Applications of Neural Networks in Classifying Trained and Novel Gestures Using Surface Electromyography

by

Erik Lloyd

A thesis

presented to the University of Waterloo

in fulfillment of the

thesis requirement for the degree of

Master of Applied Science

in

Systems Design Engineering

Waterloo, Ontario, Canada, 2019

Authors Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Current prosthetic control systems explored in the literature that use pattern recognition can perform a limited number of pre-assigned functions, as they must be trained using muscle signals for every movement the user wants to perform. The goal of this study was to explore the development of a prosthetic control system that can classify both trained and novel gestures, for applications in commercial prosthetic arms. The first objective of this study was to evaluate the feasibility of three different algorithms in classifying raw sEMG data for both trained isometric gestures, and for novel isometric gestures that were not included in the training data set. The algorithms used were; a feedforward multi-layer perceptron (FFMLP), a stacked sparse autoencoder (SSAE), and a convolution neural network (CNN). The second objective is to evaluate the algorithms' abilities to classify novel isometric gestures that were not included in the training data set, and to determine the effect of different gesture combinations on the classification accuracy. The third objective was to predict the binary (flexed/extended) digit positions without training the network using kinematic data from the participants hand.

A g-tec USB Biosignal Amplifier was used to collect data from eight differential sEMG channels from 10 able-bodied participants. These participants performed 14 gestures including rest, that involved a variety of discrete finger flexion/extension tasks. Forty seconds of data were collected for each gesture at 1200 Hz from eight bipolar sEMG channels. These 14 gestures were then organized into 20 unique gesture combinations, where each combination consisted of a different sub-set of gestures used for training, and another sub-set used as the novel gestures, which were only used to test the algorithms' predictive capabilities. Participants were asked to perform the gestures in such a way where each digit was either fully flexed or fully extended to the best of their abilities. In this way the digit positions for each gesture could be labelled with a value of zero or

one depending on its binary positions. Therefore, the algorithms used could be provided with both input data (sEMG) and output labels without needing to record joint kinematics. The post processing analysis of the outputs for each algorithm was conducted using two different methods, these being all-or-nothing gesture classification (ANGC) and weighted digit gesture classification (WDGC). All 20 combinations were tested using the FFMLP, SSAE, and CNN using Matlab.

For both analysis methods, the CNN outperformed the FFMLP and SSAE. Statistical analysis was not provided for the performance of novel gestures using ANGC method, as the data was highly skewed, and did not fall on a normal distribution due to the large number of zero valued classification results for most of the novel gestures. The FFMLP and SSAE showed no significant difference from one another for the trained ANGC method, but the FFMLP showed statistically higher performance than the SSAE for trained and novel WDGC results.

The results indicate that the CNN was able to classify most digits with reasonable accuracy, and the performance varied between participants. The results also indicate that for some participants, this may be suitable for prosthetic control applications. The FFMLP and SSAE were largely unable to classify novel digit positions and obtained significantly lower performance accuracies for novel gestures for both analysis methods when compared to the CNN. Therefore, the FFMLP and SSAE algorithms do not seem to be suitable for prosthetic control applications using the proposed raw data input, and the output architecture.

Acknowledgements

I would like to express my sincerest gratitude to my supervisor Dr. Ning Jiang for his support both for my thesis and entrepreneurship activities I have undertaken during my Masters. I would also like to thank the Department of Systems Design Engineering for the support and resources supplied to me in support of taking my work in academia and translating this work into commercial technologies.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

sEMG – surface electromyography

iEMG – intramuscular electromyography

DoF – degrees of freedom

ANGC – All or nothing gesture classification

WDGC – weighted digit gesture classification

SSAE – Stacked Sparse Auto Encoder

CNN – Convolution Neural Network

FFMLP – Feed-Forward Multi Layer Perception

ANN – Artificial Neural Network

AE – Autoencoder

TFM – Topographical Force Mapping

GC – Gesture Combination

SNR – Signal to Noise Ratio

CNS – Central Nervous System

1. Introduction

The prosthetic control systems that will be explored in the literature review of this thesis investigated the classification of trained gestures from sEMG signals collected from the forearms of participants, but did not investigate these algorithms' ability to classify novel gestures. The first commercially available pattern recognition prosthetic control system follows a similar trajectory, where the system must be provided with the corresponding sEMG data for any function, or gesture that the user wants the hand to perform, within the limited number of gestures. Theoretically, for such a system to be able to replicate every possible movement of the human hand using a conventional discrete classification method, the system would need to be trained on every single movement of the human hand. As a result, any signal patterns that correspond to a gesture or motion that the system has not been trained to recognize will be misclassified as one of the gestures within the trained model.

The goal of this thesis study was to investigate whether a myoelectric control algorithm could be developed that can classify both trained and novel gestures. (1) The first objective of this study is to evaluate the feasibility of three different algorithms in classifying raw sEMG data. These algorithms are a feed-forward multi layer perceptron (FFMLP), a stacked sparse autoencoder (SSAE), and a convolution neural network (CNN). Except for a notch and bandpass filter, the data has not been pre-processed. (2) The second objective is to evaluate the algorithms' abilities to classify novel isometric gestures that are not included in the training data set, and to determine the effect of different gesture combinations on the classification accuracy for trained and novel gestures. As will be discussed in section 2.3, prior research has demonstrated that a few select synergies can account for a high percentage of variability in both trained and novel gestures, which

provides justification for the development of algorithms that can be used to extract common signal features from sEMG data, and therefore classify both trained and novel gestures [1]. “Muscle synergies are robust and generalizable enough to *predict* the EMG patterns in individual muscles for new gestures” [1]. (3) The third objective is to predict binary digit positions without training the network with kinematic data from the participants hand, due to the real-world constraints that amputees would face in not being able to measure the kinematics of their ipsilateral hand from where the sEMG signals are recorded.

2. Literature Review

2.1. A Brief History of Modern Prosthetics

The loss of one or more hands is a traumatic experience that affects both function and social interactions. Despite the progress the field has made, especially through the 20th century, developing a complete functional replacement for the natural human hand still proves to be a difficult challenge from both engineering and clinical perspectives [4]. The earliest mechanical arm, called the Ballif arm, was invented in 1812, and was controlled with other upper arm movements [5]. The system known as the Dorrance split hook, released in 1909, became the most common functional solution for upper-limb amputees. After WWII, there was a surge in research programs to improve prosthetic technology, then another surge in the 1950s after the Thalidomide crisis [6]. The Thalidomide crisis from 1957 to 1961, occurred as a result of the drug thalidomide, which was released in the US, Europe and Japan. The drug was used to treat morning sickness in pregnant women, and after being on the market for years was eventually tied to a large number of birth defects, including limb absences [7].

