	189	58	171

contributions to maintain the accuracy of resilient gestures. Certain gestures, such as Gesture 17 (rest) in Cluster 2, display unusual resilience in maintaining accuracy even during severe ablations, suggesting a model bias where it defaults to predicting Gesture 17 (rest) when it cannot adequately process other information due to filter loss.

Conversely, Gestures 5 (TLFE), 13 (FP), and 16 (HC) in Cluster 1 are significantly impacted by ablations in the initial layers, emphasizing their dependency on the complex features processed primarily within these layers. The detailed analysis of gesture impacts across different layers and clusters reveals a complex interplay, where the ablation of specific layers leads to varied effects on gesture predictability, highlighting the nuanced roles these layers play within the network architecture.

This comprehensive analysis provides critical insights into the resilience and vulnerability of different network layers and their specific roles in gesture classification. Understanding these dynamics is pivotal for optimizing neural network designs, particularly for prosthetic control systems, to enhance fault tolerance and operational efficiency.

5.3 Model Recovery

Following a comprehensive analysis of all ablation scenarios, the focus shifted towards the recovery of selected models. This phase is crucial for evaluating the potential to restore network functionality and for assessing the resilience of neural networks against significant losses of filters. Models for recovery were chosen from the general ablation study rather than single-layer ablations. This strategic decision was driven by the need to understand the network's overall robustness and to assess recovery strategies across scenarios that impact all convolutional layers broadly, thereby providing more comprehensive and applicable insights into network operations.

The 25% general ablation level was selected because it marks a critical threshold where performance noticeably deteriorates, as detailed in the results (Figures 60 and 61). This scenario strikes a balance between moderate and severe impairment, making it an ideal test case to evaluate the network's capacity to recover from significant yet manageable filter loss. Conversely, the 75% general ablation was chosen to explore the limits of network recovery. This severe level of ablation serves to investigate whether there is a recovery threshold beyond which the model cannot effectively regain its original performance. Additionally, it facilitates an understanding of the recovery process nuances compared to less severe ablations, including variations in the number of training epochs needed and the effectiveness of different recovery techniques.

Recovery efforts focused on two primary models: one from the 25% ablation scenario and another from the 75% scenario, covering both clusters. This approach aimed to determine if significant differences in recovery outcomes exist between moderately and severely ablated networks.

The models were retrained using the same learning rate as in the original mode, integrating

regularization techniques like early stopping to prevent overfitting. This careful tuning is crucial to ensure that the recovery does not introduce new biases or errors.

The effectiveness of these recovery efforts was evaluated using a suite of performance metrics, accuracy, precision, recall, F1 score, and test dataset prediction. These metrics helped not only to quantify the extent of recovery but also to compare the performance between the models recovered from the 25% and 75% ablation scenarios.

In recovering the model from the 75% ablation, the study sought to identify whether a non-recoverable threshold exists and, if recovery was achievable, to analyze the training intensity required. This included comparing the number of epochs necessary to restore functionality in models subjected to moderate versus severe ablations.

The recovery of the model subjected to a 25% general ablation in Cluster 1 demonstrates not only a successful restoration of network functionality but also the efficiency of the recovery process. The model's training, executed over a total of 11 epochs, using an early stopping mechanism with a patience of three, highlights the strategic approach taken to prevent overfitting while ensuring optimal recovery performance. Model recovery plot over the total epochs in Appendix C (Figure C.141).

Table 26: Model comparisons between the original model, the effect of 25% general ablation Cluster 1, and the recovery model.

Model	Prediction	Accuracy	Precision	Recall	F1 Score
Original Model	2138	0.8888	0.8904	0.8882	0.8884
25% General Ablation	1601	0.6112	0.6855	0.6106	0.6106
Recovery 25% General Ablation	2135	0.8826	0.8847	0.8822	0.8822

From the data in Table 26, it is evident that the recovered model closely approximates the performance of the original model. The recovered model achieves an accuracy of 88,26%, a remarkable improvement from the ablated model's accuracy of 61,12% and being inside the deviation of the original model's accuracy of 88,88%. This recovery indicates not only the effectiveness of the recovery process but also the resilience of the network, as it is able to regain high levels of accuracy and other performance metrics post-ablation.

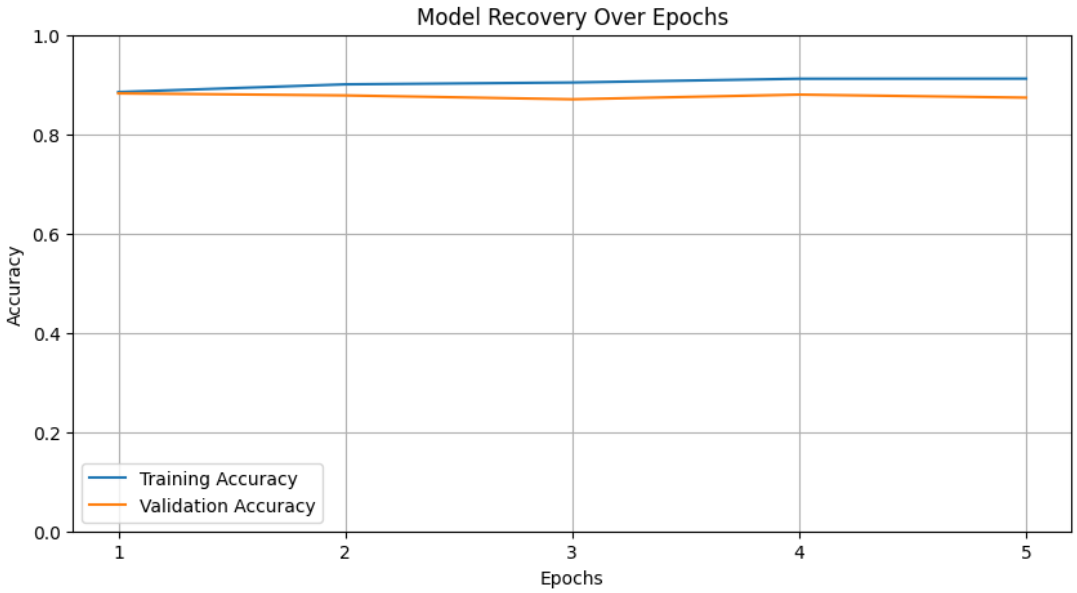


Figure 104: Model recovery 25% general abaltion Cluster 1 over epoch of the first 5 epochs.

Figure 104 illustrates the training and validation accuracy of the recovered model across the five first epochs, showcasing the effectiveness of the recovery process and the stability of the model post-recovery. Notably, both training and validation accuracies rapidly converge to a high level of accuracy from the first epoch, indicating that the model quickly regained a significant portion of its pre-ablation performance. This rapid convergence suggests that the recovery interventions were highly effective, enabling the model to restore its performance almost immediately.

The graph also demonstrates that the accuracies are consistently maintained around 90% throughout the recovery process, underscoring the robustness of the model and the success of the recovery strategy. The close alignment between training and validation accuracies throughout the process suggests that the model is not experiencing overfitting, which is crucial in recovery scenarios to ensure that the model generalizes well rather than merely memorizing specific training examples.

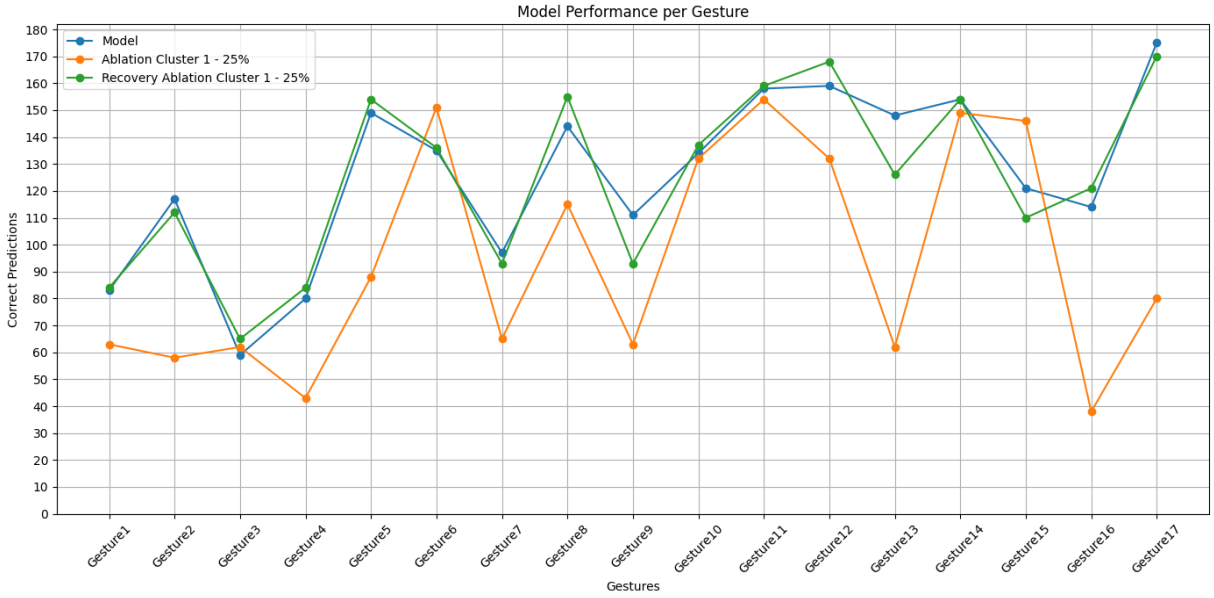


Figure 105: Model performance per gesture between the original mode, the 25% general ablation Cluster 1 model, and the recovery of the 25% general ablation Cluster 1model.

In addition to evaluating recovery metrics, a comparative analysis of gesture prediction accuracy across the original, ablated, and recovered models was undertaken. This analysis is critical as it underscores the practical implications of the recovery in terms of the model's operational effectiveness. It highlights how accurately each model predicts various gestures, illustrating the nuanced improvements achieved through the recovery process.

Table 27: Model comparisons between the original model, the effect of 25% general ablation Cluster 2, and the recovery model.

Model	Prediction	Accuracy	Precision	Recall	F1 Score
Original Model	2138	0.8888	0.8904	0.8882	0.8884
25% General Ablation	942	0.4094	0.6295	0.4141	0.4225
Recovery 25% General Ablation	2129	0.8991	0.8982	0.8992	0.8981

The recovery of the model subjected to a 25% general ablation in Cluster 2 demonstrates a successful restoration of network functionality, like in Cluster 1. The model's training, executed over a total of 13 epochs using an early stopping mechanism with a patience of three, highlights the strategic approach taken to prevent overfitting while ensuring optimal recovery performance. Model recovery plot over the total epochs in Appendix C (Figure C.142). The comparison of performance metrics before and after the recovery process is presented in Table 27, present that the recovered model closely approximates the performance of the original model. The recovered model achieves an accuracy of 89,91%, a remarkable improvement from the ablated model's accuracy of 40.94% and being inside the deviation of the original model's accuracy.

This recovery indicates not only the effectiveness of the recovery process but also the resilience of the network, as it is able to regain high levels of accuracy and other performance metrics post-ablation.

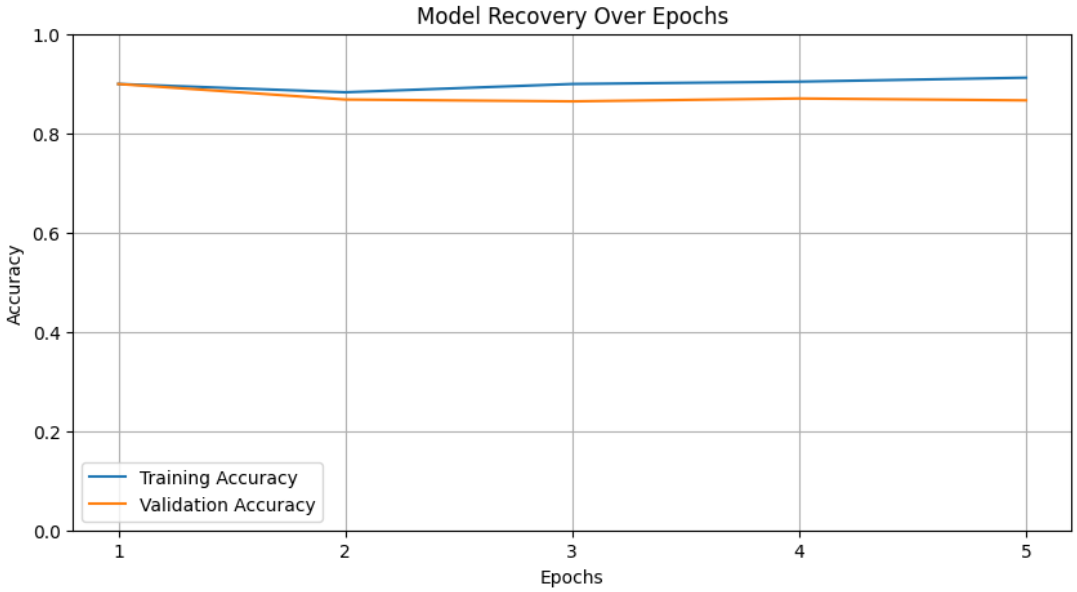


Figure 106: Model recovery 25% general abaltion Cluster 2 over epoch of the first 5 epochs.

Figure 106 illustrates the training and validation accuracy of the recovered model across the five first epochs, showcasing the effectiveness of the recovery process and the stability of the model post-recovery. Notably, both training and validation accuracies rapidly converge to a high level of accuracy from the first epoch, indicating that the model quickly regained a significant portion of its pre-ablation performance. The graph also demonstrates that the accuracies are consistently maintained around 90% throughout the recovery process, underscoring the robustness of the model and the success of the recovery strategy. The close alignment between training and validation accuracies throughout the process suggests that the model is not experiencing overfitting, which is crucial in recovery scenarios to ensure that the model generalizes well rather than merely memorizing specific training examples.

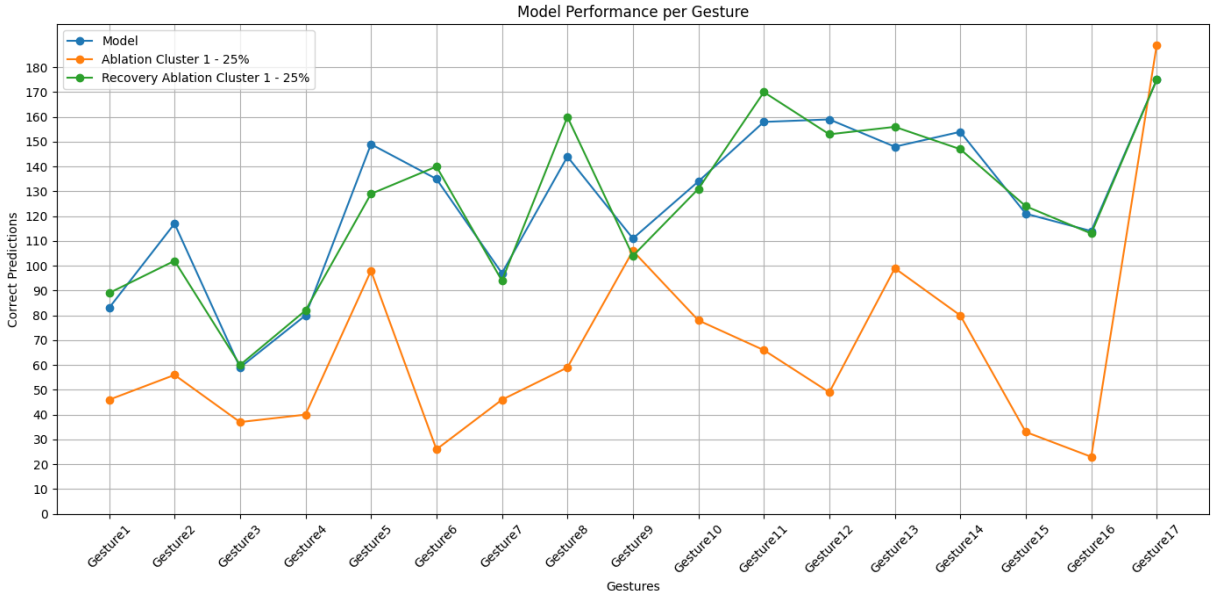


Figure 107: Model performance per gesture between the original mode, the 25% general ablation Cluster 2 model, and the recovery of the 25% general ablation Cluster 2 model.

The recovery of the models subjected to a 25% general ablation in both Cluster 1 and Cluster 2 showcases a successful restoration of network functionality, as detailed in Tables 26 and 27. These tables illustrate that both clusters not only achieved remarkable recovery in terms of accuracy but also displayed significant similarities in precision, recall, and F1 scores compared to the original model.

Initially, the ablated models exhibited considerable differences in performance, Cluster 1 had a drop in post-ablation accuracy to 61.12%, while Cluster 2 experienced a more drastic reduction, with accuracy plummeting to 40.94%. However, post-recovery, both models demonstrated a remarkable convergence in performance metrics, with Cluster 1 achieving an accuracy of 88,26% and Cluster 2 reaching a closely comparable accuracy of 89.91%.

This convergence is visually supported by Figures 104 and 106, where both clusters show rapid convergence of training and validation accuracies to around 90% from the first epoch of the recovery process. The graphs indicate that despite the initial differences in the severity of the impact from the ablation, the recovery interventions enabled both models to not only regain their functionality but also to align closely in terms of overall performance.

The uniform recovery across different gestures, depicted in Figure 104 and 106, underscores an interesting aspect of the neural network adaptive capabilities. Even though the two clusters started with varied levels of accuracy and gesture recognition performance post-ablation, the recovery process standardized the outcomes across both clusters.

The ability of both clusters to not only recover but also to converge in their performance metrics post-recovery provides a compelling case for the resilience of the network design. It implies that the network, when subjected to uniform recovery protocols, can mitigate initial disparities caused by localized impairments. For applications where consistency in performance across different modules or clusters of a network is crucial—such as in adaptive prosthetics

or advanced interactive systems—these findings are particularly valuable. They suggest that recovery techniques can be effectively standardized across different parts of a network to achieve uniform functionality.

Interestingly, while the recovery of Cluster 1 required only 11 epochs, as noted in Figure C.141, Cluster 2 needed slightly more, totaling 13 epochs, as depicted in Figure C.142 to complete its recovery process. This difference in the number of epochs required for recovery suggests variations in the learning efficiency or architectural complexities between the two clusters. Cluster 1's quicker recovery could indicate a more streamlined or less complex network structure that facilitates faster adaptation and restoration of functionalities.

Encouraged by the successful recovery of both clusters from a 25% general ablation, which highlighted the resilience of the network design, the research was extended to examine recovery from a 75% general ablation. This decision was influenced by the need to explore the limits of the network's recovery capabilities and to determine if similar resilience could be observed under more severe conditions. Investigating recovery at this level allows for a deeper understanding of the network's robustness and the effectiveness of the recovery protocols across a broader range of ablation severities.

This progression to more extreme ablation scenarios aims to answer critical questions about the network's ability to maintain functionality under significant stress. It will assess whether the recovery interventions that proved effective at lower levels of ablation can still facilitate recovery when a larger proportion of the network is compromised.

This phase of the research not only deepens the inquiry into neural network resilience but also enhances the practical application of these findings, ensuring that recovery techniques can be optimized and standardized for varying degrees of network impairment.

Table 28: Model comparisons between the original model, the effect of 75% general ablation Cluster 1, and the recovery model.

Model Description	Prediction	Accuracy	Precision	Recall	F1 Score
Original Model	2138	0.8888	0.8904	0.8882	0.8884
75% General Ablation	579	0.1895	0.1674	0.1902	0.1211
Recovery 75% General Ablation	2102	0.8711	0.8742	0.8697	0.8707

The recovery process of the model subjected to a severe 75% general ablation in Cluster 1 exemplifies a significant restoration of network functionality. Utilizing the same early stopping mechanism previously applied in the 25% recovery scenario, the model was trained over 24 epochs. This approach ensured that the training ceased at an optimal point, preventing overfitting while maximizing recovery performance. Model recovery plot over the total epochs in Appendix C (Figure C.143).