The first myoelectric prosthesis was developed in Munich, and was initially not a portable system, but was instead tethered to a building power supply. This arm used vacuum tubes for the control system and had basic open and close functionality. Later iterations incorporated batteries into the prosthetic [5]. The first clinically viable myoelectric prosthesis was developed by Russian experts in the 1960s, and since then, the innovations of upper extremity myoelectric prosthetic control schemes have been incremental [8].

There is still much improvement to be made as it relates to prosthetic controls, and as a result, much attention in the academic community has been focused on improving prosthetic control

systems using pattern recognition algorithms. One of the earliest applications of pattern recognition for sEMG classification was explored in Hudgins et al. from 1993, which employed an artificial neural network to classify sEMG signals collected from surface EMG electrodes [9].

2.2. What is Myoelectric Control?

Surface Electromyography or sEMG signals are the sum of trains of action potentials, which are generated by active motor units [10]. These signals are recorded as electrical potentials along skeletal muscles, and in the case of sEMG, are measured using electrodes placed on the surface of the skin. This type of biosignal is used in both diagnostic applications, for conditions such as neuromuscular diseases, and in control applications for peripheral systems such as computers and robotics [11]. An example of a visual representation of these sEMG signals in the time domain can be seen in Figure 2.1, which is a figure showing recorded sEMG signals from [12].

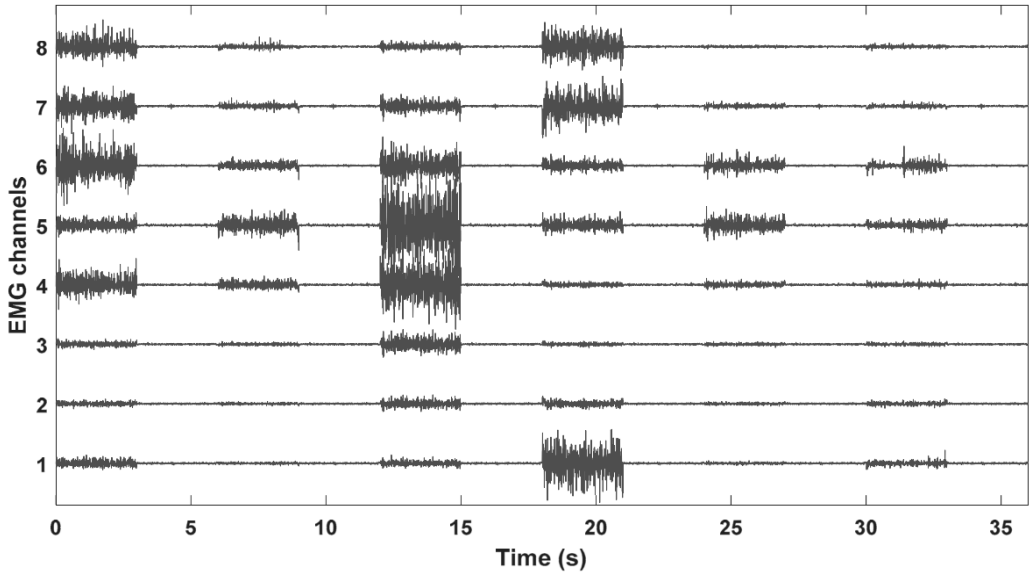

Figure 2.1, this figure depicts a subsection of sEMG signals recorded from the myo armband from a study in [12], The y-axis shows the amplitude of each signal on each of the 8 sEMG channels, and the x-axis is time.

In the United States, 57% of total amputees are arm amputees, with 80% of these patients using a prosthesis of some kind, and 30-50% using myoelectric systems [13]. In order to provide these prosthetic users with higher degrees of functionality beyond basic binary or state machine control systems, such as the ability to control multiple DoF at the same time, more complex and robust control methods must be developed [14].

Oskoei and Hu describe three categories that existing myoelectric control systems can be divided into. First generation control is a basic on/off system with no speed variability. Second generation systems include proportional control, threshold manipulation, signal simplification and adjustment of muscle contraction rate. The third generation involves the application of microprocessors that allow for an infinite range of adjustments of myoelectric characteristics. Programmable microprocessor technology allows for the implementation of advanced signal processing methods, as well as the implementation of machine learning algorithms. This 3rd generation control system has allowed for the development of pattern recognition control schemes [8].

There are two groups of myoelectric control schemes, pattern recognition and non-pattern recognition. Non-pattern recognition-based controllers are mainly threshold or state machine systems, meaning that their responses or outputs to control a device are all predetermined. In pattern recognition-based control, classifiers are used to identify predefined classes of functions from myoelectric signal patterns [8]. This literature review and by extension this thesis, focuses on pattern recognition-based controls.

2.3. Muscle Synergies

In order to better understand what these pattern recognition algorithms are learning, it is important to understand the concept of muscle synergies. Muscle synergies are described as the complex

interaction between multiple muscles, which are initiated by an individuals higher-level control inputs. According to [3] “muscle synergies are influential in myoelectric control schemes due to sEMG inputs encoding muscle activation timing, shape and intensity”. According to [15], several studies have suggested that the CNS may simplify control by activating a small number of modules formulated as muscle synergies, or activation profiles across a set of muscles, the linear combination of which can generate diverse motor patterns. There are two main approaches for developing muscle synergy-based control systems, these being pattern recognition and motor learning. Pattern recognition systems decode muscle activity for intuitive control systems, by associating complex patterns of sEMG signals with specific outputs. Motor learning-based control systems consist of specific mapping functions that relate sEMG inputs to system outputs. A user learns to operate this system by receiving feedback signals while interacting with the control system. As stated earlier, the focus of this report will be on the use of pattern recognition, in the case of muscle synergies this is largely due to the fact that motor learning methods “have yet to be validated as a viable simultaneous control scheme for any myoelectric applications” [3]. [1] describes how a few muscle synergies can be used as a predictive framework that can account for up to 90% of the variance in sEMG signals generated by individual muscles using iEMG. This study did not focus on classifying new gestures but did address the fact that multiple synergies can predict the variance in the activation of individual muscles in both trained and untrained hand postures.

2.4. State-of-the-art Commercial myoelectric prosthetics

2.4.1. Current prosthetics

Myoelectric prosthetics have made significant strides since the 20th century with the introduction of multiple multi-articulating mechatronic devices. [16] discusses six state-of-the-art

commercially available prosthetic hand technologies. Devices such as the i-Limb and Vincent hand have six powered DoFs, and therefore offer a greater number of possible movements that the user can control, when compared to single DoF prosthetic devices. Myoelectric devices like the Bebionic 3 by Ottobock, which is capable of five motorized DoFs, offers 14 different functions available for amputees [17], which are possible due to the sophisticated electromechanical design of these prosthetics. Despite the robotic sophistication of these 21st century prosthetic devices, their control systems are still limited, and most of these arms are not controlled through a means that are intuitive or natural [18]. Many commercial prosthetics use a sequential control strategy. This type of control strategy uses a signal such as co-contraction to switch to different preprogrammed grasp patterns. Another type of control strategy, referred to as movement trigger, involves patterns of muscle contraction signals which correspond to different grasp patterns [18].

The i-Limb Quantum, a commercially available prosthetic developed by Ossur. employs a method called “Gesture Control”, which involves switching gestures using a specific movement of the users arm. This switching action is initiated by recording signals from sensors such as gyroscopes, accelerometers, and magnetometers built into the hand, to switch through functions. Alternatively, the i-Limb also uses “Grip Chips”, with are Bluetooth enabled tags attached to specific objects, that activate certain movements when the prosthesis is brought into proximity of the chips. Neither of these alternative methods are intuitive, as they are not comparable to the means that amputees used to control their natural limbs before their amputations [18]. The control system for most modern prosthetics, until recently, were largely non-pattern recognition systems as discussed previously. While pattern recognition controls have been the focus of academic research for several decades, this is only beginning to change for commercial prosthetics.

2.4.2. Current Commercial Pattern Recognition Systems

Coapt released the first commercially available pattern recognition prosthetic controller in the form of the Complete Control system. This prosthetic sEMG controller employs pattern recognition algorithms to provide amputees with a more intuitive means of controlling their myoelectric prosthesis [18]. The algorithm used in the complete control system is an LDA, which is a linear classifier [19]. The Coapt Complete Control system is a major improvement in clinical prosthetic control, but is still limited to preprogrammed classes, where each class must be provided with three seconds of corresponding sEMG data for the algorithm to learn the associated signal sEMG signal patterns related to each class [20]. According to [18], users of the Complete Control system can typically control three to six grasp patterns. So, while this type of pattern recognition system is beginning to see functional clinical applications, it is still limited in the number of different functions it can provide prosthetic users.

2.5. Architecture of an EMG Pattern Recognition System

2.5.1. Conventional Pattern Recognition Control

A conventional myoelectric pattern recognition control system consists of several stages responsible for taking raw sEMG data and transforming it into a viable control signal for a peripheral device, such as rehabilitation robots, wheelchairs [21] or prosthetic arms [14]. EMG signals are first amplified, filtered and digitized in order to be used with a programmable microprocessor-based system. The modified signal data is then segmented, and the necessary features can then be extracted from the segmented signal using a variety of time domain, frequency domain, time-frequency domain, and feature projection techniques. Based on these extracted features, a classifier is then trained to recognize patterns in the signals and assigns them to predetermined categories [8]. Processes for data segmentation, feature extraction and classification

will be discussed in greater detail. Based on the label outputs generated by the classifier, the controller generates output commands for the external systems, such as a prosthetic limb. If the system is a closed loop, any low-level feedback signals will be sent into the controller for processing, such as position feedback of joints or torque feedback. High level feedback signals can be sent as sensory feedback signals to the user's nervous system [8].

These types of algorithms represent a major shift towards prosthetics that are more intuitive to control, and more closely match the performance capabilities of natural limbs when these algorithms are paired with prosthetic devices such as the commercially available devices discussed in section 2.4.1. These algorithms are still limited in the performance they provide, as each of their possible control outputs, or classes, must each be provided with associated training data, as with the Coapt system discussed in section 2.4.2. Despite these advances, it is still not possible for amputees to initiate simultaneous control of individual fingers of the prosthesis, and instead control individual or combinations of finger movements based on predetermined gesture classes [18].

2.1. Feature Extraction Methods

Feature extraction should be used to increase information density of the EMG signals, retaining information that allows a classifier to identify different contraction patterns, while eliminating irrelevant data [22]. There are three major categories of features, time-domain, frequency domain, and time-frequency domain. Two different types of features are described by [3], these being synergy features and EMG features. Synergy feature methods extract data simultaneously from multiple EMG channels which provides cross-channel information about muscle synergies. EMG feature methods on the other hand extract structural characteristics about a single channel EMG signal.

2.1.1. Synergy Feature Projection Techniques

Feature projection methods transform multichannel inputs into lower dimensional sub spaces, and usually follow some type of EMG feature extraction. These techniques represent linear mixtures of sEMG signals which are associated with muscle synergies. Methods such as Principle Component Analysis, Independent Component Analysis, Non-Negative Matrix Factorization, Linear Discriminant Analysis, Clustering, Nonlinear Projections and Spatial Filtering are all methods used to reduce the dimensionality of incoming data. The method of projecting multi-channel information into lower dimensions inevitably results in a loss of data [3].

2.1.2. Alternatives to Discrete Gesture Classification

In order to develop a control system that allows for the manipulation of individual fingers, some papers have proposed classifying finger joint positions [23][14][24]. These papers have investigated the use of this labelling method to classify the joint angles of the hand for isometric gestures that the algorithms were trained with. One obvious limitation for this method for prosthetic control applications is in the simple fact that amputees no longer possess a hand that can be used simultaneously to construct these joint angle labels that correspond to certain EMG patterns. To circumvent this problem, one study proposed using bilateral movements to provide both the sEMG data and joint labels for amputees. The data collection protocol involved placing the sEMG electrodes on the amputees residual limb, while a sensor glove, used to record joint angles, was worn on their contralateral hand. The participant performed bilateral isometric contractions to provide both the sEMG data and corresponding labels. This method achieved good results, with classification accuracies of 86% on average across 7 gestures [23]. Another paper by Hioki and Kawasaki discusses a similar methodology of mapping sEMG signals to joint angles on the hands of able-bodied participants. Because this study used able-bodied participants with intact

limbs, the joint angles were recorded using a Cyberglove on the ipsilateral hand with respect to the arm on which the four sEMG channels were mounted [14], as opposed to measuring contralateral joint angles as was the case with [23].

These strategies for measuring joint angles have limitations, as they require additional hardware in order to label the sEMG data with the corresponding joint angles, which makes the training process more complicated and tedious for amputees when considering this method for commercial prosthetic applications. A glove with integrated sensors is not the only explored method for recording joint angles. A study by Au et al uses a camera system to track joint angles in the shoulder and elbow of a participant, while simultaneously recording the corresponding surface EMG data [25]. This method could be applied in a similar manner to the previous paper that mapped finger joint angles and would eliminate the need for the user to don an extra piece of hardware, like a Cyberglove, in order to measure and record joint angles. However, this method still requires an external apparatus to develop the training labels, and camera systems depend on sufficient brightness, distance and unobstructed line of site [11].