Table 28 presents a comparison of performance metrics before and after the recovery process. The recovered model achieved an accuracy of 87.11%, which is a remarkable improvement from the ablated model's accuracy of 18.95%. This level of accuracy post-recovery not only surpasses expectations given the extent of ablation but also falls within the deviation range of the original model's performance, indicating an effective return to functional norms.

This extraordinary recovery, especially when considering the severe initial reduction in accuracy due to the ablation, underscores the robustness of the recovery protocols and the adaptability of the network architecture. Achieving such high accuracy post-recovery highlights the network's capacity to reconstitute essential functionalities even after significant disruption.

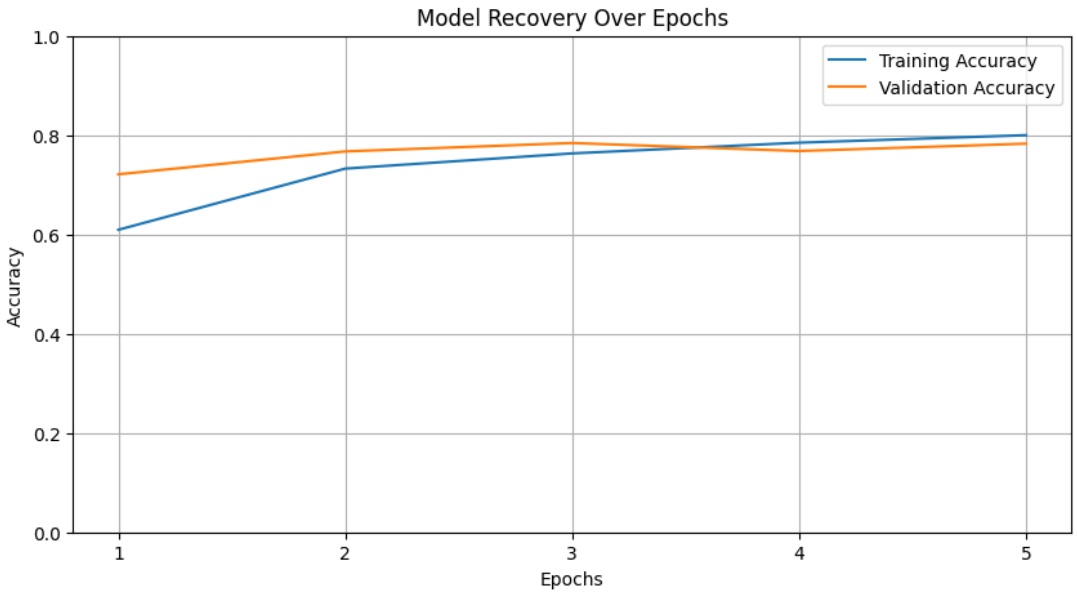


Figure 108: Model recovery 75% general abaltion Cluster 1 over epoch of the first 5 epochs.

Figure 108 illustrates the training and validation accuracy of the recovered model from a 75% general ablation in Cluster 1, captured over the initial five epochs. The graph reveals an impressive yet not instantaneous recovery trajectory, starting from around 60%, both training and validation accuracies swiftly rise to approximately 80% by the end of the fifth epoch. This rapid increase in accuracy indicates that the model quickly adapted and regained a significant portion of its pre-ablation performance.

It is notable that the accuracies achieve a stable growth around 80% during these early stages of recovery, demonstrating the robustness of the model and the effectiveness of the recovery strategy. While the graph only represents the first five epochs, it is important to note that the recovery process extended over a total of 24 epochs, as depicted in Figure C.143. By the conclusion of the full training period, the model's accuracy further improved to 87.11%, as indicated in Table 28. The continued improvement without overfitting, facilitated by the early stopping mechanism, highlights the model's capacity to enhance its performance progressively while maintaining generalization across additional epochs.

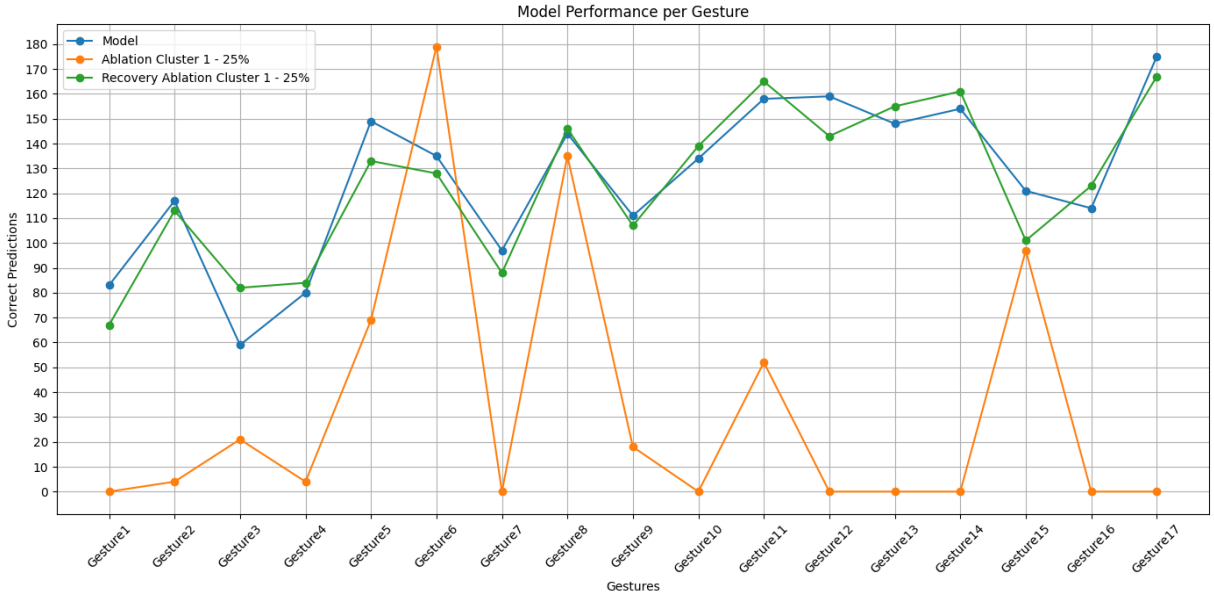


Figure 109: Model performance per gesture between the original mode, the 75% general ablation Cluster 1 model, and the recovery of the 75% general ablation Cluster 1model.

The recovered model, represented by the green line in Figure 109, demonstrated notable improvements in accuracy across most gestures when compared to the ablated model. Impressively, in several cases, the accuracy of the recovered model not only matched but also surpassed that of the original model.

The recovery process of the model in Cluster 2, after a severe 75% general ablation, exemplifies significant restoration of network functionality. The model was trained over 22 epochs, using the same early stopping mechanism that proved effective in the 25% recovery scenario. This strategy was crucial in preventing overfitting while ensuring optimal recovery of the model's capabilities.

Table 29: Model comparisons between the original model, the effect of 75% general ablation Cluster 2, and the recovery model.

Model Description	Prediction	Accuracy	Precision	Recall	F1 Score
Original Model	2138	0.8888	0.8904	0.8882	0.8884
75% General Ablation	189	0.0610	0.0036	0.0588	0.0068
Recovery 75% General Ablation	2135	0.8686	0.8700	0.8682	0.8676

Table 29 presents a remarkable transformation in model performance from the ablation to the recovery phase. Initially, the ablated model's accuracy stood at a mere 6.1%, dramatically demonstrating the severity of the impairment. Post-recovery, the model achieved an accuracy of 86.86%, a substantial improvement, although narrowly missing the original model's accuracy

range by only 0.0018. Despite this slight deviation, the near-complete restoration to functional norms is notable and reflects the efficacy of the recovery interventions.

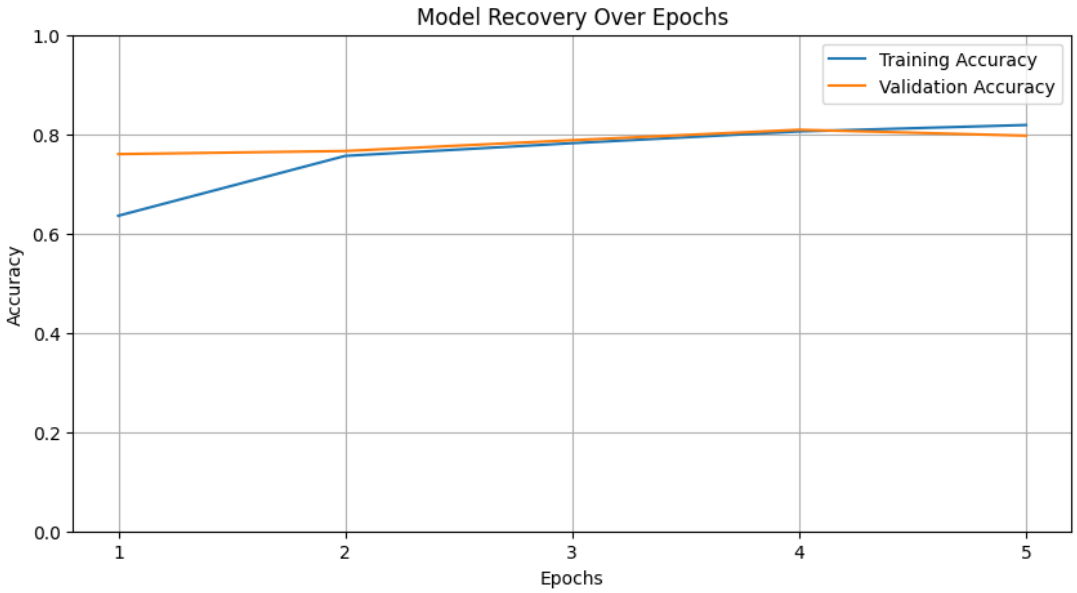


Figure 110: Model recovery 75% general abaltion Cluster 2 over epoch of the first 5 epochs.

Figure 110 illustrates the training and validation accuracy trajectories during the initial five epochs of the recovered model. Starting from around 60%, the training accuracy quickly escalated to approximately 80% by the fifth epoch. This sharp rise indicates rapid adaptation of the model, effectively regaining a significant portion of its pre-ablation performance.

The model's performance stabilized around 80% during these early stages, underscoring the robustness of the recovery strategy. While this graph captures only the initial epochs, it is crucial to note that the recovery extended across a total of 22 epochs, as depicted in Figure C.144. By the end of this period, the model's accuracy further improved to 86.86%, affirming the progressive and sustained enhancement of its capabilities. Model recovery plot over the total epochs in Appendix C (Figure C.144).

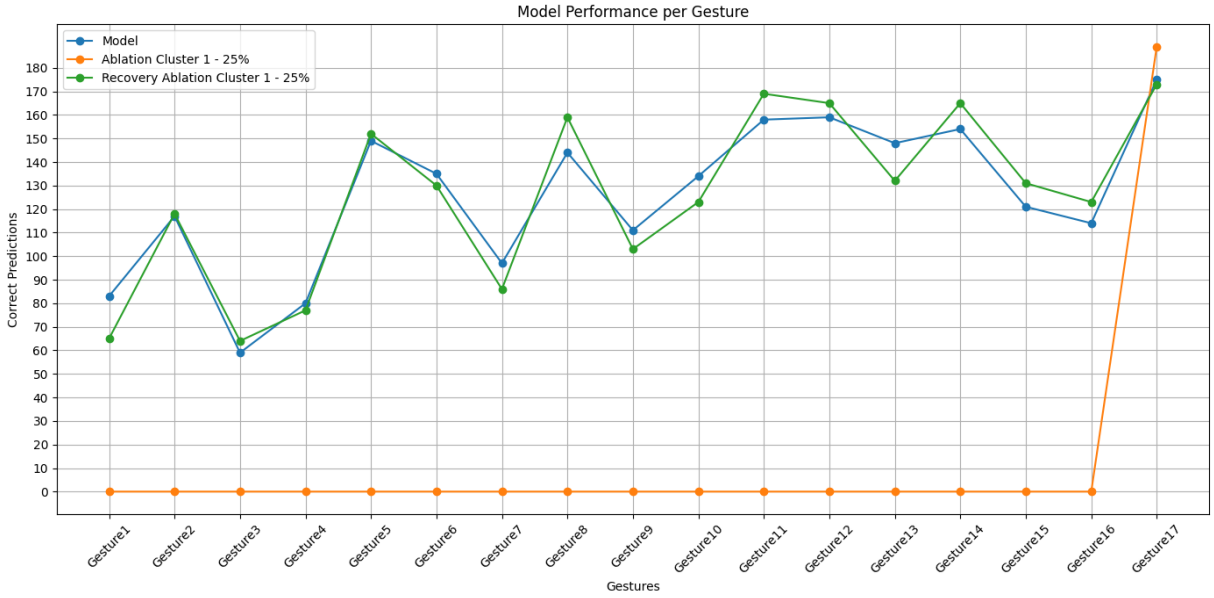


Figure 111: Model performance per gesture between the original mode, the 75% general ablation Cluster 2 model, and the recovery of the 75% general ablation Cluster 2 model.

Significant improvements in gesture prediction accuracy were observed, with the recovered model, depicted by the green line in Figure 111, showing substantial accuracy gains across most gestures compared to the severely ablated state. Notably, the pre-ablation model had rendered the network effectively unusable, defaulting every prediction to Gesture 17 (rest). The successful recovery to an accuracy so close to the original model, despite slightly exceeding the deviation threshold by 0.0018, is particularly impressive. This demonstrates the model's capability to not only restore but nearly replicate its initial performance, highlighting the resilience and adaptability of the neural network architecture. The recovery process following a severe 75% general ablation in both Cluster 1 and Cluster 2 highlights the remarkable resilience and adaptability of the network. Both clusters demonstrated significant restoration of network functionality, as evidenced by Tables 28 and 29, which detail the performance metrics before and after recovery. This recovery showcases not only a return to functional norms but also a substantial alignment with the original model's performance.

Initially, both models experienced a drastic reduction in performance metrics due to the ablation—Cluster 1's accuracy fell to 18.95%, and Cluster 2's plummeted even further to 6.1%. Post-recovery, however, Cluster 1 achieved an accuracy of 87.11%, and Cluster 2 closely followed with 86.86%. Such marked improvements underscore the efficacy of the recovery interventions and the robustness of the network's architecture.

The recovery trajectory for both clusters is visually represented in Figures 106 and 110, which illustrate the rapid convergence of training and validation accuracies to around 80% within the first five epochs. The continued improvement across the total of 24 and 22 epochs for Cluster 1 and Cluster 2, respectively, culminated in accuracies that nearly matched those of the original model, highlighting the network's ability to progressively enhance its performance while maintaining generalization without overfitting.

Figure 107 and 111 offer crucial insights into the recovery's impact on gesture prediction accuracy for both clusters. These figures illustrate significant improvements in most gestures, with the recovered models often matching or even surpassing the original model's performance. This exceptional recovery underscores the network's capability to not only restore but also enhance its functionality after severe disruptions. The recovery was particularly noteworthy given that prior to interventions, the network's predictive capabilities were nearly unusable, defaulting most predictions to a single gesture.

However, while the overall recovery trajectory was positive, it was uneven across different gestures, highlighting variability in the network's ability to adapt post-disruption. For Cluster 1, gestures such as 1 (LP), 5 (TLFE), 6 (TIFE), and 15 (HO) demonstrated less robust recovery, showing worse predictions compared to the original model. Similarly, in Cluster 2, gestures 1 (LP), 9 (IFE), 10 (TE), and 13 (FP) encountered more significant difficulties in achieving accurate predictions.

This variance can be attributed to the intrinsic complexity and distinct feature requirements of each gesture, which may not be equally well-represented or compensated for during the recovery process.

The recovery processes from 25% and 75% general not only demonstrate the resilience and adaptability of the underlying architecture but also highlight its capability to restore functionality significantly after substantial disruptions. Recovery outcomes detailed in Tables 26, 27, 28, and 29 illustrate remarkable improvements in accuracy, precision, recall, and F1 scores across all scenarios, aligning closely with the performance metrics of the original model.

For the 25% ablation, Cluster 1 initially saw accuracy drop to 61.12%, and Cluster 2 to 40.94%. Post-recovery, however, there was a noticeable convergence in performance, with Cluster 1 achieving an accuracy of 88.26% and Cluster 2 reaching 89.91%. This rapid recovery is visually supported by Figures 104 and 106, where both training and validation accuracies quickly approached 90%, suggesting that despite the initial severity of the ablation, the network's inherent robustness facilitated a swift and effective restoration of its capabilities.

The 75% ablation scenario presented more severe challenges, with initial accuracies plummeting to 18.95% for Cluster 1 and 6.1% for Cluster 2. Remarkably, both clusters managed to recover, achieving accuracies of 87.11% and 86.86%, respectively. Figures 108 and 110 depict this impressive recovery trajectory, where accuracies climbed to around 80% within just the first five epochs, demonstrating the network's ability to adapt and recover swiftly from extreme conditions.

Despite these successes, differences in gesture prediction capabilities were noted. As depicted in Figures 109 and 111, while most gestures saw significant improvements, some gestures in both clusters showed less robust recovery, potentially indicating more complex or unique feature dependencies. Notably, the number of epochs required for recovery varied between the ablation scenarios, with the 25% ablation requiring fewer epochs (11 and 13 for Clusters 1 and 2, respectively) compared to the 75% ablation (24 and 22 epochs for Clusters 1 and 2, respectively).

This variation suggests that the severity of the ablation impacts both the duration and complexity of the recovery process.

5.4 Visualization

This section of the thesis presents the results obtained from the Unity-based visualization method, designed to offer a more intuitive and engaging demonstration of the gesture recognition model's capabilities. Unity Engine, known for its robust and flexible development environment, enabled the creation of a virtual interface that simulates real-time gesture recognition. This setup not only showcases the model's performance but also highlights potential applications, such as in adaptive prosthetics, by providing immediate visual feedback on gesture prediction accuracy.

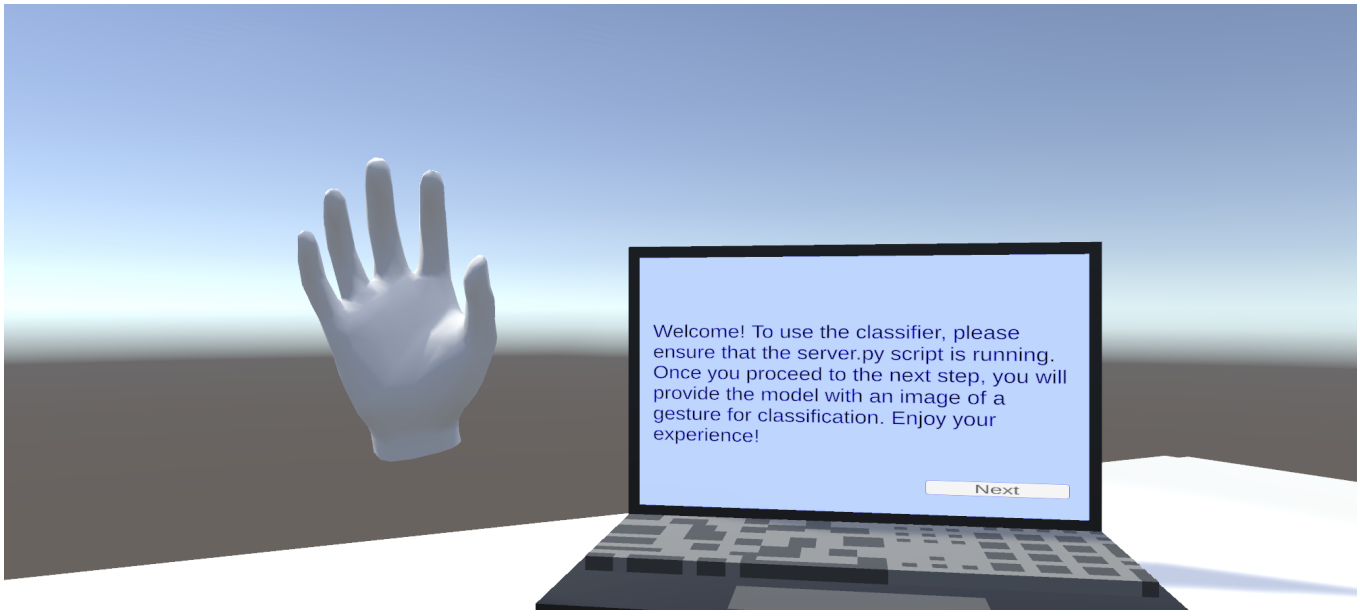


Figure 112: Unity project initial environment.

Figure 112 illustrates the initial environment presented to the user, including a brief introduction and a reminder to activate the server. Users must modify the server code to update the image paths in the dictionary, ensuring that the system can access the correct files. To assist in this setup, a secondary Python script is provided, which generates the necessary dictionary format from a specified folder containing the images. Once configured, users can proceed by clicking the "Next" button.

Figure 113: Unity project input field.

Figure 113 demonstrates the functionality of the input field where users enter the name of the gesture they wish to test. This is detailed further in Figure 114, which shows the filled input field. Upon submission, the Unity application communicates with the local Flask server, which retrieves the corresponding file path from the dictionary, processes the image through the model, and returns the classification result and its true label to Unity.

Figure 114: Unity project input field filled.

Figures 115 and 116 depicts the response within the Unity environment following the reception of the model's output. Depending on whether the predicted gesture matches the true label, the system triggers different visual cues.

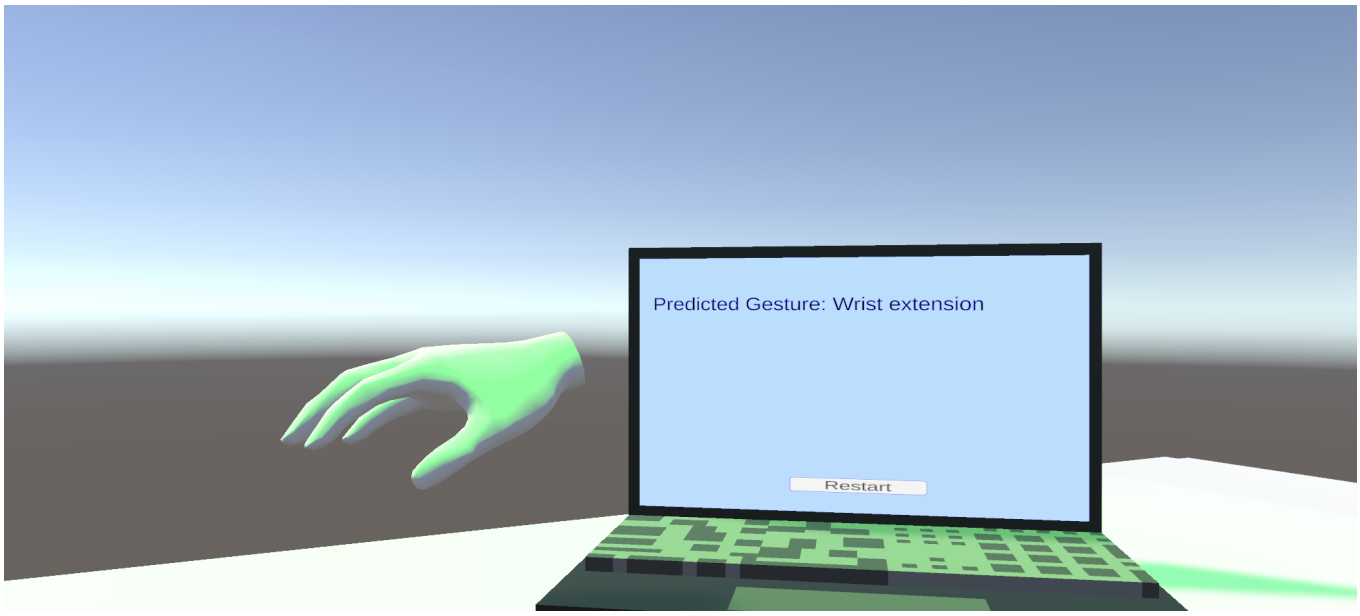


Figure 115: Unity project correct prediction.

In Figure 115 represents what happens when the prediction is correct, a green light is activated, symbolizing accurate gesture recognition. The name of the predicted gesture is displayed both on the laptop screen and alongside the corresponding hand animation.

Figure 116: Unity project incorrect predcition.

In instances where the prediction does not match the true label, as shown in Figure 116, a red light appears. The screen displays both the predicted gesture and the true label to provide clear and immediate feedback to the user. This interactive setup not only verifies the model's performance in real-time but also illustrates how this technology could be seamlessly integrated into applications requiring dynamic user interactions, such as adaptive prosthetics.

Chapter 6

Conclusion

The primary goal of this thesis was to explore and enhance the application of surface Electromyography (sEMG) and neural networks in developing intuitive and adaptive prosthetic limbs, with a specific focus on hand and wrist gesture recognition. This exploration was underpinned by a rigorous application of Convolutional Neural Networks (CNNs) with an AlexNet architecture, chosen for its potential in feature extraction and gesture classification.

The literature reviewed provided a robust foundation for the methodologies applied. By synthesizing insights from past studies, particularly those involving sEMG and neural network applications in prosthetics, this thesis could adopt and adapt advancements in computational techniques to the specific challenges of gesture recognition. The state-of-the-art review highlighted the evolving nature of neural network applications in biomedical engineering, which lead to the selection of AlexNet architecture due to its balance of depth and computational efficiency, making it suitable for real-time applications in prosthetic control. A supplementary table of the main elements of the state-of-the-art can be found in Table A1.

Utilizing the GRABMyo dataset, which comprises EMG data from 43 participants performing various hand and wrist gestures, was pivotal for the success of this research. The efficacy of any machine learning model, especially in the domain of prosthetic control, depends on the quality of the dataset employed for training and evaluation. Opting for the GRABMyo dataset, known for its stability and reliability, ensured that the insights and advancements derived from this study are robust and applicable in real-world scenarios. The ablation methodology, inspired by Meyes et al., 2019, was adapted to better suit the specific nuances of the experiments. These modifications were instrumental in yielding deeper insights into the CNN model's functionality and efficiency. Dissecting the results of these single-layer and general ablations will highlight how such methodologies contribute to a comprehensive understanding of neural network architectures. Meyes et al., 2019 utilized the VGG-19 architecture, which comprises 19 layers with learnable weights, including 16 convolutional layers. Their study focused on performing ablations specifically in the two layers deemed most crucial for their classification tasks. In contrast, this study employed a CNN with an AlexNet architecture, which consists of only 5 convolutional layers. This difference in architecture necessitated a distinct approach to selecting

and evaluating layers for ablation. Instead of limiting the study to just a few layers, ablations were performed across all five convolutional layers, providing a comprehensive view of each layer's impact on the network's overall functionality.

In Meyes et al., 2019 filters were grouped based on the similarity of their weights, calculated via the absolute Euclidean distance of normalized filter weights. This study advances the methodology by employing PCA followed by K-means Clustering to group filters. This approach not only normalized the filter weights by their Euclidean norm but also reduced the dimensionality of the data to two principal components, facilitating a more nuanced analysis of filter relationships and their contributions to classification tasks. The systematic grouping into two distinct clusters allowed for a targeted exploration of each cluster's role within the network.

While Meyes et al., 2019 limited ablations to proportions of 1%, 5%, 10%, and 25%, this study extended the ablation to 50% and 75% for each cluster of filters. This extension was critical in pushing the model to its limits and testing the resilience of the network under more extreme conditions. The broader range of ablations provided deeper insights into the network's robustness and the criticality of various features components under stress. Another major distinction lies in the approach to recovery. Meyes et al., 2019 focused on the recovery of specific layers after a 25% ablation, using five epochs of subsequent training. In contrast, this study implemented recovery strategies not only for a 25% and 75% general ablation, assessing the network's ability to recover from more severe disruptions. This comprehensive recovery analysis offered valuable insights into the neural network's adaptability and the effectiveness of recovery protocols at different levels of network impairment. The ablation studies conducted in this thesis were meticulously structured into two distinct types: general ablation and single-layer ablation. In the general ablation approach, a uniform percentage of filters was removed from each convolutional layer across the entire model. This method provided insights into the overall resilience of the network and the interdependencies between layers. Conversely, the single-layer ablation targeted only one convolutional layer at a time, leaving the others untouched. This approach allowed for a precise identification of the critical layers and filters that significantly influence the model's accuracy and performance. Together, these ablation strategies enhanced our understanding of the neural network's architecture and its functionality in the context of gesture recognition for prosthetic control. The studies revealed that early convolutional layers, particularly Layers 1 and 2, play a critical role in processing primary visual features such as edges, contours, and basic shapes, as can be observed in the feature's maps depicted in Figure 47 to 51. These features are foundational for all subsequent layers, setting the stage for effective gesture recognition. The resilience of the network was evident when minor percentages of filters (1% to 10%) were deactivated across all layers, as depicted from Table 6 to 19, suggesting a built-in redundancy that mitigates minor losses without significant degradation in performance. However, as deactivation levels increased to 50% and 75%, significant performance declines were observed, especially affecting critical gestures. For instance, Table 6 shows a sharp decline from an initial accuracy of 88.88% to just 18.95% at 75% filter loss in general ablation for Cluster 1, and this performance

degradation was particularly pronounced in gestures heavily reliant on the foundational features processed by the early layers, as detailed in Figure64. When Layers 1 and 2 were specifically targeted in single-layer ablations, gestures such as Gesture 1 (LP) and Gesture 10 (TE) showed significant vulnerability, with marked decreases in accuracy. This dependency showcases the layers' crucial nature in maintaining model accuracy across a variety of gesture recognitions. When these key layers are heavily ablated, other layers do not fully compensate for the loss, leading to model biases and a propensity to default predicting simpler gestures like Gesture 17 (rest) under severe ablation conditions. Conversely, gestures such as Gesture 5 (TLFE), 13 (FP), and 16 (HC) display significant vulnerabilities when initial layers are compromised, emphasizing their reliance on the complex features processed predominantly within these layers. This nuanced interplay between layer functionality and gesture predictability underscores the intricate architecture of the network and its critical components. The implications of these ablation studies are profound, indicating that while some redundancy is built into the network, the essential processing capabilities of early layers cannot be completely compensated by other layers. This understanding is vital for optimizing neural network architectures for prosthetic control, where precision in gesture recognition is crucial. Following the ablation studies, the recovery phase of the models, subjected to a 25% general ablation across both Cluster 1 and Cluster 2 showed impressive recovery, demonstrating the network's ability to standardize performance across different modules, which is crucial for ensuring consistent operation of prosthetic devices. This was conclusively detailed through comparative analysis of accuracy, precision, recall, and F1 scores, which not only approached but, in some instances, exceeded the performance metrics of the original model. The recovery process highlights the network's robust architecture, capable of compensating for significant disruptions in functionality.

Initially, the ablated models experienced a notable drop in performance, with Cluster 1 reducing to 61.12% accuracy and Cluster 2 more dramatically to 40.94%. However, following the implementation of recovery protocols, both clusters exhibited a convergence in performance, with Cluster 1 reaching an accuracy of 88.26% and Cluster 2 an impressive 89.91%. This uniform recovery across clusters not only underscores the adaptive capabilities of the neural network but also emphasizes the effectiveness of the standardized recovery interventions.

Visual representations, such as Figures 104 and 106, supported these findings by illustrating a rapid convergence of training and validation accuracies to around 90% from the onset of the recovery process. These figures emphasize that despite initial variances in the impact of the ablation, the recovery strategies enabled both models to not only regain their original functionalities but also to standardize outcomes across different gestures, as depicted in Figures 105 and 107.

The understanding of the model features importance can lead to prosthetic limbs that are more sensitive and accurate, closely mimicking natural motion and thus significantly enhancing the user experience. Furthermore, the demonstrated ability of the network to recover from significant disruptions ensures that prosthetic limbs can maintain functionality in various environments and

under stress conditions, increasing their reliability and safety for users.

Insights into the network's layer-specific roles allow for the development of customizable control systems tailored to individual users needs and specific gestures. This adaptability is crucial for personalizing prosthetic functionality, catering to a wide range of activities from basic day-to-day tasks to more complex actions. Additionally, the recovery findings suggest that prosthetic systems could include real-time monitoring and adjustment features, ensuring optimal performance by continuously adapting to changes in the user's muscular activities and external conditions.

This thesis has demonstrated the critical role of combining sEMG and neural network architectures, particularly CNNs with an AlexNet architecture, in enhancing the intuitiveness and adaptability of prosthetic limbs. Through rigorous experimentation using the GRABMyo dataset and comprehensive ablation studies, this research has provided significant insights into the neural network's capacity for gesture recognition and the robustness of its architectural design. The successful recovery of model functionality after significant ablation underscores the resilience of the employed methods and highlights the potential for these techniques in real-world applications. Encouraged by the success from a 25% general ablation recovery, the study progressed to explore the network's resilience under more severe conditions—specifically a 75% general ablation. Both clusters demonstrated significant recovery post-severe ablation, with accuracies nearly matching those of the original model, showcasing the network's capability to handle and adapt to extreme operational stress.

Future Work

Building upon the findings and methodologies developed in this thesis, several avenues for future research are proposed to enhance the capabilities and applicability of EMG-based prosthetic control systems. These suggestions aim to push the boundaries of current technology, creating more responsive and intuitive prosthetic solutions.

Future studies should focus on improving the CNN model's ability to predict gestures from very short segments of EMG signals, ideally within the first milliseconds of muscle activation. This development would drastically reduce response times, enabling nearly seamless prosthetic control. Researchers could explore adaptive algorithms that adjust classifications in real-time as more data becomes available, optimizing both speed and accuracy.

It is essential to conduct advanced ablation studies that assess the impact of network components over varied time frames. These studies would help refine the network by identifying which features are critical at different stages of muscle activation. The goal would be to streamline the CNN architecture, removing non-essential filters to enhance efficiency without sacrificing accuracy.

Investigating models that adapt dynamically to ongoing muscle activity presents a significant opportunity. Such models would adjust their parameters in real-time, learning from the continuous stream of EMG data. This approach could lead to more precise and reliable prosthetics that adjust their behavior based on the user's muscle activity patterns.

Applying these advanced predictive models in practical prosthetic designs is a critical next step. Future work could involve integrating enhanced models into user-friendly prosthetic devices and conducting extensive testing to evaluate their practical performance and user satisfaction in everyday activities.

The techniques developed for EMG signal classification have potential applications beyond prosthetics. Future research could explore other possibilities, potentially leading to innovations that transcend the initial scope of this study.

By pursuing these research directions, future studies can significantly advance the field of prosthetic technology, ensuring that EMG-based systems not only meet the current needs of users but also anticipate and adapt to future challenges. This progressive approach aims to foster technological advancements that enhance the quality of life for individuals relying on prosthetic devices.

Bibliography

Alkan, A., & Günay, M. (2012). Identification of EMG signals using discriminant analysis and svm classifier. *Expert Systems with Applications*, *39*(1), 44–47.

Arts, L. P., & van den Broek, E. L. (2022). The fast continuous wavelet transformation (fCWT) for real-time, high-quality, noise-resistant time–frequency analysis. *Nature Computational Science*, *2*(1), 47–58.

Azhiri, R. B., Esmaeili, M., & Nourani, M. (2021). Real-time EMG signal classification via recurrent neural networks. *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, 2628–2635.

Bau, J., Wang, Y. L., & Wang, W. (2001). Reduction of skin impedance by the improvement of the blood circulation. *2001 Conference Proceedings of the 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, *3*, 3081–3082.

Belkhou, A., Achmamad, A., & Jbari, A. (2019). Classification and diagnosis of myopathy EMG signals using the continuous wavelet transform. *2019 Scientific Meeting on Electrical-Electronics & Biomedical Engineering and Computer Science (EBBT)*, 1–4.

Bu, D., Guo, S., & Li, H. (2022). Semg-based motion recognition of upper limb rehabilitation using the improved YOLO-v4 algorithm. *Life*, *12*(1), 64.

Carreiras, C., Alves, A. P., Lourenço, A., Canento, F., Silva, H., Fred, A., et al. (2015). BioSPPy: Biosignal processing in Python. https://github.com/PIA-Group/BioSPPy/

Chowdhury, R. H., Reaz, M. B., Ali, M. A. B. M., Bakar, A. A., Chellappan, K., & Chang, T. G. (2013). Surface electromyography signal processing and classification techniques. *Sensors*, *13*(9), 12431–12466.

Chun-Lin, L. (2010). A tutorial of the wavelet transform. *NTUEE, Taiwan*, *21*(22), 2.

Da Silva, H. P., Guerreiro, J., Lourenço, A., Fred, A., & Martins, R. (2014). BITalino: A novel hardware framework for physiological computing. *International Conference on Physiological Computing Systems*, *2*, 246–253.

Daube, J. R., & Rubin, D. I. (2009). Needle electromyography. *Muscle & Nerve: Official Journal of the American Association of Electrodiagnostic Medicine*, *39*(2), 244–270.

Guerreiro, J., Martins, R., Silva, H., Lourenço, A., & Fred, A. L. (2013). BITalino-a multimodal platform for physiological computing. *ICINCO (1)*, 500–506.

Hariri, W. (2022). Efficient masked face recognition method during the covid-19 pandemic. *Signal, image and video processing*, *16*(3), 605–612.

Hemingway, M. A., Biedermann, H.-J., & Inglis, J. (1995). Electromyographic recordings of paraspinal muscles: Variations related to subcutaneous tissue thickness. *Biofeedback and Self-regulation*, *20*, 39–49.

Kuiken, T. (2003). Consideration of nerve-muscle grafts to improve the control of artificial arms. *Technology and Disability*, *15*(2), 105–111.

Kuiken, T. A., Li, G., Lock, B. A., Lipschutz, R. D., Miller, L. A., Stubblefield, K. A., & Englehart, K. B. (2009). Targeted muscle reinnervation for real-time myoelectric control of multifunction artificial arms. *Jama*, *301*(6), 619–628.

Li, S., Zhang, Y., Tang, Y., Li, W., Sun, W., & Yu, H. (2023). Real-time semg pattern recognition of multiple-mode movements for artificial limbs based on cnn-rnn algorithm. *Electronics*, *12*(11), 2444.

Lolure, A., & Thool, V. (2015). Wavelet transform based EMG feature extraction and evaluation using scatter graphs. *2015 International Conference on Industrial Instrumentation and Control (ICIC)*, 1273–1277.

Lyu, Z., Pan, L., Huang, Q., & Qu, W. (2022). Pattern recognition of sEMG recordings on arm muscles among different training motions. *Journal of Physics: Conference Series*, *2386*(1), 012091.

Meyes, R., Lu, M., de Puiseau, C. W., & Meisen, T. (2019). Ablation studies in artificial neural networks. *arXiv preprint arXiv:1901.08644*.

Mezzarane, R. A., & Kohn, A. F. (2009). A method to estimate EMG crosstalk between two muscles based on the silent period following an H-reflex. *Medical Engineering and Physics*, *31*, 1331–1336. https://doi.org/10.1016/j.medengphy.2009.09.005

Ngeo, J. G., Tamei, T., & Shibata, T. (2014). Continuous and simultaneous estimation of finger kinematics using inputs from an EMG-to-muscle activation model. *Journal of neuro-engineering and rehabilitation*, *11*, 1–14.