A method by Jiang et al describes an alternative method that does not require any additional hardware systems to label the data. This method involved labelling the data depending on whether each digit in a particular gesture was flexed or extended [26]. This method provided the inspiration for the output architecture used in the study conducted for this thesis.

2.2. Data Segmentation

Data segmentation involves separating incoming data into frames based on a determined window length. The sum of the length of a data segment and the processing time for the system should be less than 300 ms, as a time delay any longer than this is noticeable by the user. There is a trade-off

between segment length and time delay, as shorter segments can result in greater bias and variance in the signal, but a longer segment results in greater delay. In order to reduce this delay to 128ms or even 32 ms without a substantial decrease in performance, a continuous segmentation method should be adopted. “In continuous segmentation, a dense stream of decisions is produced using overlapped segments. Continuous segmentation relies on both transient and steady-state myoelectric data. Post-processing methods are designed to manage excessive classified output, and improve system performance” [8].

Transient signals on their own are less viable for use in myoelectric control, as these signals are generated by muscle contractions initiated from a resting state [8]. This means that for a user to switch between classes, they must first relax their muscles, thereby increasing the time it will take a user to switch between movements. Most errors in classification for steady state signals appear during transitions between classes. Therefore, it is recommended that data sets from these transition states be eliminated to improve controller accuracy [8].

In data segmentation there are two methods for windowing data, adjacent windowing and overlapping windowing. Adjacent windowing has the sEMG recordings being separated into neighbouring and disjointed frames of data. For overlapping windowing, the system generates more frames of data, with the new generated segments containing both new signal values, as well as signals values from the previous segment of data. The amount of overlap determines the composition of the new frame in relation to new data points versus data from the previous frame. A smaller segment shift increment produces a semi-redundant set of class decisions that results in improved response time and accuracy [8].

Englehart discusses utilizing post-classification processing methods to improve classification accuracy. In [2], majority voting is used to reduce classification error. The method compared a current classifier decision to the decisions of the past 500 ms. Most errors the system made after post-processing with majority voting occurred during the transitional states. Considering this, it may also be beneficial to train a classifier to recognize data from transitional regions and produce a state of inactivity for a controlled device, such as a prosthetic arm, when the signals indicate a transition. Alternatively, Hargrove used majority voting as well, and compared a classifier output with the previous eight and the next eight decision, which compares a decision to classifications made both ahead and behind it by 256 ms [27].

2.3. Classification Methods

As mentioned in section 2.2, pattern recognition algorithms have been the focus of prosthetics control research for decades. Most forms if statistical and learning classifiers have been applied to myoelectric control [22]. Techniques such as feed forward multi-layer perceptrons, autoencoders, convolution neural networks, Bayesian classifiers, fuzzy logic, linear discriminant analysis, support vector machines, hidden Markov model and k nearest neighbours have all been applied to classify sEMG signals for gesture recognition applications [28].

According to Scheme et al, the most popular classifiers are Linear Discriminant Analysis, Support Vector Machine, and hidden Markov Model. The advantage of the LDA is the simplicity of the algorithm and ease of training [22], which is evident when considering that the LDA is the algorithm of choice for the first commercially available myoelectric pattern recognition system as discussed in section 2.4.2.

2.4. Barriers to Robustness of EMG Control

Higher performance capabilities of classification algorithms are often observed in a controlled lab environment, but these same ideal conditions do not exist in the real world. Several factors can result in substantial performance degradation for many pattern recognition-based control schemes. These factors include electrode shift, variation in force, variation in limb position or posture and changes in the sEMG signals [22].

Scheme and Englehart observed that a 1cm shift of four electrodes placed around the forearm resulted in a 5-20% increase classification error if the shift was distal, and 40% if the shift occurred rotationally. The use of wider electrodes, as well as placing larger gaps between electrodes improved robustness, but not to the same effect as incorporating data from shift positions into training data set. Daily training sessions may be necessary whenever the prosthesis is donned in order to account for slightly different electrode positions day to day. Over time, the accumulation of training data from slightly shifted electrode positions may result in a more robust control system that can account for slight shifts [22]. The number of electrodes has also been shown to impact the performance of sEMG classification algorithms. As demonstrated in a study by Hargrove et al, a 16-channel array of equally spaced electrodes was applied, and symmetrical sets of 2, 4, 8 and 16 channels were compared to optimal sets of the same size. In this experiment, six different features were performed with six healthy male subjects. The time domain features of the collected sEMG recordings were used to train an LDA classifier. This study demonstrated that the improvement in performance begins to diminish after four channels [27].

Hargrove et al also describe the ideal electrode placement locations when using a reduced set of channels. The authors state that surface electrodes placed over the extensors/supinator, flexor carpi ulnaris, and flexor digitorum subliminus provide good classification accuracy [27]. While some

studies such as [29] and [30] use fewer sEMG channels of four and two electrodes respectively. Other studies such as [12] and [31] used eight evenly spaced electrodes around the forearm for the classification of sEMG signals for CNN classification applications. Therefore, the use of eight evenly spaced bipolar channels was adopted for the study proposed in this thesis.

According to Ives et al, the upper limit of the sEMG frequency range is 400-500 Hz. For sEMG sampling, in order to avoid aliasing, the sampling rate must be at least a factor of two larger than the maximum frequency. In the case of sEMG signals, this means that the sampling rate of the sEMG sensors must be at least 800-1000 Hz [32].

Limb position, or posture, will also impact the real-world performance of a pattern recognition algorithm. This can be due to involuntary muscle contractions that manifest when the limb is held at certain positions, and can result in increased classification errors between 3.8-18% [33]. For prosthetic users, the effect of limb posture can be due to the mounting of an upper-limb prosthesis to a user's residual limb via a socket, which depending on the posture of the residual limb, can apply pressure to different muscle groups in the forearm. This compression can displace muscles and alter the sEMG signal, as well as result in mechanical stimulation of the muscle [22]. Topographical force maps (TFMs) could be used as an extra dimension for training, which may help a system to compensate for posture changes by measuring the change in pressure on contact points inside of the socket. TFM uses the mechanical response of soft tissue during muscle contractions in order to measure neural activity. In the case of upper limb prosthetic devices, the inside of a socket would be lined with an array of force resistive sensors to measure the pressure that soft tissue places on the inside of the prosthetic socket [34]. "The socket and residuum are not rigidly connected, so placing a heavy weight in the prosthetic hand would shift the weight distribution across the surface if the limb and would unload some sensors while increasing load on

other sensors" [34]. Adding accelerometers can also make classifiers more robust against limb posture changes [3], however, the collection of the necessary training data to account for the various limb positions will still fall to the user [22].

Amplitude cancellation occurs when positive and negative phases of separate motor units overlap, causing deconstructive interference which results in a loss of information. This resulting loss of information means that sEMG in situations where amplitude cancellation occurs, will not be able to accurately estimate the motor unit activity [10]. Muscle synergy features can make a system more robust towards amplitude cancellation [3].

Some EMG signal changes over time are the result of both short and long-term variations in the recording environment. These changes can be the result of external interference, shifting electrodes, changes in electrode impedance, and muscle fatigue. Some of these signal changes can be removed using filtering and shielding techniques to eliminate external interference, but intrinsic sources of variation are more difficult to remove [22]. Creating a pattern recognition system that can adapt to these changes is not trivial, as the system must know what phenomena it needs to compensate for, and how to compensate for them. To date, there has yet to be a stable unsupervised solution developed [22].

Variation in the level of contraction initiated by the user can cause problems in reliability for pattern-recognition based classifiers, as pattern recognition systems work by clustering patterns of EMG signals. Therefore, muscle contractions with varying degrees of intensity can be different enough from one another to result in misclassification of a movement. This was tested in an experiment where 11 subjects were told to perform motions at 20% to 80% of the strongest contractions, they could comfortably generate for nine motion classes, and one inactive class.

Using a time domain feature set with an LDA classifier, the system was trained using the force levels described. The classifier was tested using all force levels in order to determine the performance of the pattern recognition system to classify levels of contractions it had not previously encountered. The error for these types of contractions was greater than 32%. One means of counteracting the degradation in performance is to train a classifier at every force level. This method was then tested at all force levels and produced a classification error of 17%. The second method tested was an attempt to reduce the amount of training data that a user would have to generate and involved training the system on only the lowest 20% and highest 80% contraction levels. The performance from this training set was only slightly worse, at 19% error [22].

2.5. Classification Methods for the Proposed Study

Classification methods are used to take preprocessed data points and transform them into control signals. Classifiers that use supervised learning must be provided with both input information, and the desired outputs. Some classifiers will use this information to generate discrete output values, corresponding to predefined classes or categories. Other classifiers can generate a continuous output, which makes them suitable as both discrete classifiers and as classifiers for proportional control applications. The classifiers discussed in this section will be the focus of this study. The applications of these algorithms in sEMG signal detection applications will be explored in greater detail to provide a deeper background on their relevance for the classification of sEMG signals as it relates to this thesis.

The classifiers chosen for this thesis study are different types of artificial neural networks. These being a feed-forward multi-layer perceptron, a stacked-sparse autoencoder, and a convolution neural network. ANNs have proven to be effective and robust in real-time myoelectric control schemes [3], and the specific algorithms as will be discussed in section 2.5 have been explored in

previous studies involving sEMG signal classification, with promising results. ANNs are applicable for simultaneous control compared to algorithms like LDAs or SVMs, as they are capable of supporting dual association, and do not need to train separate classes in order to classify simultaneous movements for multiple DOFs [3]. Another reason for this choice of classifier is that ANNs can be used to produce discrete on/off outputs, as well as continuous variable output signals for proportional control.

2.5.1. Feed Forward Multi-Layer Perceptron (FFMLP)

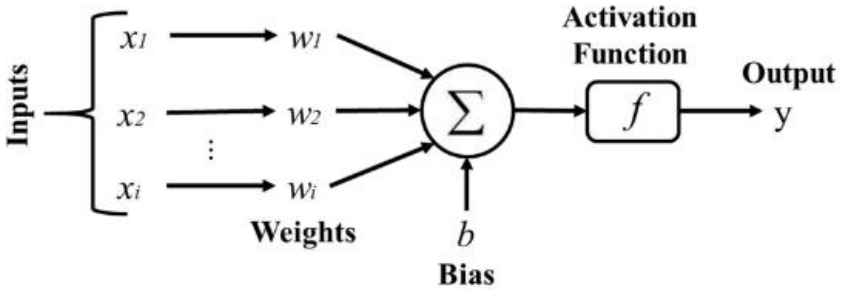

Figure 2.2, This figure depicts a single node or neuron from an neural network, with 3 inputs and one output [11].

An FFMLP is made up of a series of interconnected nodes, or neurons, and multiple nodes can be connected to form a network [11]. Each node is a computational unit that produces an output using a function called an activation function from the sum of inputs. These multi-layer networks can be trained to approximate a given function based on predefined input and output data. This works by adjusting the weighted connections between neurons within the network, until the error between the actual outputs, and the desired outputs is below a desired threshold [35]. This iterative weight adjustment process is referred to as backpropagation [11]. There have been several studies that focused on the applications of MLPs, also referred to as artificial neural networks, for the

classification of gestures from sEMG signals collected from a subjects forearm [11][14] [36][30][37][29].

2.5.1.1. Relevant Studies for MLPs in sEMG signal classification

One study used an artificial neural network with wavelet transform features in order to classify the movement of individual fingers using sEMG. This experiment used both two sEMG channels and four sEMG channels to collect data at a sampling frequency of 2000 Hz from 10 healthy subjects. These volunteers performed six different finger motions, with the end goal to detect the movement of the extension and flexion of the index finger, middle finger and thumb for each subject. The neural network design used for this experiment contained three layers, with 36 nodes in the hidden layer. The actions for each digit were represented by two output nodes, one for flexion and one for extensions with a total of six output nodes, while the input layer consists of 20 nodes. The performance for each motion using two sEMG channels was below 80%, but the use of four channels resulted in a performance for each motion of over 80% [26]. It would have been beneficial for the researchers to explore the ability of the proposed system to generalize for unseen digit position combinations, given that the proposed architecture has the capability of producing novel combinations of binary finger positions. Also, the authors provided no further description of the logic rules to handle the continuous outputs of a neural network, nor how a rest gesture would be classified with the proposed output architecture. [30] used two sEMG electrodes to record sEMG signals from the forearms of each participant at 1000 Hz. A multi-dimensional input which consisted of a multitude of feature extraction algorithms was passed to the input of the artificial neural networks that were tested. The authors used Mean Absolute Value, Root Mean Square, Mean Frequency, Zero Crossing, Slope Sign Change, and Standard Deviation for the classification of four different movements, which consist of open, close, index flexion, and middle/ring finger

flexion. For the classifier, the authors used a feedforward artificial neural network with a Levenberg-Marquard algorithm, and hidden layer with 10 hidden units. The researchers obtained an average classification accuracy of 83.5%. In this study, only two recording sites were used to collect sEMG data. The authors could have explored the use of a higher number of channels to improve classification accuracy. As demonstrated in other studies such as [27], the performance improves when using a higher number of collection sites.