Pradhan, A., Jiang, N., & He, J. (2022). Gesture recognition and biometrics electromyogram (GRABMyo) dataset. *IEEE Dataport*.

Röhrle, O., Yavuz, U. Ş., Klotz, T., Negro, F., & Heidlauf, T. (2019). Multiscale modeling of the neuromuscular system: Coupling neurophysiology and skeletal muscle mechanics. *Wiley Interdisciplinary Reviews: Systems Biology and Medicine*, *11*(6), e1457.

Rupom, F. F., Jannat, S., Tamanna, F. F., Al Johan, G. M., & Islam, M. M. (2020). EMG controlled bionic robotic arm using artificial intelligence and machine learning. *2020 IEEE Region 10 Symposium (TENSYMP)*, 334–339.

Salminger, S., Stino, H., Pichler, L. H., Gstoettner, C., Sturma, A., Mayer, J. A., Szivak, M., & Aszmann, O. C. (2022). Current rates of prosthetic usage in upper-limb amputees–have innovations had an impact on device acceptance? *Disability and Rehabilitation*, *44*(14), 3708–3713.

Sobahi, N. (2011). Denoising of EMG signals based on wavelet transform. *Asian Transactions on Engineering*, *1*(5), 17–23.

Stanek, K., Barnhart, N., & Zhu, Y. (2018). Control of a robotic prosthetic hand using an EMG signal based counter. *ASME International Mechanical Engineering Congress and Exposition*, *52033*, V04AT06A008.

Tuncer, S. A., & Alkan, A. (2022). Classification of EMG signals taken from arm with hybrid CNN-SVM architecture. *Concurrency and Computation: Practice and Experience*, *34*(5), e6746.

Zhang, X., Wang, Y., & Han, R. P. (2010). Wavelet transform theory and its application in emg signal processing. *2010 Seventh International Conference on Fuzzy Systems and Knowledge Discovery*, *5*, 2234–2238.

Appendix A

Table A1: Supplementary table overview of main elements in the state-of-the-art

Elements	**Function/Detail**	**Significance in EMG/Prosthetics**	**Key Citations**
Neuromuscular System	Integral for the coordination and execution of movement, involving neurons, muscle fibers, and neuromuscular junctions.	Essential for translating neural signals into muscle actions.	Röhrle et al., 2019
Skeletal Muscle Structure	Structured in a hierarchy: endomysium, perimysium, and epimysium surround and support muscle fibers.	Provides a framework that influences force dispersion and muscle integrity during movement.	Röhrle et al., 2019
Action Potential (AP)	Neuron's AP triggers neurotransmitter release at neuromuscular junctions, initiating muscle contraction.	The fundamental process for muscle activation, crucial for EMG signal interpretation in prosthetics.	Röhrle et al., 2019

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Upper Limb Muscular Architecture	Focuses on muscle groups that control movements in the wrist and hand, important for precise hand and finger movements.	Targeted in prosthetic applications to enable complex and refined motor tasks.	Bu et al., 2022; Ngeo et al., 2014
Radial Nerve	Controls extension of the wrist and fingers, essential for coordinated hand movements.	Data from muscles innervated by the Radial Nerve are vital for prosthetic designs that mimic natural movements.	Ngeo et al., 2014
Ulnar Nerve	Governs intricate movements of the hand, affecting grip strength and fine motor skills of smaller fingers.	Critical for prosthetic designs requiring detailed control over fine finger movements.	Ngeo et al., 2014
Median Nerve	Affects thumb and fingers, crucial for gripping and performing most hand grips.	Involvement in controlling muscles for dexterity and broad range of grips in prosthetic applications.	Ngeo et al., 2014

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Intramuscular Electrodes	Thin needle-like electrodes placed directly into the muscle tissue to record signals from deep muscles or individual motor units. Highly specific but invasive.	Used for capturing precise muscle activity patterns essential for advanced prosthetic control systems.	T. A. Kuiken et al., 2009
Subdermal Electrodes	Electrodes inserted just beneath the skin, not as deep as intramuscular, to capture EMG activity with reduced invasiveness. Offers a balance between signal quality and patient comfort.	Suitable for semi-long-term monitoring with less invasiveness, useful in rehabilitation and prosthetic feedback systems.	Daube and Rubin, 2009
Surface Electrodes	Non-invasive electrodes placed on the skin over muscle groups to record EMG signals. Easiest to apply and suitable for long-term monitoring.	Most common in prosthetic control due to non-invasiveness, although prone to external disturbances and noise.	Chowdhury et al., 2013
Motion Interference	Noise generated by the relative movement of muscle, skin, and electrode, similar in frequency to genuine EMG signals.	Important to minimize for accurate prosthetic control and muscle movement analysis.	Chowdhury et al., 2013; Bau et al., 2001

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Ambient Interference	External disturbances from electrical and magnetic sources that may overlay EMG readings.	Critical challenge in maintaining EMG signal integrity, especially in electronically dense environments.	Chowdhury et al., 2013
Crosstalk Interference	Misleading signals from adjacent muscles not targeted by the electrodes.	Strategies to minimize crosstalk are essential for precise muscle activity mapping in diagnostics and prosthetic controls.	Chowdhury et al., 2013; Mezzarane and Kohn, 2009; Hemingway et al., 1995
EMG Ideal Frequency	The most valuable EMG signal information typically lies between 10 – 500 Hz, with the 50 – 150 Hz range being particularly significant for capturing muscle activity characteristics.	Frequency analysis helps isolate the muscular activity signals from noise, which is pivotal in designing effective and responsive prosthetic systems.	Zhang et al., 2010

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Wavelet Transform (WT)	Utilized for time-frequency analysis, WT is particularly adept at capturing non-stationary characteristics of EMG signals, decomposing them into different frequency bands for analysis.	WT's ability to detail signal components at various scales enhances prosthetic control systems by providing more precise signal interpretation.	Zhang et al., 2010; Lolure and Thool, 2015; Chowdhury et al., 2013
Discrete Wavelet Transform (DWT)	Processes signals quickly through downsampling, reducing data volume at each decomposition level, ideal for analyzing non-stationary signals like EMG for quick and efficient processing.	DWT's fast processing capabilities make it suitable for real-time prosthetic applications where quick signal interpretation is crucial.	Zhang et al., 2010; Sobahi, 2011; Chowdhury et al., 2013
Continuous Wavelet Transform (CWT)	Provides a detailed and flexible representation of signals across all scales, ideal for applications requiring high precision and granularity in the time-frequency domain.	The high-resolution output of CWT is invaluable for advanced diagnostics and research in muscle dynamics and prosthetic control.	Chun-Lin, 2010; Arts and van den Broek, 2022; Belkhou et al., 2019

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
BITalino EMG Sensor	A compact, versatile biosignal acquisition system popular in educational and commercial settings for its affordability and ease of use. Integrates sensors for EMG and other physiological signals.	Ideal for prototyping and educational purposes, providing accessible and reliable EMG data acquisition for research and development in prosthetics.	Da Silva et al., 2014
Board Configuration	The default setup where the BITalino is used as-is, with integrated sensors connected to a single board for straightforward use and data collection.	Enables rapid deployment and testing of sensor capabilities, crucial for real-time observation of muscle activities in prosthetic control.	Da Silva et al., 2014
Plugged Configuration	Allows disconnection of sensor blocks from the main board, using plugs to enable flexible combinations of sensors as per user requirements.	Facilitates customization of sensor arrays for targeted EMG data acquisition, enhancing the adaptability of prosthetic devices.	Da Silva et al., 2014

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Freestyle Configuration	Offers maximum customization by detaching all individual blocks from the main BITalino board, enabling users to rearrange or mix and match components freely.	Supports innovative EMG sensor applications and integration into complex prosthetic systems or experimental setups.	Da Silva et al., 2014
K-Nearest Neighbors (KNN)	A simple algorithm that classifies new data points based on the majority class of 'k' nearest points, often used for its efficiency in various applications including EMG.	Suitable for real-time applications due to its simplicity and effectiveness, especially in systems with a clear proximity of data points in feature space.	Rupom et al., 2020; Lyu et al., 2022
Support Vector Machines (SVM)	Uses hyperplanes to classify data into different groups, effective in creating a robust decision boundary between different movement types.	Highly valued for its ability to provide precise and robust classifications, crucial for the accuracy of prosthetic controls.	Lyu et al., 2022; Alkan and Günay, 2012; Rupom et al., 2020

Table A1: (continuation)

Elements	Function/Detail	Significance in EMG/Prosthetics	Key Citations
Convolutional Neural Networks (CNN)	Specialized deep learning models that process data with grid-like topology, applying convolution operations to capture spatial and temporal patterns.	Enables detailed feature extraction from EMG signals, improving the functionality and responsiveness of prosthetic devices.	Tuncer and Alkan, 2022; Li et al., 2023
Recurrent Neural Networks (RNNs)	Excels in handling sequential data, adapting their internal state to the sequence of inputs, useful for time-dependent signal processing.	Ideal for EMG systems where understanding the temporal sequence of muscle activity is crucial for accurate motion detection.	Azhiri et al., 2021; Li et al., 2023
Hybrid CNN-SVM Models	Combines the feature extraction capabilities of CNNs with the precise classification boundaries of SVMs.	Enhances classification accuracy and reliability by leveraging the strengths of both CNN and SVM, useful in complex prosthetic control scenarios.	Rupom et al., 2020
Hybrid CNN-RNN Models	Integrates CNN for feature extraction with RNN for managing sequential dependencies, providing a comprehensive approach to pattern recognition.	Offers a robust solution for classifying multi-channel EMG signals, especially effective in adaptive prosthetics for real-time applications.	Tuncer and Alkan, 2022; Li et al., 2023

Appendix B

Table B.2: Detailed Model Performances Across Various Gestures and Ablations

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Original model		83	117	59	80	149	135	97	144	111	134	158	159	148	154	121	114	175
Ablation 1% - 5 conv	C1	80	114	53	78	140	136	88	146	101	138	160	161	140	156	134	113	174
Ablation 5% - 5 conv	C1	75	112	39	72	135	133	80	147	85	139	156	161	107	159	144	90	175
Ablation 10% - 5 conv	C1	64	92	33	60	106	127	73	123	77	142	150	163	85	158	134	71	172

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 25% - 5 conv	C1	63	58	62	43	88	151	65	115	63	132	154	132	62	149	146	38	80
Ablation 50% - 5 conv	C1	68	53	27	15	117	134	60	95	94	117	131	110	0	148	145	65	0
Ablation 75% - 5 conv	C1	0	4	21	4	69	179	0	135	18	0	52	0	0	0	97	0	0
Ablation 1% - 5 conv	C2	79	114	59	80	153	130	91	146	108	133	158	159	154	158	120	115	175
Ablation 5% - 5 conv	C2	85	119	60	91	152	134	100	144	109	119	154	153	144	155	119	98	180
Ablation 10% - 5 conv	C2	83	106	55	76	160	128	93	139	108	130	155	159	156	162	93	109	183
Ablation 25% - 5 conv	C2	46	56	37	40	98	26	46	59	106	78	66	49	99	80	33	23	189
Ablation 50% - 5 conv	C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	189
Ablation 75% - 5 conv	C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	189

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 1% - conv 1	C1	83	118	60	81	152	134	97	144	111	134	158	158	148	155	121	114	176
Ablation 1% - conv 2	C1	83	120	57	82	152	134	95	143	110	134	158	161	149	154	119	114	176
Ablation 1% - conv 3	C1	82	115	59	80	146	136	96	146	111	133	158	159	148	154	122	113	175
Ablation 1% - conv 4	C1	83	119	60	81	150	134	98	144	110	135	158	159	148	154	121	114	176
Ablation 1% - conv 5	C1	83	117	61	79	148	135	97	146	108	135	158	162	147	154	121	113	175
Ablation 1% - conv 1	C2	83	117	59	80	149	135	97	144	111	134	158	159	148	154	121	114	175
Ablation 1% - conv 2	C2	81	111	58	78	142	140	94	147	109	132	158	155	149	158	122	116	174
Ablation 1% - conv 3	C2	83	111	58	78	147	137	94	146	112	134	157	158	150	156	120	116	175

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 1% - conv 4	C2	82	117	63	78	149	139	95	145	113	132	158	155	148	156	121	113	174
Ablation 1% - conv 5	C2	83	115	60	80	150	130	97	143	113	132	158	163	147	156	114	117	176
Ablation 5% - conv 1	C1	60	110	51	68	116	128	86	128	101	139	150	156	162	157	128	83	181
Ablation 5% - conv 2	C1	82	106	59	83	151	139	92	154	97	128	161	161	137	156	131	119	178
Ablation 5% - conv 3	C1	84	119	57	78	149	138	94	144	106	133	156	159	140	156	133	106	174
Ablation 5% - conv 4	C1	83	116	53	80	152	130	93	144	107	136	154	163	150	158	122	117	176
Ablation 5% - conv 5	C1	82	119	65	83	153	137	95	146	112	134	157	168	136	154	120	111	175
Ablation 5% - conv 1	C2	90	101	66	84	151	132	104	151	115	108	165	143	147	153	113	117	155

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 5% - conv 2	C2	91	121	55	90	143	135	103	142	116	130	156	152	126	150	118	102	178
Ablation 5% - conv 3	C2	81	115	49	78	142	129	90	145	106	141	155	165	153	154	116	113	176
Ablation 5% - conv 4	C2	81	113	68	78	152	138	93	147	110	130	157	150	155	155	124	112	174
Ablation 5% - conv 5	C2	83	116	57	79	146	129	92	145	111	130	159	158	150	158	117	118	176
Ablation 10% - conv 1	C1	65	118	62	63	160	147	65	154	115	87	159	143	139	149	123	76	188
Ablation 10% - conv 2	C1	80	109	78	84	151	140	102	153	117	120	153	165	148	155	118	108	176
Ablation 10% - conv 3	C1	83	121	60	75	149	141	91	147	108	121	160	153	135	155	132	106	171
Ablation 10% - conv 4	C1	78	116	50	72	148	126	95	141	105	140	155	169	150	159	118	119	180
Ablation 10% - conv 5	C1	88	124	61	89	150	134	92	144	115	134	157	168	110	153	120	106	178

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 10% - conv 1	C2	114	99	51	102	151	125	111	131	112	118	156	128	130	150	93	68	167
Ablation 10% - conv 2	C2	75	112	62	83	139	117	106	140	114	133	152	170	155	159	97	122	178
Ablation 10% - conv 3	C2	74	114	54	78	139	122	91	142	108	142	155	169	158	154	105	117	180
Ablation 10% - conv 4	C2	82	109	70	77	152	142	94	149	109	120	160	149	154	155	131	108	170
Ablation 10% - conv 5	C2	83	113	56	80	145	128	93	148	109	130	158	161	151	158	109	118	176
Ablation 25% - conv 1	C1	61	66	46	63	80	112	94	125	79	147	137	156	146	165	89	84	178
Ablation 25% - conv 2	C1	90	80	90	76	132	152	95	153	100	94	170	134	132	148	149	91	123
Ablation 25% - conv 3	C1	85	109	65	64	146	148	86	151	111	112	164	147	119	147	139	101	144

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 25% - conv 4	C1	64	102	47	75	128	112	91	132	102	149	153	164	159	164	90	121	186
Ablation 25% - conv 5	C1	89	132	55	82	153	127	97	147	111	125	159	170	120	147	120	106	177
Ablation 25% - conv 1	C2	95	115	33	79	171	111	86	86	119	129	165	147	130	154	98	94	187
Ablation 25% - conv 2	C2	55	113	55	91	145	93	74	130	107	103	117	126	140	149	59	70	188
Ablation 25% - conv 3	C2	58	93	62	76	122	91	90	126	106	138	145	162	166	161	65	124	186
Ablation 25% - conv 4	C2	85	98	72	75	150	147	92	153	112	113	164	150	131	155	135	104	152
Ablation 25% - conv 5	C2	80	106	51	75	138	132	95	145	109	136	155	163	153	161	112	115	175
Ablation 50% - conv 1	C1	92	4	0	6	54	12	63	32	75	162	68	149	104	161	5	101	146

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 50% - conv 2	C1	57	7	122	27	50	165	59	149	51	34	171	117	80	147	160	60	56
Ablation 50% - conv 3	C1	69	59	101	30	137	176	84	164	110	51	165	111	70	116	135	70	66
Ablation 50% - conv 4	C1	52	64	53	49	97	64	83	119	112	144	120	151	172	165	46	124	188
Ablation 50% - conv 5	C1	101	143	53	83	156	129	82	135	117	122	165	156	119	151	109	99	161
Ablation 50% - conv 1	C2	95	37	35	71	164	139	27	57	117	36	180	34	68	98	96	12	188
Ablation 50% - conv 2	C2	0	35	24	26	35	7	30	74	70	0	9	0	13	2	24	0	189
Ablation 50% - conv 3	C2	40	29	61	34	44	9	56	86	105	107	91	155	165	165	11	112	189
Ablation 50% - conv 4	C2	94	88	85	65	153	167	83	152	123	81	164	124	111	134	133	75	149
Ablation 50% - conv 5	C2	68	100	42	61	137	121	93	131	100	117	140	172	151	165	106	129	174

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 75% - conv 1	C1	76	11	1	9	106	57	82	71	101	160	19	102	43	131	3	39	132
Ablation 75% - conv 2	C1	10	0	135	1	8	164	19	114	25	2	164	47	0	101	154	13	0
Ablation 75% - conv 3	C1	21	6	110	10	68	184	84	153	67	6	157	57	1	19	93	9	0
Ablation 75% - conv 4	C1	44	48	33	6	25	8	35	74	102	128	66	160	158	158	4	103	189
Ablation 75% - conv 5	C1	103	128	61	85	162	126	83	131	128	113	165	163	93	141	113	66	138
Ablation 75% - conv 1	C2	62	5	6	1	123	144	5	9	116	24	152	32	32	39	68	1	189
Ablation 75% - conv 2	C2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	189
Ablation 75% - conv 3	C2	23	28	27	2	0	0	1	21	87	40	22	96	145	130	0	84	189

Continued on next page

Table B.2 – continued from previous page

Model	Cluster	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14	G15	G16	G17
Ablation 75% - conv 4	C2	56	27	127	34	108	183	88	163	118	30	165	87	65	95	120	40	58
Ablation 75% - conv 5	C2	67	93	29	47	115	110	85	119	95	106	122	175	140	156	81	128	171

Table B.3: Summary of Model Performances

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Original model		2138	0.3478	0.8888	0.8904	0.8882	0.8884
Ablation 1% - 5 conv	C1	1938	0.3609	0.8855	0.8867	0.8850	0.8848
Ablation 5% - 5 conv	C1	1834	0.4903	0.8381	0.8512	0.8371	0.8382
Ablation 10% - 5 conv	C1	1658	0.6879	0.7710	0.8014	0.7706	0.7705
Ablation 25% - 5 conv	C1	1521	1.1401	0.6112	0.6855	0.6106	0.6126

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 50% - 5 conv	C1	1379	1.7968	0.4699	0.5403	0.4764	0.4511
Ablation 75% - 5 conv	C1	579	2.6079	0.1895	0.1674	0.1902	0.1211
Ablation 1% - 5 conv	C2	1957	0.3548	0.8826	0.8840	0.8817	0.8820
Ablation 5% - 5 conv	C2	1936	0.4557	0.8447	0.8518	0.8443	0.8451
Ablation 10% - 5 conv	C2	1912	0.4813	0.8348	0.8373	0.8350	0.8335
Ablation 25% - 5 conv	C2	942	2.0640	0.4094	0.6295	0.4141	0.4225
Ablation 50% - 5 conv	C2	0	5.7445	0.0610	0.0036	0.0588	0.0068
Ablation 75% - 5 conv	C2	0	3.5965	0.0610	0.0036	0.0588	0.0068