A study by Au and Kirsch sought to predict shoulder and elbow kinematics from sEMG signals. The experiment used the sEMG signals from six shoulder and elbow muscles to predict three shoulder movements. In order to track the movement of the shoulder an elbow, the trunk, upper arm and forearm were tracked using nine LED markers. The collected sEMG signals from six different muscle groups were filtered at 20-200Hz, then sampled at 500Hz. The data was collected from six able-bodied subjects, and two individuals with C5 Tetraplegia. The ANN was trained using backpropagation, with the sum of squared errors (SSE) between the actual values and the predicted values being applied to determine when to cease training. There were two hidden layers, with 20 nodes each that contained a tansig activation function. The architecture of the hidden layers was determined empirically, testing two and three layers, with between 1-200 hidden units per layer. The output layer consisted of four nodes with linear transfer functions. The system produced an error rate of 14.2% for joint angle of the shoulder elevation/depression task [25]. While the authors demonstrate that joint angle can be classified from sEMG data, it would have been useful for the researchers to explore other classifier algorithms, as in [12], where both SSAE and CNN were compared for the classification of sEMG signals.

2.5.2. Convolution Neural Network (CNN)

A convolution neural network is a neural network that is composed of one or more convolution layers, followed by one or more layers whose connectivity is similar to that of a standard multilayered network. The structure of a convolution neural network is designed to be used for 2D input images, or some other type of 2 dimensional inputs. CNNs can be beneficial as they are easier to train than fully connected networks like MLPs with the same number of hidden nodes, and have fewer parameters [38]. In recent years CNNs have become a greater interest to academic researchers seeking higher performance pattern recognition classifiers that can classify raw sEMG data without the need for a manual feature extraction pre-processing. Papers such as [12][39][40] use a variety of CNN architecture for the classification of discrete gestures.

CNNs have different types of layers that are utilized in their architecture to perform different operations on the outputs of the previous layers. An example of a convolution network used in sEMG application is shown in Figure 2.3, [12].

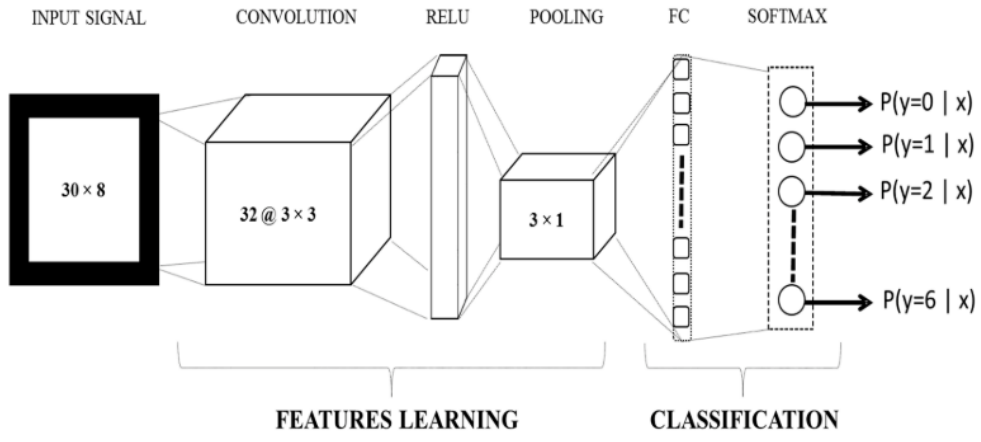

Figure 2.3, this figure depicts the CNN used in [12]*, which was trained to classify sEMG signals based on discrete target outputs. This figure provides a visual representation of how CNNs have been applied in sEMG signal processing.*

2.5.2.1. Convolution Layer

The convolution layer as mentioned above is often the first layer in a CNN. The convolution operation occurs when a kernel, or 2-dimensional tensor, is convolved with an input image. In the case of most sEMG applications, the input is usually a 2-dimensional tensor. The output of this convolution layer is a 3-dimensional tensor, with the 3rd dimensions size dependant on the number of kernels applied to the input tensor in the convolution layer. The size of the kernels as well as the stride, or the number of units by which each kernel is shifted over the image is determined by the designer [41].

2.5.2.2. Pooling Layer

The pooling layer serves as a means of dimensionality reduction within the CNN. This layer operates by taking the input to the pooling layer and dividing it into sub-regions. A dimensionality reduction operation is then performed on each subregion, either a max pooling function or an average pooling function, which is determined by the designer. The output from each of these sub-regions is therefore a single value, while the total output of the pooling layer is a 3-dimensional tensor. In the case of the CNNs explored for sEMG signals processing applications, the pooling layer is used after convolution layers, and is first superseded by a ReLu layer and a Batch Normalization layer that also follows the convolution layer [41].

2.5.2.3. ReLu Layer

ReLu layer is used to reduce the training time for CNNs. This layer takes any input values less than zero and sets them equal to zero. This way, the backpropagation algorithm is only passed positive values, which reduces training time by introducing sparsity into the network [42], and solves the problem of vanishing gradient [41]. The problem of vanishing gradient is when the error propagated through layers becomes smaller the further it is propagated through the network.

2.5.2.4. Batch Normalization

Batch Normalization is used to improve training by stabilising the distribution of inputs in a mini-batch, to the layer, and are a common component in many modern deep learning models [43]. The most widely accepted explanation of BN is that it reduces internal covariate shift, which is described as "the change in the distribution of layer inputs caused by updates to the preceding layers" [43]. Batch normalization is implemented in an additional layer which controls the mean and variance of the distribution of inputs to the layer.

2.5.2.5. Fully Connected Layer

Fully connected layer has a similar architecture to the FFMLP described in section 2.5.1, where each input to the fully connected layer from the previous layer has connections to all the activation functions in the following layer, hence the name, fully connected layer. These often follow a convolution layer, and as per the studies in which CNNs, are used between the final convolution layer of a network, and the network outputs [41].

2.5.2.6. Dropout Layer

This type of layer is only active during training but is not present during the prediction process. Because this type of network randomly eliminates certain connection with a set probability, it forces the network to evenly distribute weighting over multiple connections, rather than weighting one connection heavily and lower on others. This is therefore used to prevent overfitting in CNNs [41]. For a more in-depth explanation of the architecture of convolution neural networks, please see [41].