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 1% - conv 1	C1	2144	0.3664	0.8830	0.8840	0.8820	0.8818
Ablation 1% - conv 2	C1	2141	0.3628	0.8847	0.8863	0.8841	0.8843
Ablation 1% - conv 3	C1	2133	0.3462	0.8909	0.8922	0.8905	0.8905
Ablation 1% - conv 4	C1	2144	0.3477	0.8892	0.8908	0.8886	0.8888
Ablation 1% - conv 5	C1	2139	0.3474	0.8888	0.8904	0.8882	0.8884
Ablation 1% - conv 1	C2	2138	0.3478	0.8888	0.8904	0.8882	0.8884
Ablation 1% - conv 2	C2	2124	0.3612	0.8855	0.8868	0.8846	0.8849
Ablation 1% - conv 3	C2	2132	0.3447	0.8880	0.8890	0.8875	0.8874
Ablation 1% - conv 4	C2	2138	0.3512	0.8880	0.8896	0.8875	0.8876

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 1% - conv 5	C2	2134	0.3490	0.8851	0.8864	0.8844	0.8844
Ablation 5% - conv 1	C1	2004	0.3981	0.8703	0.8731	0.8690	0.8695
Ablation 5% - conv 2	C1	2134	0.4054	0.8641	0.8673	0.8633	0.8633
Ablation 5% - conv 3	C1	2126	0.3684	0.8797	0.8815	0.8792	0.8792
Ablation 5% - conv 4	C1	2134	0.3519	0.8863	0.8876	0.8860	0.8859
Ablation 5% - conv 5	C1	2147	0.3583	0.8847	0.8861	0.8844	0.8844
Ablation 5% - conv 1	C2	2095	0.4397	0.8616	0.8687	0.8617	0.8618
Ablation 5% - conv 2	C2	2108	0.3904	0.8756	0.8780	0.8747	0.8755

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 5% - conv 3	C2	2108	0.3465	0.8892	0.8900	0.8886	0.8887
Ablation 5% - conv 4	C2	2137	0.3528	0.8847	0.8854	0.8844	0.8840
Ablation 5% - conv 5	C2	2124	0.3472	0.8843	0.8844	0.8836	0.8833
Ablation 10% - conv 1	C1	2013	0.6698	0.7834	0.7989	0.7819	0.7808
Ablation 10% - conv 2	C1	2157	0.4731	0.8488	0.8575	0.8478	0.8487
Ablation 10% - conv 3	C1	2108	0.3875	0.8740	0.8750	0.8740	0.8734
Ablation 10% - conv 4	C1	2121	0.3620	0.8843	0.8845	0.8840	0.8832
Ablation 10% - conv 5	C1	2123	0.3683	0.8793	0.8823	0.8795	0.8796

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 10% - conv 1	C2	2006	0.3790	0.8756	0.8783	0.8760	0.8754
Ablation 10% - conv 2	C2	2114	0.4514	0.8476	0.8495	0.8466	0.8471
Ablation 10% - conv 3	C2	2102	0.3700	0.8756	0.8766	0.8751	0.8751
Ablation 10% - conv 4	C2	2131	0.3749	0.8744	0.8761	0.8739	0.8732
Ablation 10% - conv 5	C2	2116	0.3546	0.8834	0.8844	0.8828	0.8827
Ablation 25% - conv 1	C1	1650	0.6950	0.7657	0.7895	0.7627	0.7619
Ablation 25% - conv 2	C1	1886	0.6951	0.7681	0.7930	0.7678	0.7682
Ablation 25% - conv 3	C1	1894	0.5284	0.8237	0.8294	0.8241	0.8233
Ablation 25% - conv 4	C1	1853	0.4701	0.8369	0.8423	0.8357	0.8356

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 25% - conv 5	C1	1940	0.3851	0.8756	0.8795	0.8758	0.8766
Ablation 25% - conv 1	C2	1812	0.7494	0.7681	0.7846	0.7711	0.7672
Ablation 25% - conv 2	C2	1627	1.0421	0.6960	0.7431	0.6950	0.7025
Ablation 25% - conv 3	C2	1785	0.5794	0.8007	0.8101	0.7985	0.7990
Ablation 25% - conv 4	C2	1936	0.4378	0.8488	0.8531	0.8487	0.8470
Ablation 25% - conv 5	C2	1926	0.3792	0.8740	0.8762	0.8744	0.8734
Ablation 50% - conv 1	C1	1088	2.0327	0.4848	0.6710	0.4863	0.4447
Ablation 50% - conv 2	C1	1456	1.7964	0.5379	0.6530	0.5366	0.5104

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 50% - conv 3	C1	1648	1.3181	0.6351	0.6877	0.6374	0.6233
Ablation 50% - conv 4	C1	1615	0.9227	0.7022	0.7307	0.7018	0.6948
Ablation 50% - conv 5	C1	1920	0.4742	0.8381	0.8489	0.8399	0.8404
Ablation 50% - conv 1	C2	1266	2.0497	0.4798	0.6364	0.4872	0.4424
Ablation 50% - conv 2	C2	349	5.4714	0.1845	0.5735	0.1866	0.1861
Ablation 50% - conv 3	C2	1270	1.6440	0.5605	0.6411	0.5590	0.5261
Ablation 50% - conv 4	C2	1832	0.7405	0.7496	0.7709	0.7505	0.7465
Ablation 50% - conv 5	C2	1833	0.4720	0.8295	0.8374	0.8298	0.8298

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 75% - conv 1	C1	1011	2.0476	0.4049	0.5257	0.4076	0.3713
Ablation 75% - conv 2	C1	957	3.6348	0.3122	0.4358	0.3112	0.2249
Ablation 75% - conv 3	C1	1045	2.7517	0.3933	0.5372	0.3960	0.3282
Ablation 75% - conv 4	C1	1152	1.8673	0.4930	0.6128	0.4934	0.4433
Ablation 75% - conv 5	C1	1861	0.6597	0.7685	0.7889	0.7720	0.7691
Ablation 75% - conv 1	C2	819	2.8795	0.3369	0.5407	0.3467	0.2722
Ablation 75% - conv 2	C2	0	14.4978	0.0610	0.0036	0.0588	0.0068
Ablation 75% - conv 3	C2	706	3.2279	0.3402	0.3427	0.3410	0.2699
Ablation 75% - conv 4	C2	1506	1.5277	0.5758	0.6693	0.5779	0.5595

Table B.3 continued from previous page

Model	Cluster	Total	Loss	Accuracy	Precision	Recall	F1_Score
Ablation 75% - conv 5	C2	1668	0.6737	0.7615	0.7787	0.7626	0.7616

Appendix C

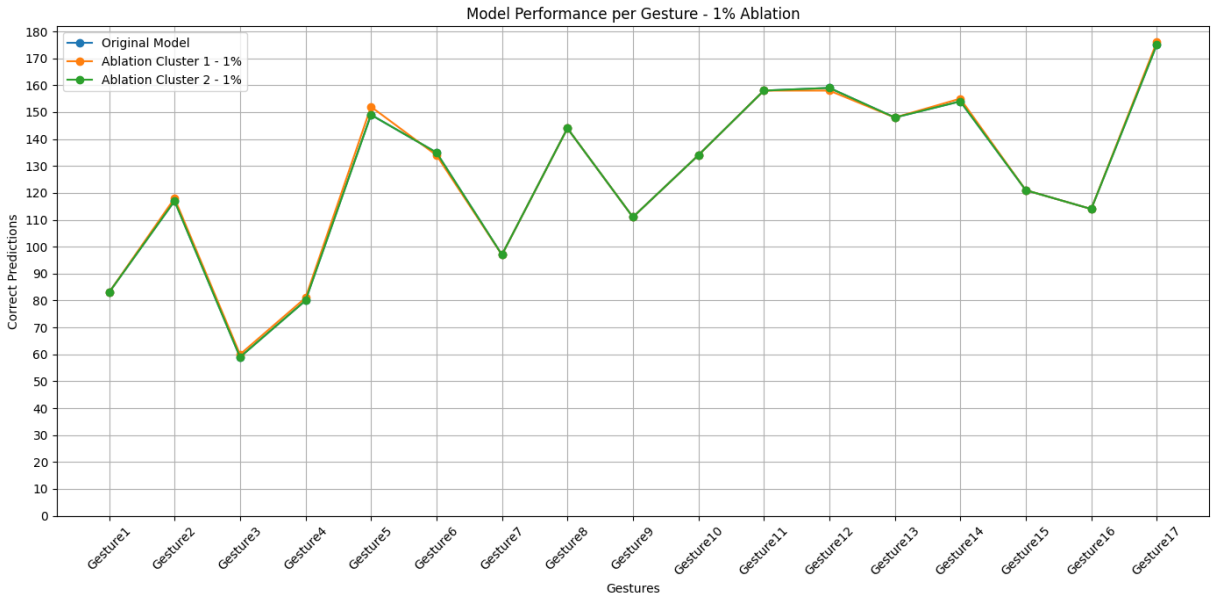


Figure C.117: Model Performance per gesture, comparison between original model, and the single layer ablation of 1% for both Clusters of the first convolutional layer.

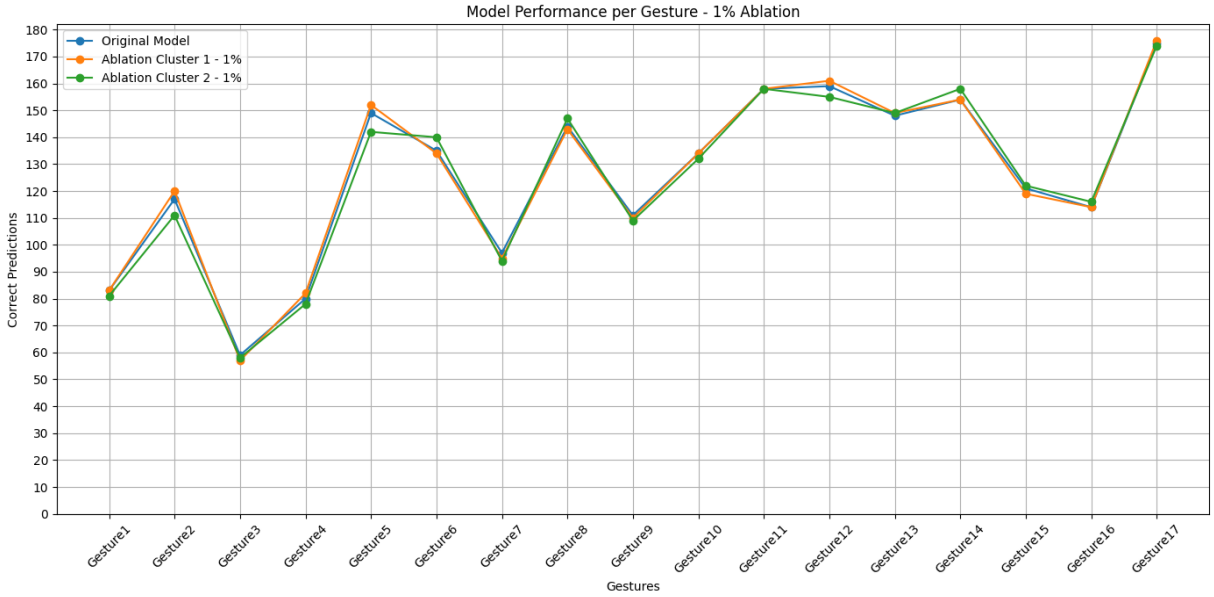


Figure C.118: Model Performance per gesture, comparison between original model, and the single layer ablation of 1% for both Clusters of the second convolutional layer.

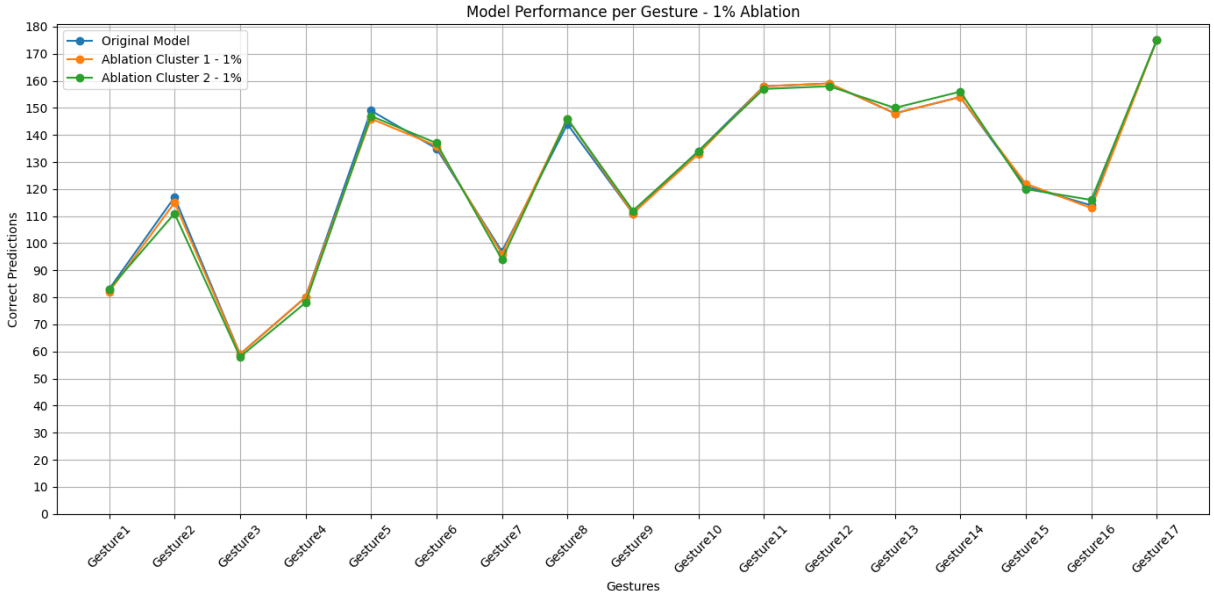


Figure C.119: Model Performance per gesture, comparison between original model, and the single layer ablation of 1% for both Clusters of the third convolutional layer.

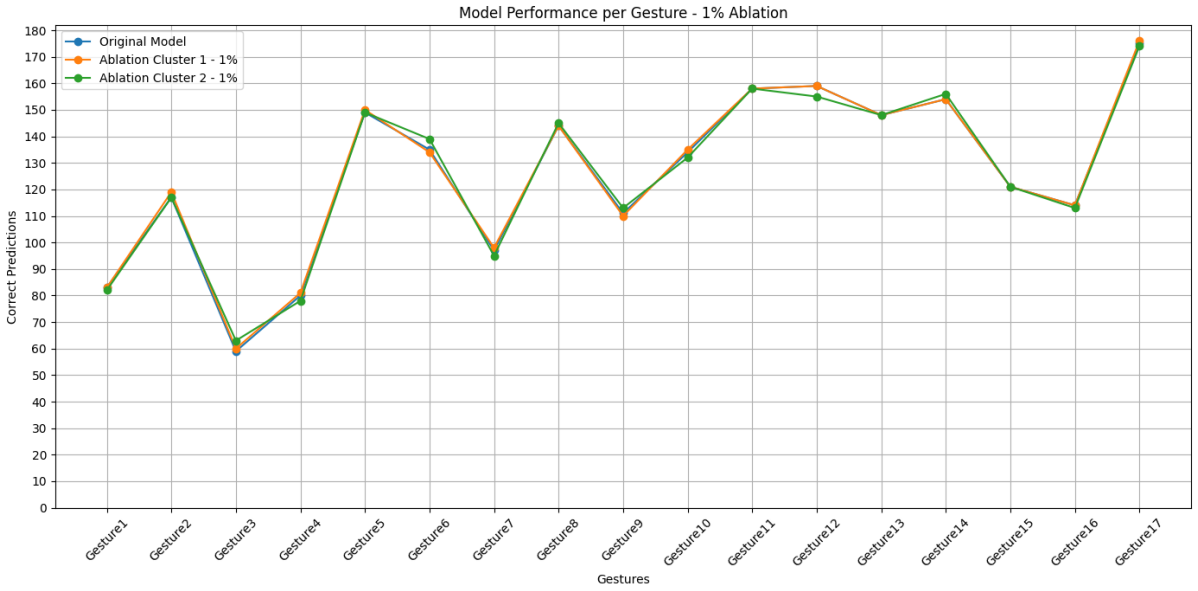


Figure C.120: Model Performance per gesture, comparison between original model, and the single layer ablation of 1% for both Clusters of the fourth convolutional layer.

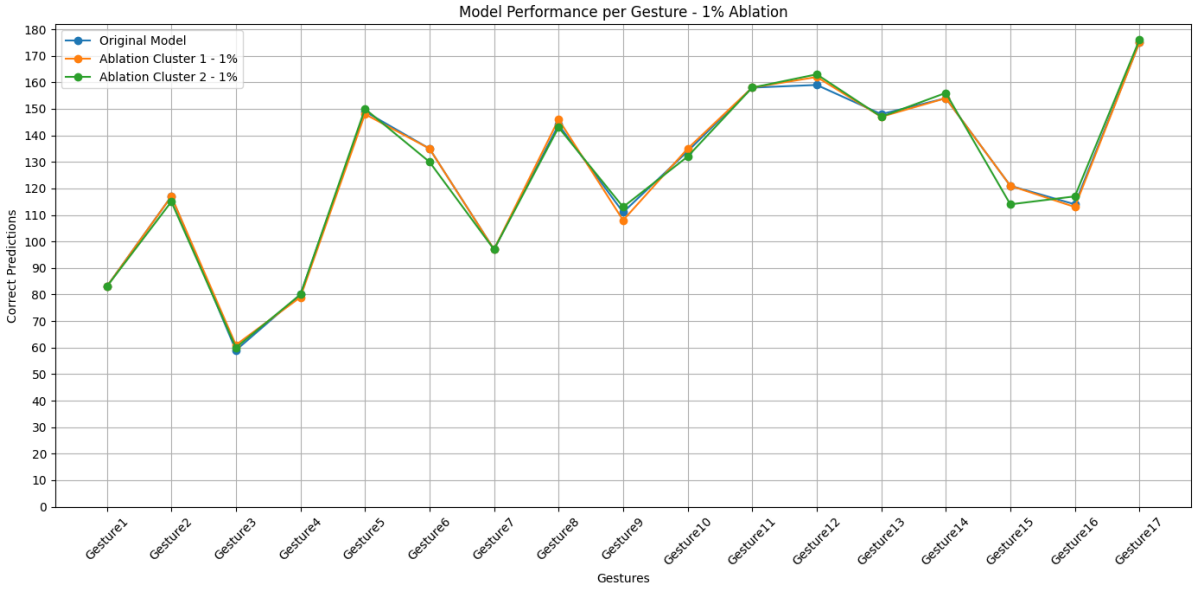


Figure C.121: Model Performance per gesture, comparison between original model, and the single layer ablation of 1% for both Clusters of the fifht convolutional layer.