2.5.2.7. Relevant Studies for CNNs in sEMG signal classification

In recent years, convolution neural networks have been applied to the processing and classification of sEMG signals for applications in prosthetic control. One study [40] used a convolution neural network to classify hand gestures through the application of a high density sEMG electrode interface and the instantaneous sEMG patterns collect from this interface. The method used did not require any time window data segmentation, nor did it require manual feature extraction. The CNN was passed raw sEMG signals from the high-density electrodes and was trained with the instantaneous raw signals in relation to their corresponding gestures or movements. The researchers conducted this study with the use of 128 sEMG channels sampled at 1000 Hz. 18 healthy subjects were instructed to perform eight hand gestures, with each gesture being recorded 10 times. The results showed that 89.3% performance accuracy from a single instantaneous set of sEMG recordings from the HD array. Accuracies of 99% and 99.5% were achieved through the use of majority voting, in which 150 decision samples were used, equating to 150 ms of data [40]. The system in Geng et al achieved exceptional performance, however, the greatest limitation to its implementation as a clinical control methodology for upper limb myoelectric prosthetics is the use of high density sEMG. Using high density (HD) sEMG in a prosthetic socket while maintaining consistent skin contact for all channels may prove to be a major barrier for clinical and commercial use. Low-density (LD) sEMG has been historically used for research in prosthetic control applications due to its greater clinical applicability. Low density refers to 16 sEMG channels or less [39].

Another study by Rehman et al used a lower density, eight channel sEMG input with a time delay input. This study used the Myo armband to collect sEMG data from participants and used 150 ms windows of data from each of the armbands eight channels to classify seven different hand and

wrist postures. The Myo armband collects data at a sampling rate of 200 Hz per channel. This data was used to train a CNN with a single convolution layer. The average within-session error for this CNN across all participants was 2.4%, indicating a very high classification accuracy. For the between session performance, the CNN 6.2% error rate, which indicates that the proposed architecture for this paper was able to classify signals from multiple sessions that occurred on separate days [12].

Not all applications of CNNs have used raw data as an input to the network. Systems such as the one proposed in Allard et al proposed using frequency features with a CNN to classify discrete hand gestures from sEMG signals. The authors demonstrated the ability for their proposed CNN to be able to classify seven different hand and wrist gestures with an accuracy of 97.9% across six days. The algorithm was evaluated with three tests. The first evaluation was used to test gesture accuracy in real-time, and participants were asked to hold a random gesture for 10 seconds. The gesture was considered correctly classified if no more than 2 false consecutive, or 4 false non-consecutive classifications were made within the 10 second period. The authors do not discuss how these cut-offs for max time period, or max false classifications were selected. The second tests involved users controlling a robotic hand, which they were required to control with wrist and arm movements while holding an object in the hand. In this test all participants were able to hold the object in the hand for the required 120 seconds. Lastly, they were tested on a picking and placing task. The picking and placing task was compared to the speed and accuracy of someone controlling the same robotic arm with a joy-stick controller. The time required to perform this task with a joystick was 1minute 33 seconds, and 1 minute 45 seconds for the myo-control. This study demonstrated the use of a CNN, and therefore demonstrates that this type of algorithm is can be robust enough for prosthetic control applications [44]. It may have provided a clearer picture of

performance if the authors had selected real time tests that were in line with the standardized tests such as the Southampton Hand Assessment Procedure [45].

2.5.3. Stacked Sparse Auto-Encoder Neural Network

An auto-encoder is a type of artificial neural network algorithm that is trained using unsupervised learning. While a regular feed forward MLP utilizes supervised learning to train input data in relation to a predefined set of outputs, an auto-encoder uses only the input training data and backpropagation to assign the correct weights to the connections. The auto-encoder, using a backpropagation algorithm, is designed to reconstruct the original inputs from a compressed or simplified representation of these inputs. The dimensionality reduction can be achieved by reducing the number of nodes in the hidden layers [41]. Different types of autoencoder neural networks have been applied to sEMG signal processing [23][29][47], and will be explored in more detail in this section.

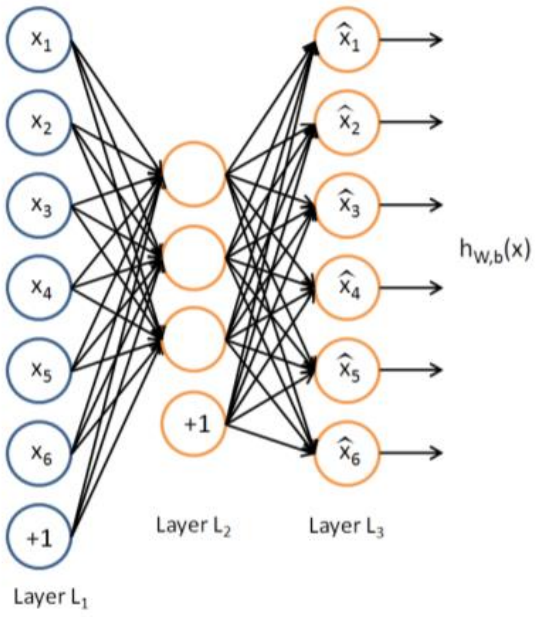

Figure 2.4, This figure from Andrew Ng [41]*, depicts a shallow feedforward autoencoder*

A stacked-sparse autoencoder is a type of semi-supervised deep learning neural network algorithm. The stacked methodology is discussed in [46], which discusses using a stacked sparse autoencoder to classify sEMG features from static hand gestures. A stacked autoencoder with reference to the methodology used in [46], is a feedforward neural network where each layer is trained as the encoded layer of an autoencoder. The autoencoders are trained sequentially, from the input layer to the output. When each of the autoencoders are trained, the layers are stacked, where the output of the previous encoding layer connects to the input of the following encoding layer. This stacked network can then be fine tuned with backpropagation training, which is the supervised portion of the training process.

The sparsity function is explained in [46][48]. The sparsity function described in [48] encourages each neuron in the hidden layer of the autoencoder to only fire or generate a high output in response to a small subset of training examples. This has the effect of introducing sparsity into the network, as most neurons will have a low output for most inputs. This means that each neuron can learn a particular feature of the input training set. One benefit of this sparsity function can be that even without a compressed hidden layer size relative to the input layer size, the network can still learn a compressed representation of the input data if such a representation exists within the data [41].

2.5.3.1. Relevant Studies for Autoencoders in sEMG signal classification

Papers such as [47] have used an autoencoder to classify sEMG gestures for wrist control functions. In Jiang et al, the autoencoder was used to classify sEMG features for the simultaneous and proportional positioning of two wrist DoFs. The DoFs of interest for this study were ulnar/radial deviation, and wrist flexion/extension, and the RMS values were collected from each of the eight channels in 100ms non-overlapping windows. This algorithm was trained with an entirely unsupervised process, with no kinematic labels recorded. An autoencoder for each DoF

was used, and contained eight input units and two hidden units, with each hidden unit representing a single DoF for the wrist. The seven participants were evaluated based on completion rate, completion time, overshoots, throughput, speed, and path efficiency. Completion rate did not show a statistically significant difference from the state of the art, completion time was lower for the AE method when compared to the state of the art, and speed of completion was also higher for the AE method. Path efficiency was also better for the AE method, and is likely due to the fact that the AE method allowed for simultaneously control of the two DoFs, while the state of the art method does not allow for simultaneous control [47]. Studies such as [46] used supervised fine tuning to train the stacked autoencoders after the initial unsupervised training processes. This second supervised training strategy could have been applied to the methods proposed in [47] to further improve classification accuracy of the AE for wrist position classification.