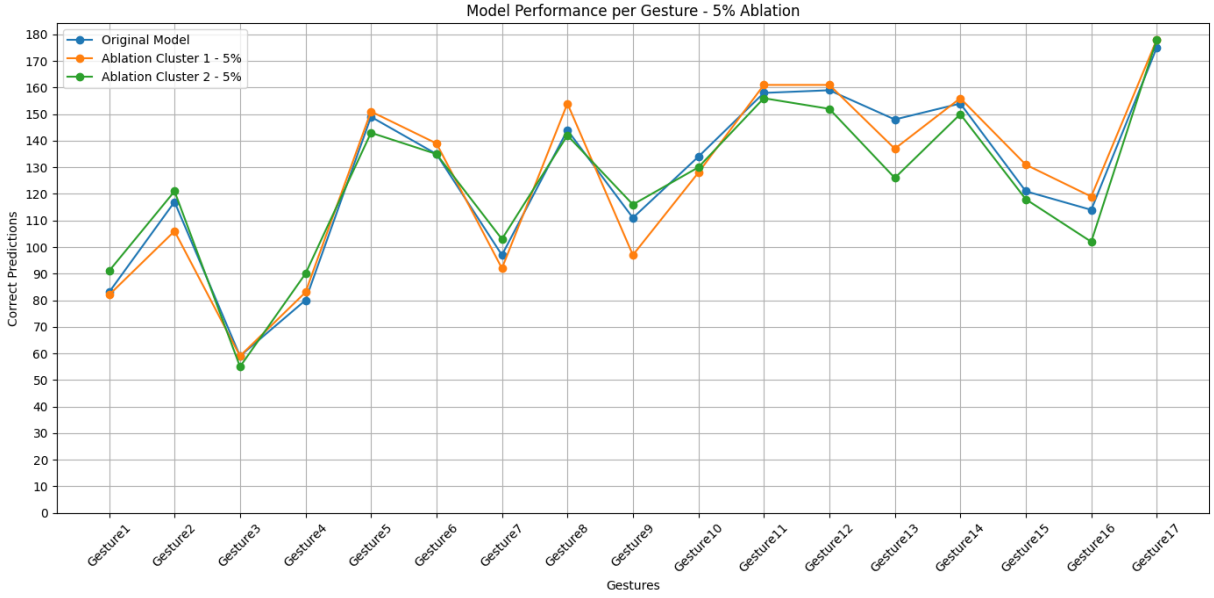


Figure C.122: Model Performance per gesture, comparison between original model, and the single layer ablation of 5% for both Clusters of the second convolutional layer.

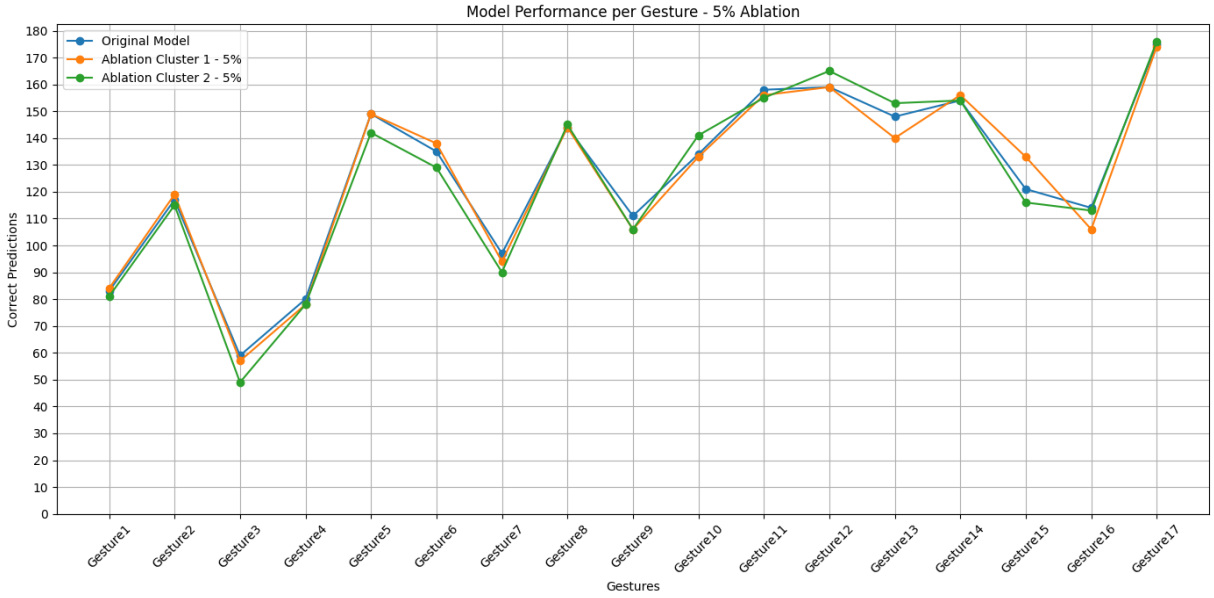


Figure C.123: Model Performance per gesture, comparison between original model, and the single layer ablation of 5% for both Clusters of the third convolutional layer.

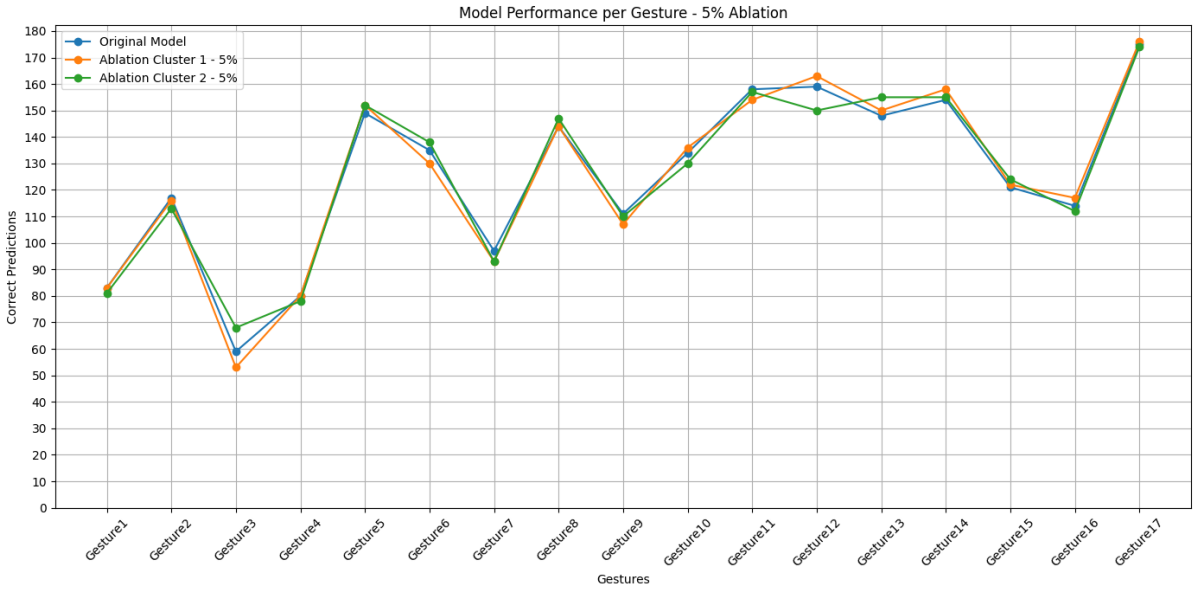


Figure C.124: Model Performance per gesture, comparison between original model, and the single layer ablation of 5% for both Clusters of the fourth convolutional layer.

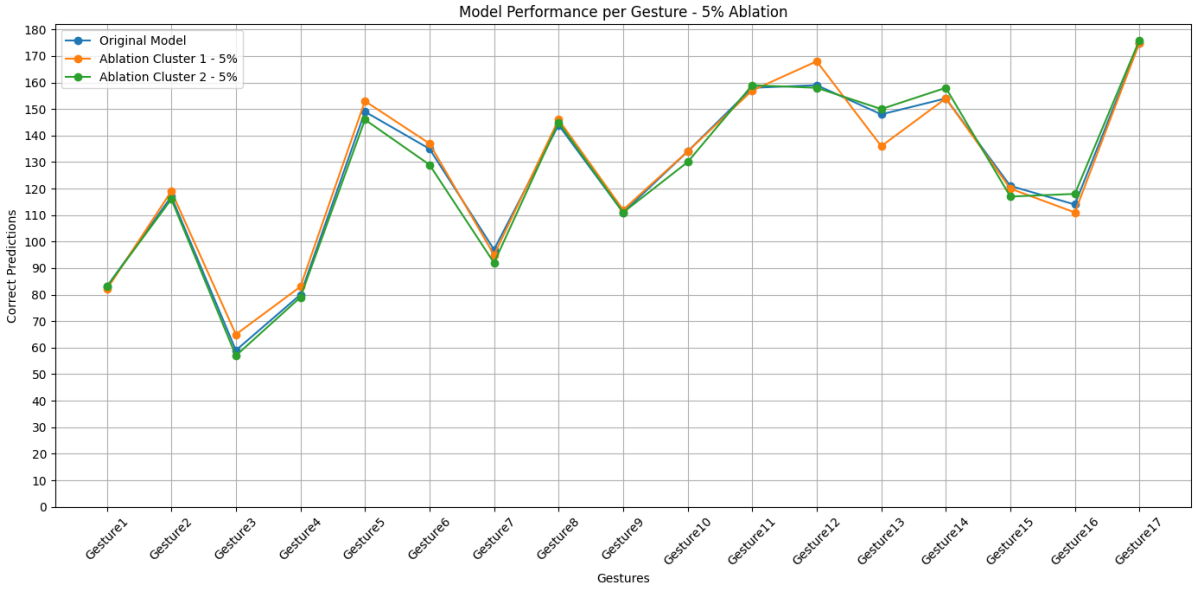


Figure C.125: Model Performance per gesture, comparison between original model, and the single layer ablation of 5% for both Clusters of the fifth convolutional layer.

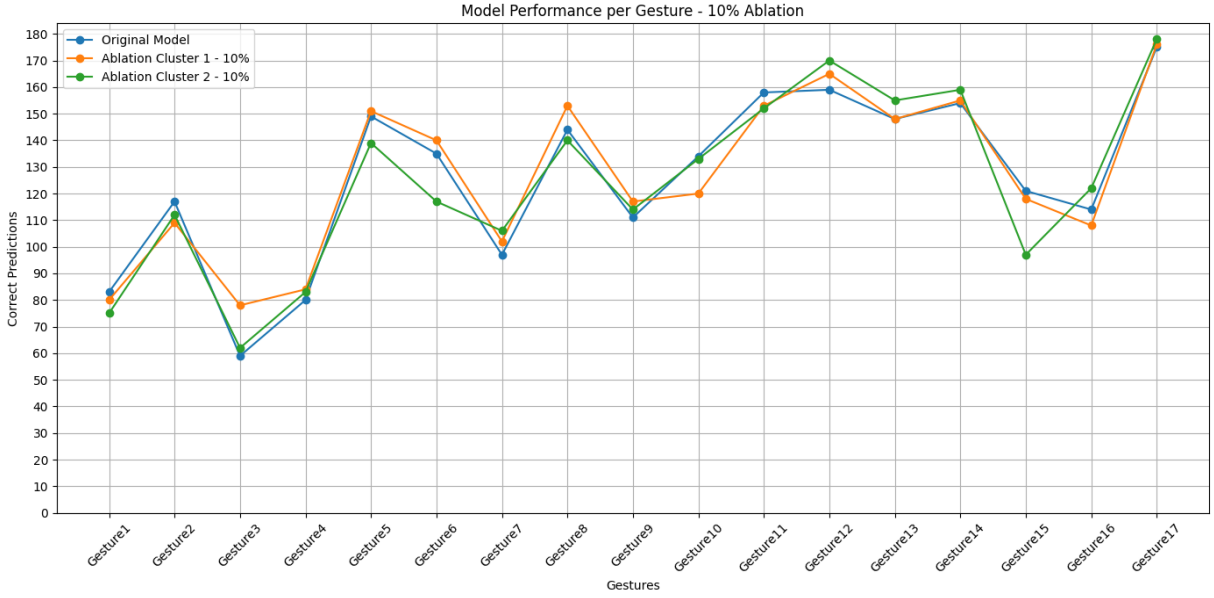


Figure C.126: Model Performance per gesture, comparison between original model, and the single layer ablation of 10% for both Clusters of the second convolutional layer.

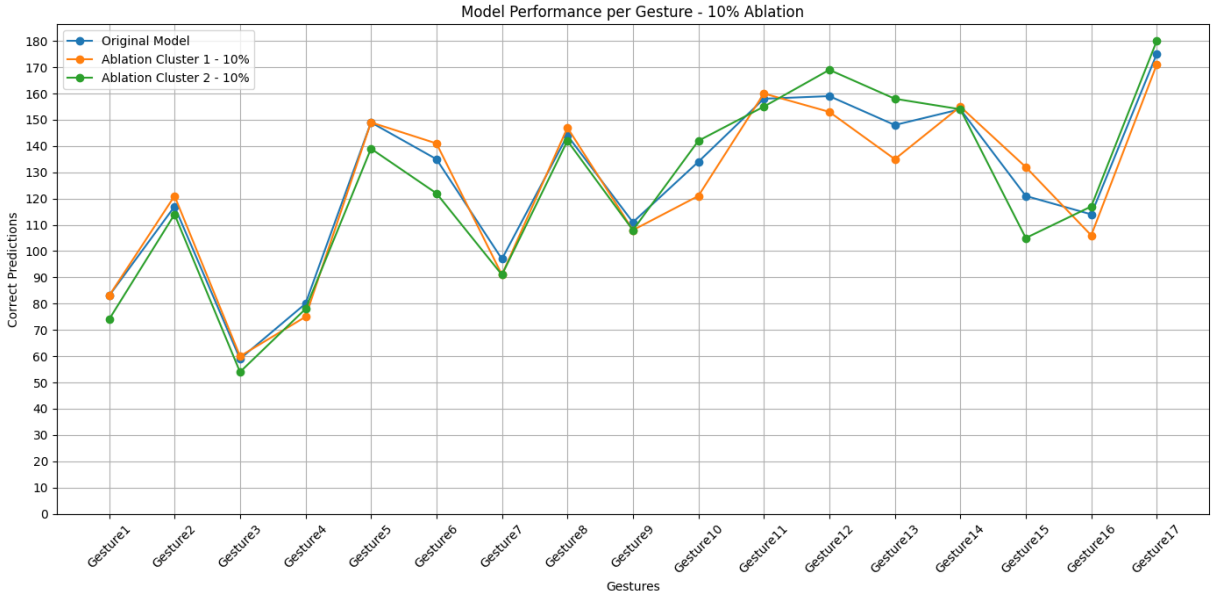


Figure C.127: Model Performance per gesture, comparison between original model, and the single layer ablation of 10% for both Clusters of the third convolutional layer.

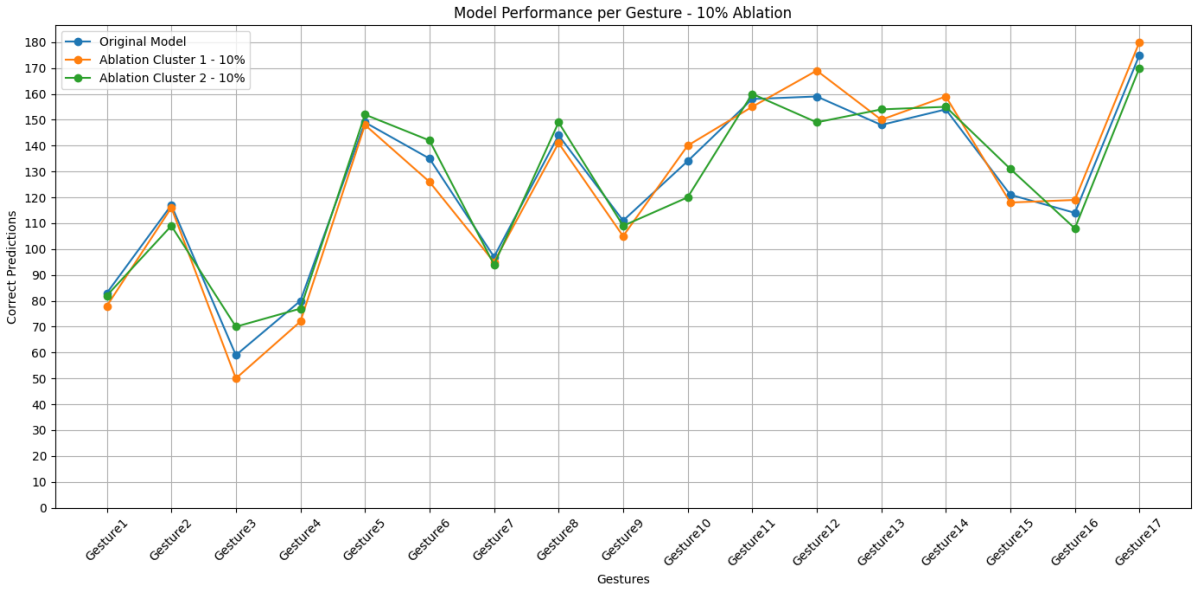


Figure C.128: Model Performance per gesture, comparison between original model, and the single layer ablation of 10% for both Clusters of the fourth convolutional layer.

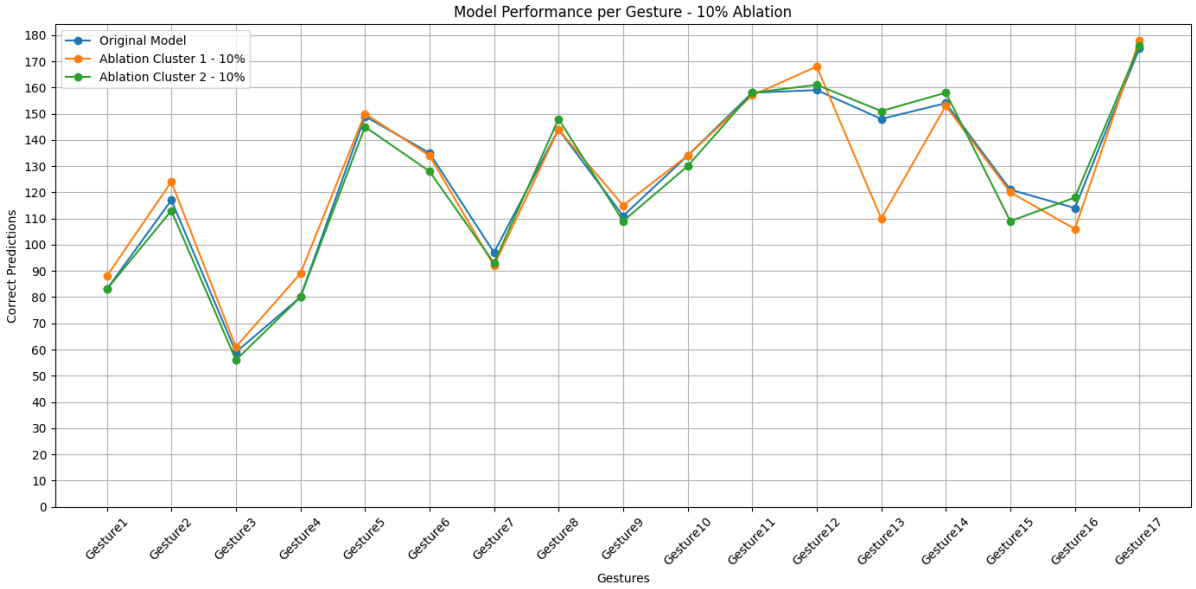


Figure C.129: Model Performance per gesture, comparison between original model, and the single layer ablation of 10% for both Clusters of the fifth convolutional layer.

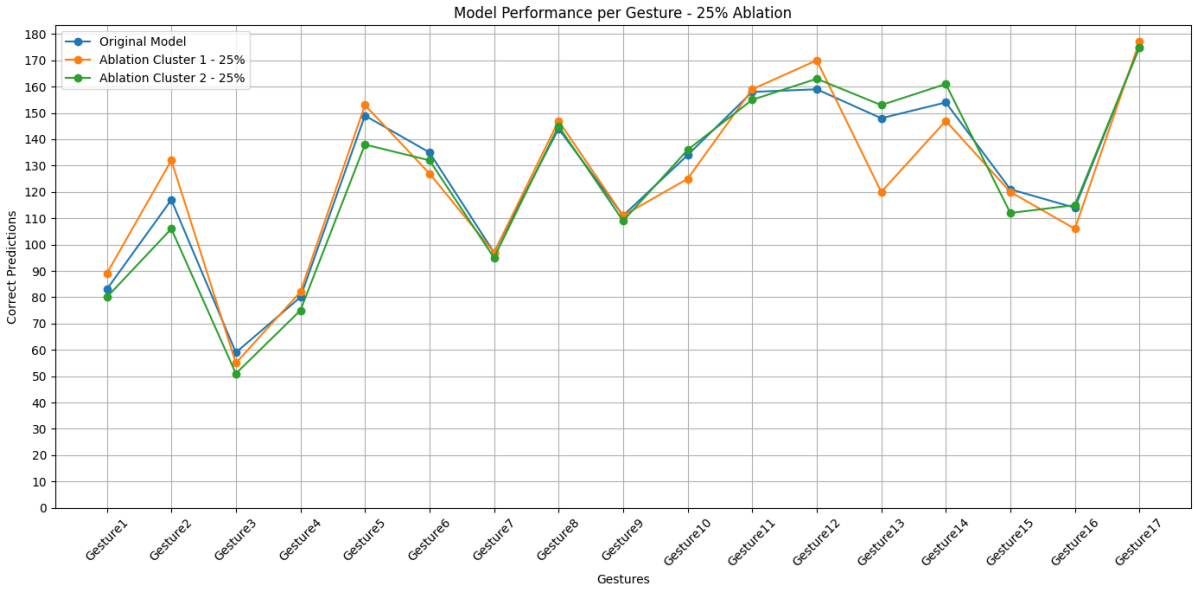


Figure C.130: Model Performance per gesture, comparison between original model, and the single layer ablation of 25% for both Clusters of the fifth convolutional layer.

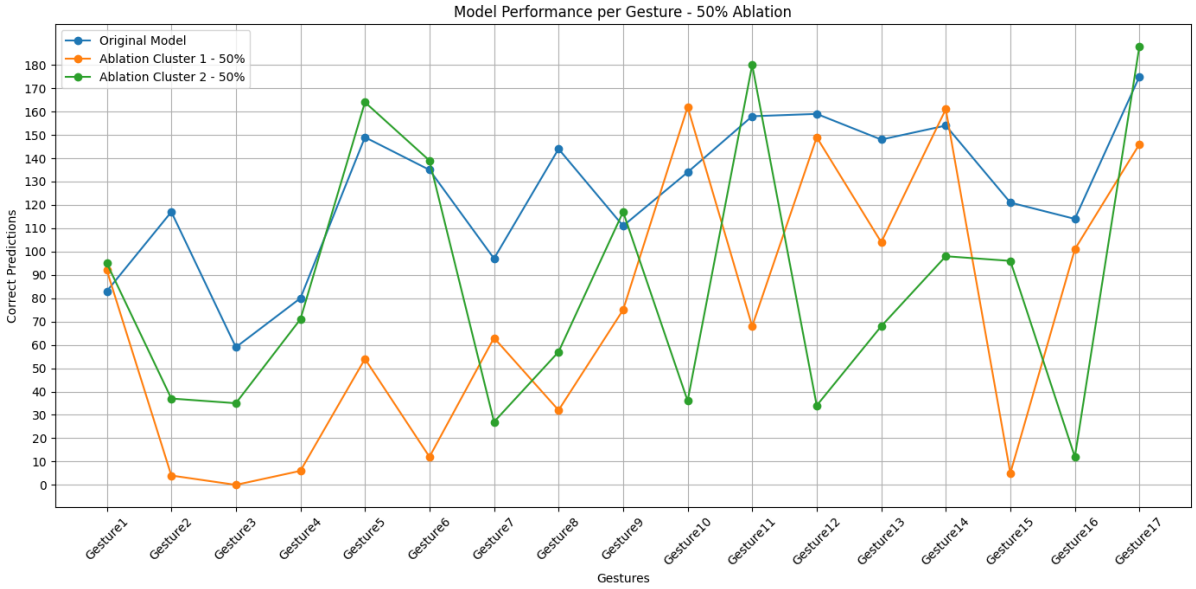


Figure C.131: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for both Clusters of the first convolutional layer.

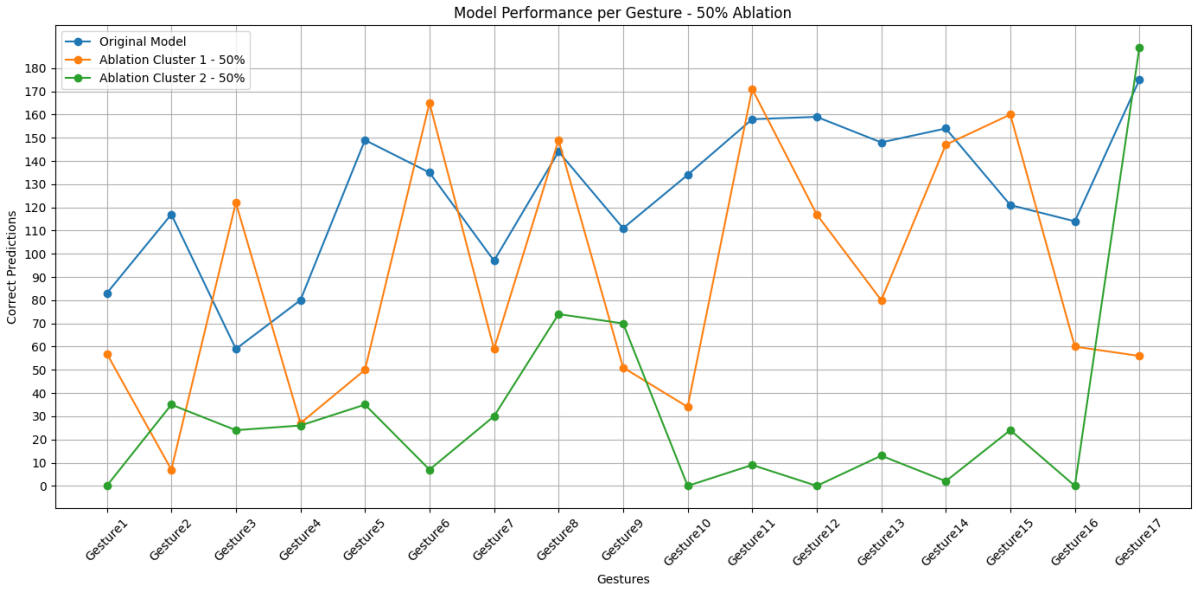


Figure C.132: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for both Clusters of the second convolutional layer.

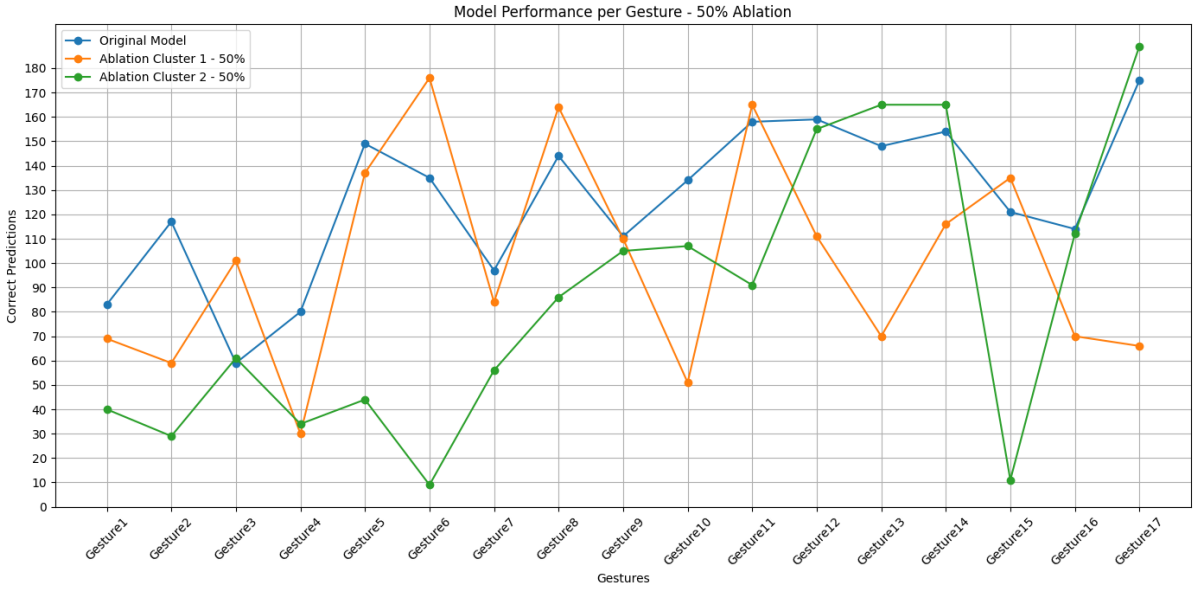


Figure C.133: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for both Clusters of the third convolutional layer.

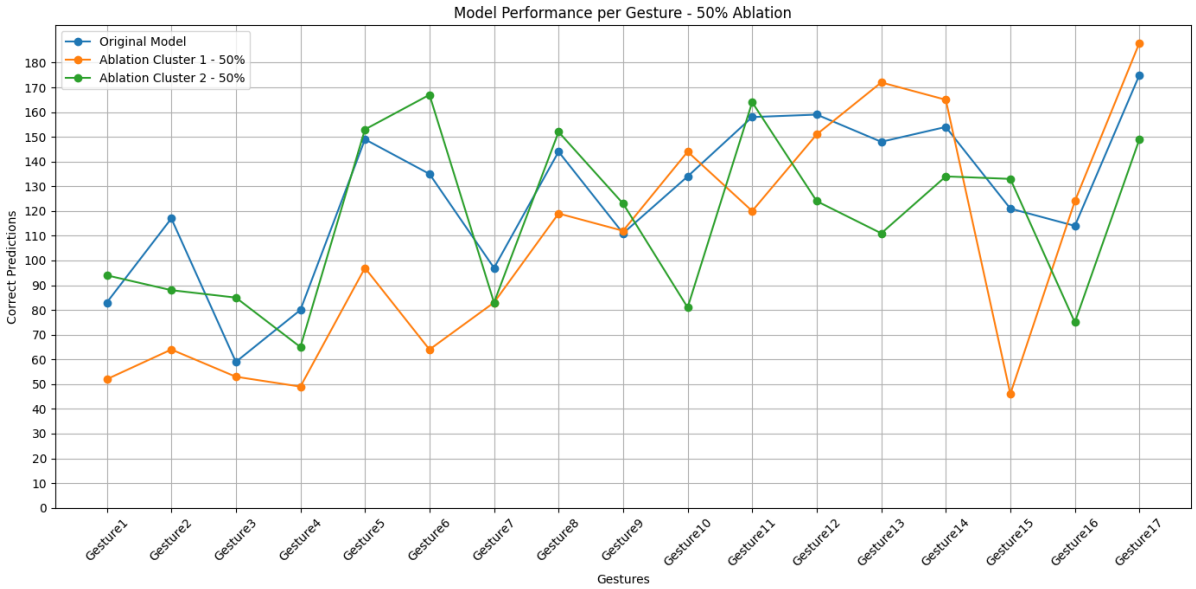


Figure C.134: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for both Clusters of the fourth convolutional layer.

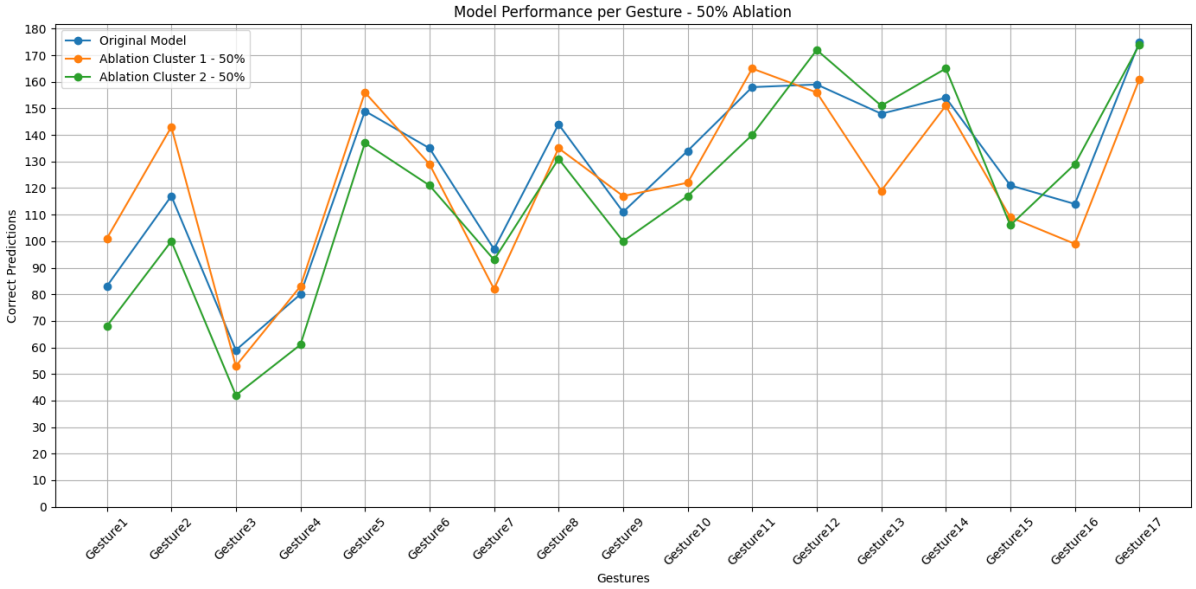


Figure C.135: Model Performance per gesture, comparison between original model, and the single layer ablation of 50% for both Clusters of the fifth convolutional layer.

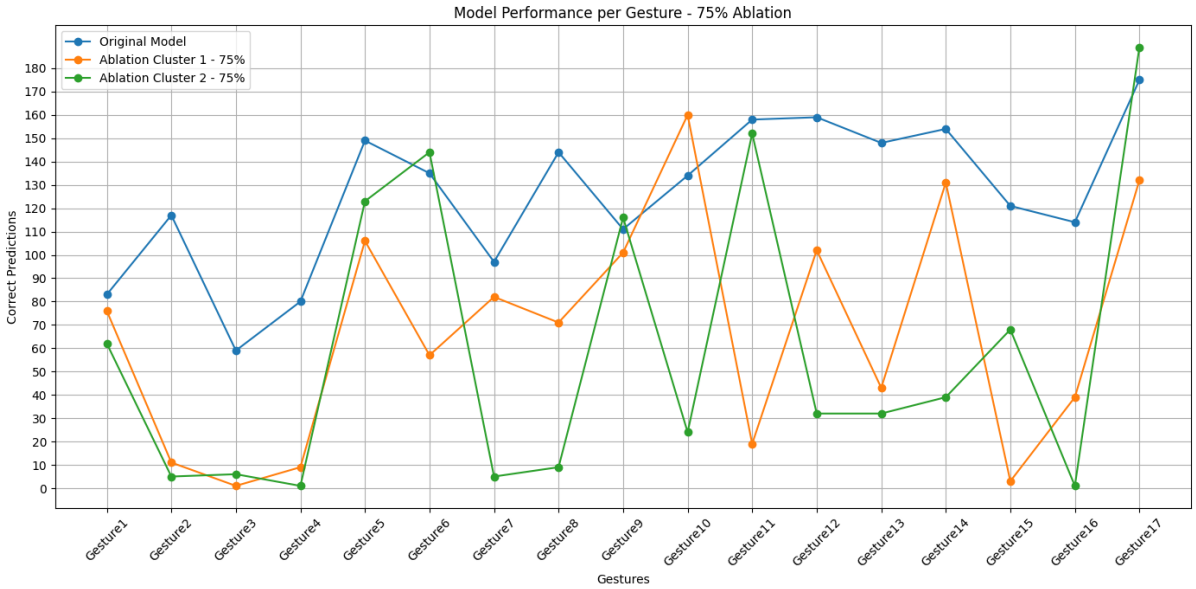


Figure C.136: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for both Clusters of the first convolutional layer.

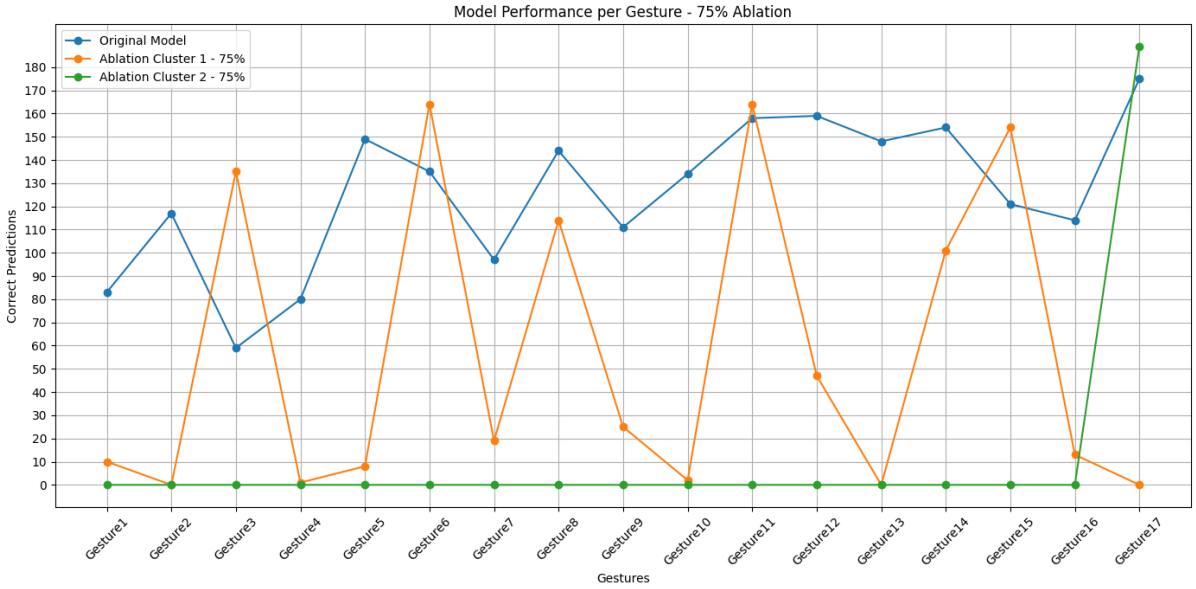


Figure C.137: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for both Clusters of the second convolutional layer.

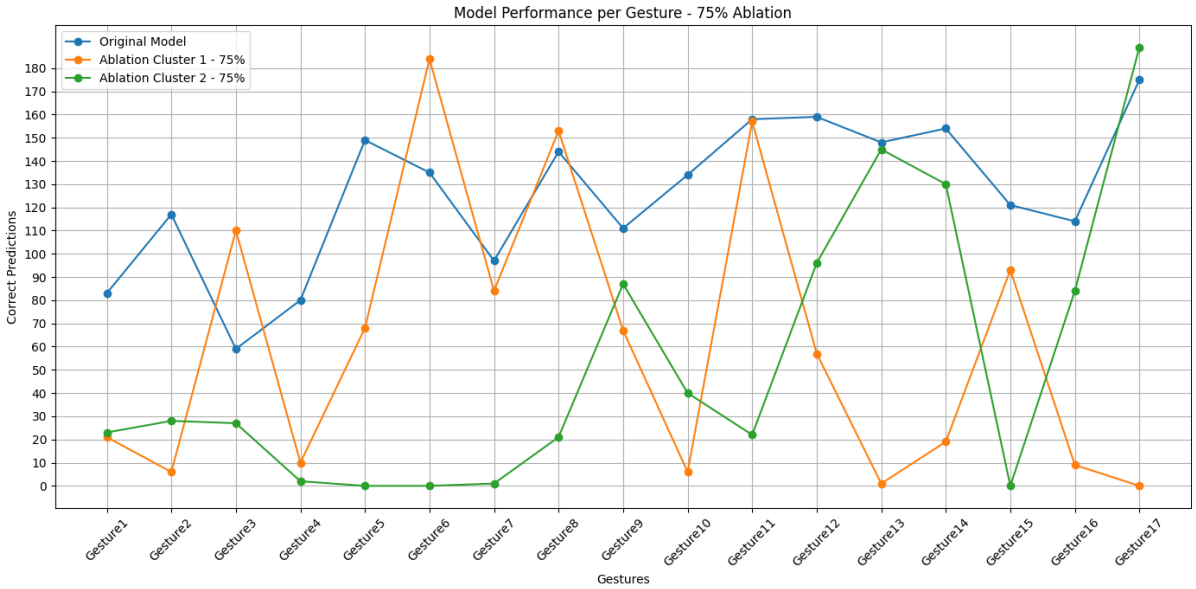


Figure C.138: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for both Clusters of the third convolutional layer.

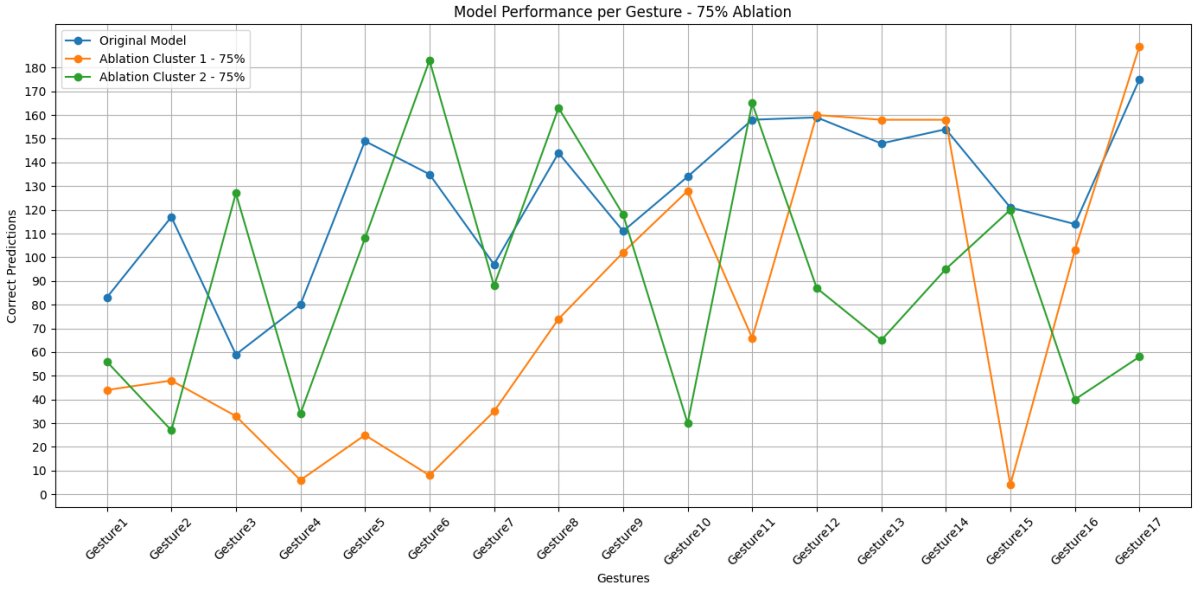


Figure C.139: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for both Clusters of the fourth convolutional layer.

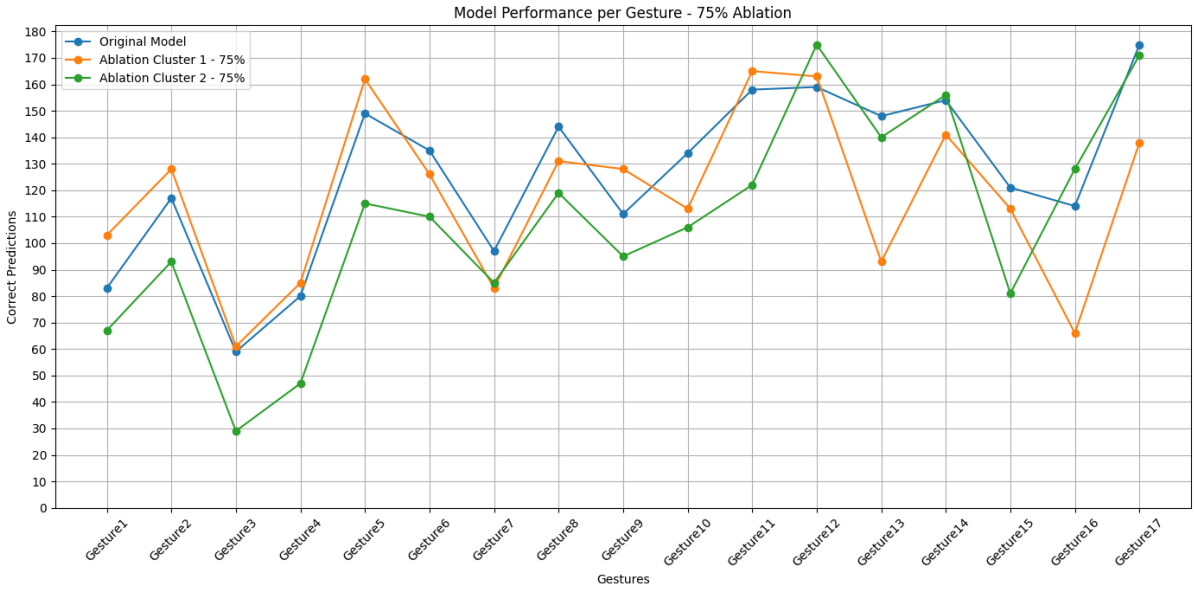


Figure C.140: Model Performance per gesture, comparison between original model, and the single layer ablation of 75% for both Clusters of the fifth convolutional layer.

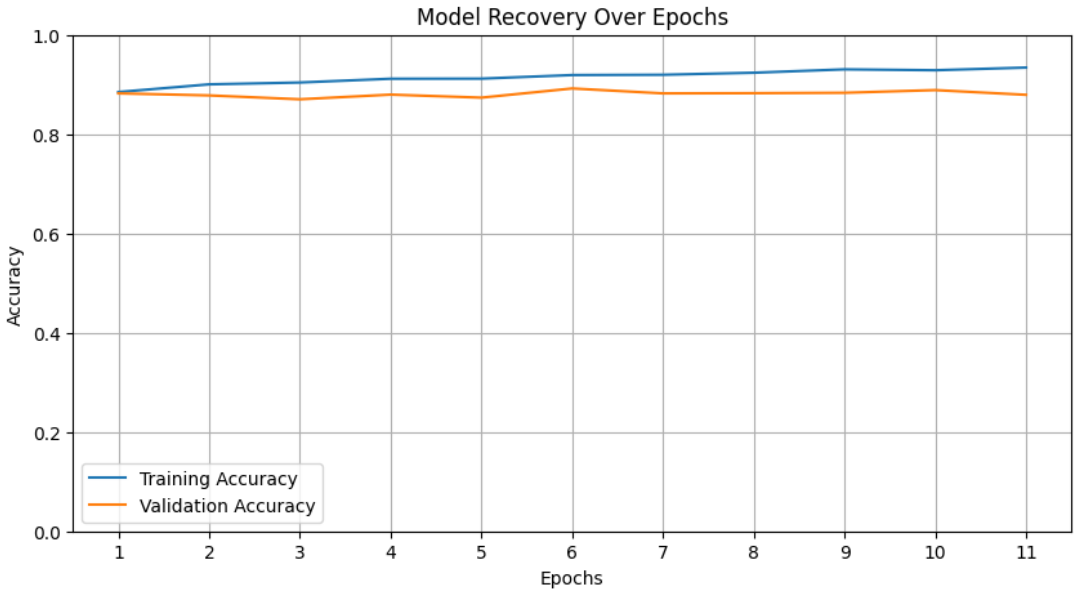


Figure C.141: Model recovery plot over the total epochs for model recovery of 25% ablation cluster 1

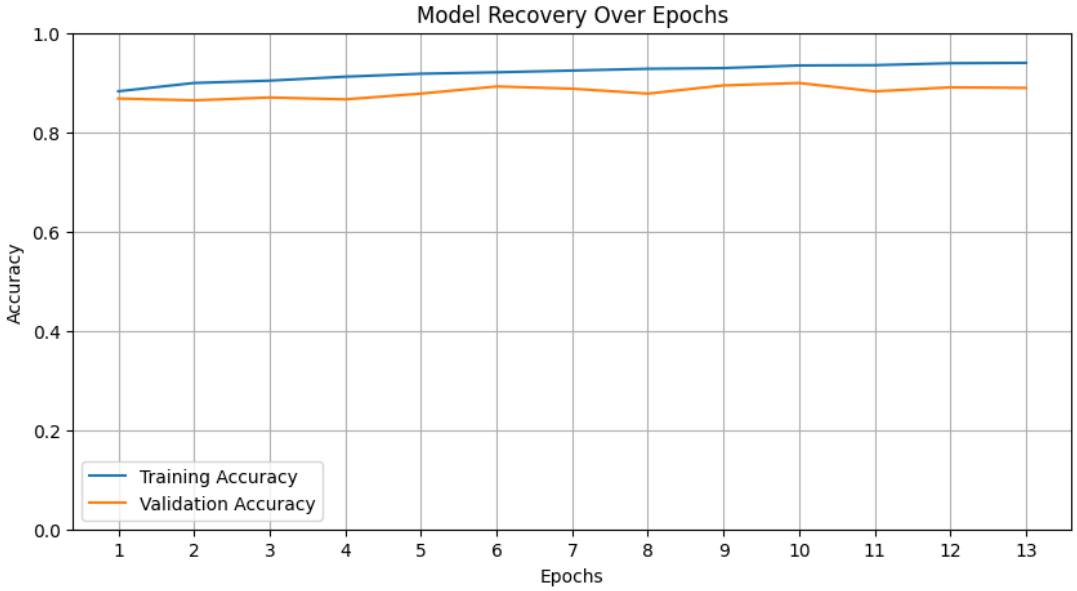


Figure C.142: Model recovery plot over the total epochs for model recovery of 25% ablation cluster 2

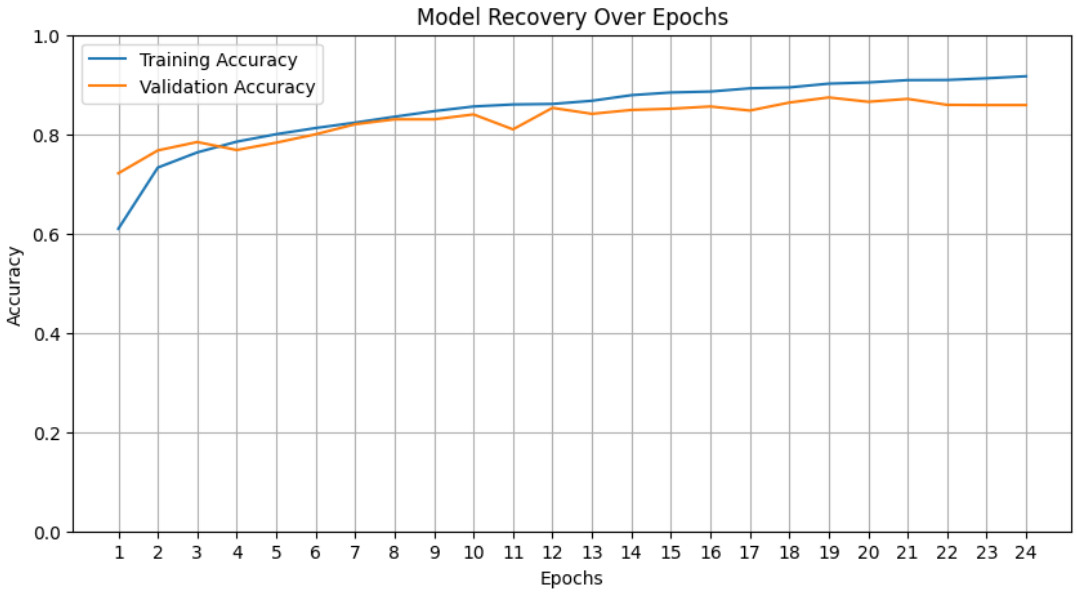


Figure C.143: Model recovery plot over the total epochs for model recovery of 75% ablation cluster 1

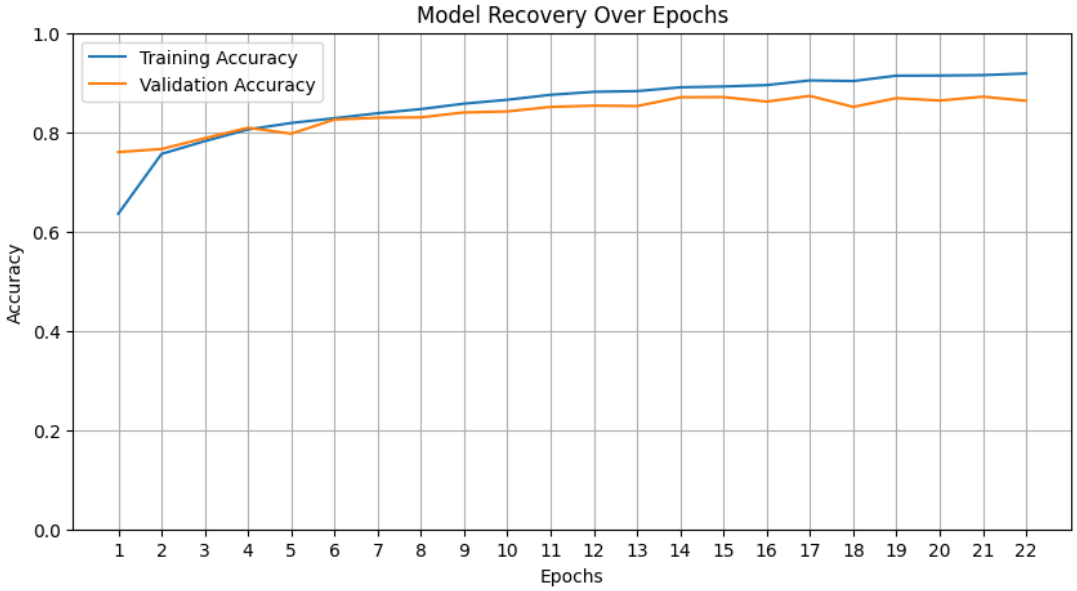


Figure C.144: Model recovery plot over the total epochs for model recovery of 75% ablation cluster 2

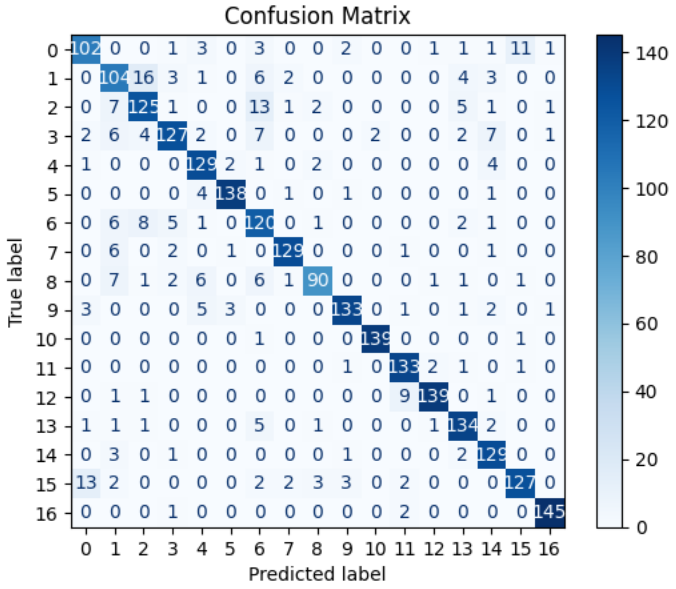


Figure C.145: Confusion matrix of recovery model for 25% ablation cluster 1

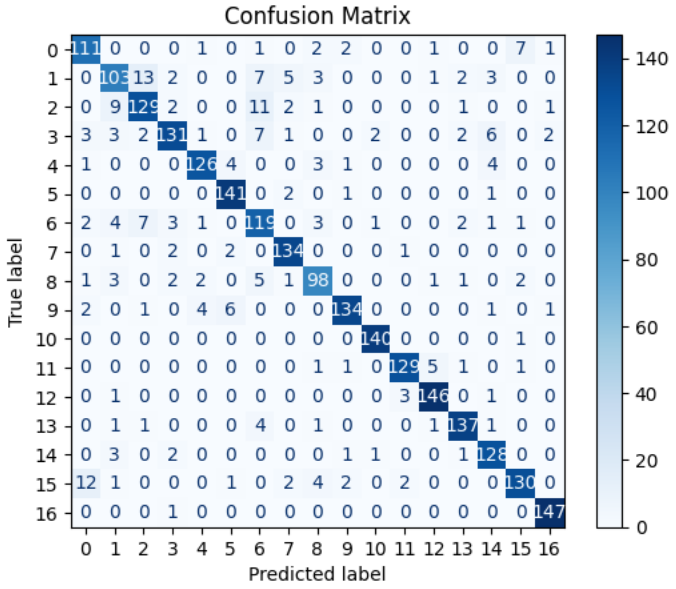


Figure C.146: Confusion matrix of recovery model for 25% ablation cluster 2

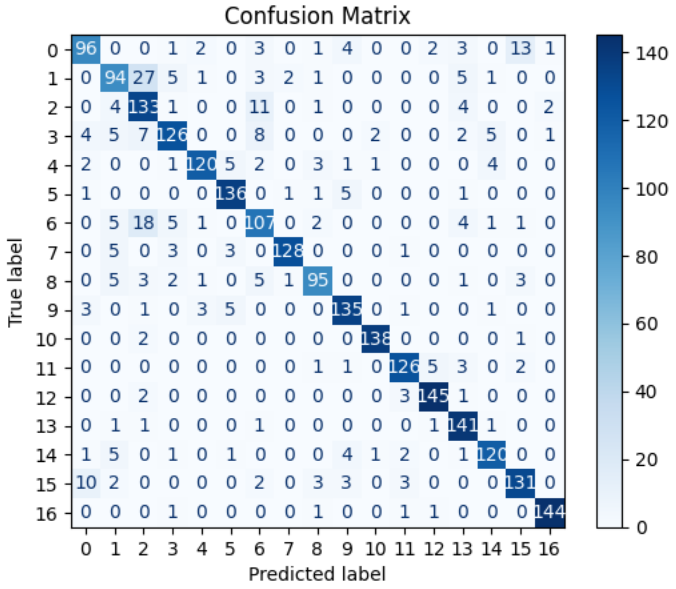


Figure C.147: Confusion matrix of recovery model for 75% ablation cluster 1

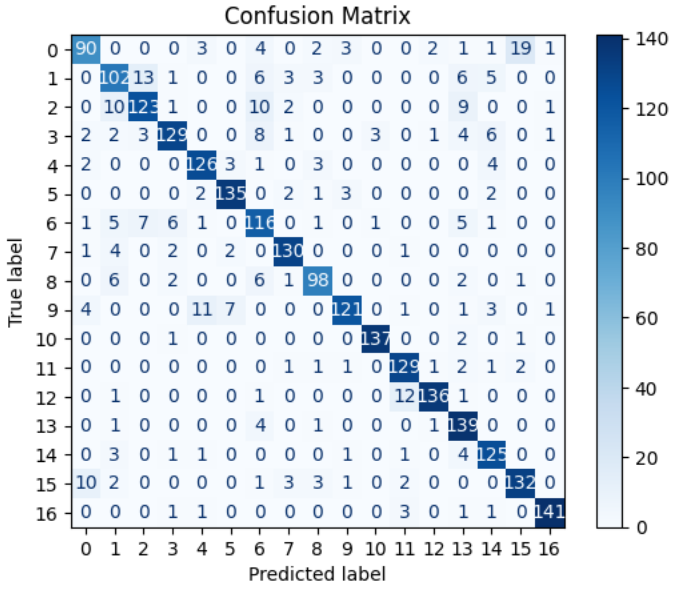


Figure C.148: Confusion matrix of recovery model for 75% ablation cluster 2