Further evolution of the use of a sparse auto-encoder comes from [46] for the classification of 11 discrete hand gestures. This autoencoder architecture was used with both sEMG and iEMG signals separately and was evaluated across multiple days. Six iEMG and six sEMG electrodes were used to collect data from each participant. Four-time domain features were extracted from an overlapping window of 200 ms from each EMG channel. These features were mean absolute value, waveform length, zero crossing and slope-sign change. The AE network had two autoencoders, with hidden layers of 24 and 12 respectively. These AEs were trained the SCG, and the SoftMax output layer was trained with supervised feedforward training, and then stacked with the AE encoding layers, which was then fine-tuned. The SSAE demonstrated higher performance than the LDA for both able-bodied and amputee subjects, obtaining within day error rates of 3.55% and 11.25% respectively. As in [12], the authors should have compared the use of both raw data and manually extracted features to explore the feature extraction capabilities of the proposed

classifiers. This would provide greater insight for other researchers into the feature extraction limits of the autoencoder applied in this study, and whether manual feature extraction was necessary when using autoencoders.

Both these previous papers demonstrate the applications of autoencoders for sEMG signal classification, however, both papers use manually extracted signal features to pass to the network for classification. A paper by Rehman et al compares the application of both a convolution neural network and a stacked sparse auto-encoder with raw data classification, and an SSAE trained on manually extracted sEMG features. For the feature based SSAE, two auto encoders were used, with 32 and 16 nodes in their hidden layers respectively. The SSAE for raw sEMG data contained two autoencoders with hidden layers of 100 and 50 nodes. The outputs for each of these networks was a SoftMax layer, with seven total outputs. The gestures used to train the autoencoders were a combination of hand and wrist gestures, as well as a rest gesture. For the raw SSAE, the average within session error was 25.18% across seven participants, and the feature-based SSAE achieved produced an error rate of 10.98% [12]. Although this performance is low when compared to the feature based SSAE, it indicates that the raw SSAE was able to extract some meaningful data from the raw sEMG signals and classify discrete gestures.

For a more in-depth description of the auto-encoders and their variations please see the following resources [41].

2.5.4. Summary of Contributions

Conventional classification systems like the Coapt system discussed in section 2.4.2 need to be trained on every possible movement the user would want to perform. This limits the number of functions that the user can perform with the prosthesis, as each function needs to be provided with

corresponding training data in order for the system to be able to classify the desired gesture as discussed in section 2.5.1, as retraining a wide variety of movements to replicate the intricacies of the human hand would be impractical each time. The goal of the proposed system for this thesis, as discussed in section 1, is to classify both trained and novel gestures. A system that can classify both trained and novel gestures has the potential to reduce the burden of future prosthetic devices on amputees by reducing the volume of training data that the prosthesis requires, while increasing the number of functions the system can perform. Improvements in ease of use for these prosthetics has the potential to reduce the rejection rates of these prosthetics devices, as complexity of control has been attributed to the rejection of upper-limb prosthetics devices as discussed in [49].

3. Methodology

3.1. Ethics Clearance

This study received approval by the Office of Research Ethics at the University of Waterloo (ID#: 32002)

Before the apparatus set-up could begin, each participant was provided with a health screening form to ensure that they met the required criteria. Each participant provided informed consent before the experiment began. The experimenter was always present to answer any question the participant may have had. If the participant met all the screening criteria and signed the consent form, the experimental set-up began.

3.2. Participants

Data was collected from 10 [26] able-bodied participants between the ages of 24 to 41. Eight subjects were dominant with their right hand, and two were left hand dominant. Participants were recruited from the University of Waterloo campus and consisted of students and faculty. Participants were required to be able-bodied, were screened for any allergies or sensitivities to gels, alcohol or adhesive electrodes, and were required to be between the ages of 18-55. The lower limit of 18 and the upper limit of 55 were chosen to focus solely on adult participants, excluding senior and underaged participants.

3.3. Data Acquisition

3.3.1. Data Collection Equipment

Data collection was performed by collecting SEMG data from eight differential SEMG channels using the g-tec USB Biosignal Amplifier. The eight channels were spaced evenly around the subjects' forearm, 1/3 the total distance between the wrist and the elbow, which was measured by

the experimenter using a measuring tape. The hardware used for data collection consisted of 16 disposable electrodes, with greater details outlined in section 3.4.1, connected to a single 16 channel driver interface called a GAMMAbox (Figure 3.1). The GAMMAbox was connected to a g-tec USB Biosignal Amplifier (Figure 3.2). The g-tec USB Biosignal Amplifier was connected to a desktop PC via a USB cable. The g-tec system records data at a rate of 1200 Hz. The raw sEMG data was also processed using a digital bandpass filter in MATLAB between 20 and 500 Hz [30], and a 60 Hz notch filter to remove power line noise.

Figure 3.1, g-tec 16 Channel g.GAMMAbox, depicting 2 of the 16 electrodes connected to the device

Figure 3.2, g-tec g.USBamp USB Biosignal Amplifier

3.4. Experimental Procedure

3.4.1. Electrode and Equipment Set-up

Electrode preparation took approximately 15-20 minutes. The electrode sites on the participants skin were cleansed with 70% ethanol wipes, followed by the application of conductive gel to the participants skin around the forearm electrode sites, reference site, and ground site. For some participants, it was necessary to remove excess hair from the electrode sites with a razer before the alcohol and gel were applied in order to improve electrode adhesion and reduce discomfort when removing the electrodes upon concluding the experiment.

Eight bipolar electrode channels were used, where each channel was constructed using two disposable surface EMG electrodes, one such pair is outlined in blue in Figure 3.3. The signals from each pair of channels were subtracted from one another to give a differential output. Taking

the difference between two electrodes removes the common noise between electrodes, resulting in an improved SNR [50].

The 16 recording sites were placed in two concentric rings of eight electrodes each around the forearm of the participant, where the center point between the two rings was approximately 1/3 down the length of the users forearm from the elbow, which was measured by the experimenter, Figure 3.3. One reference electrode was placed on an electrically neutral location on the olecranon, and a ground electrode was placed approximately 1/3 down the forearm from the wrist, on the ventral side of the forearm, Figure 3.3. The electrode leads connected to the g-tec GAMMABox were then connected to each of the adhesive electrodes placed around the users forearm.

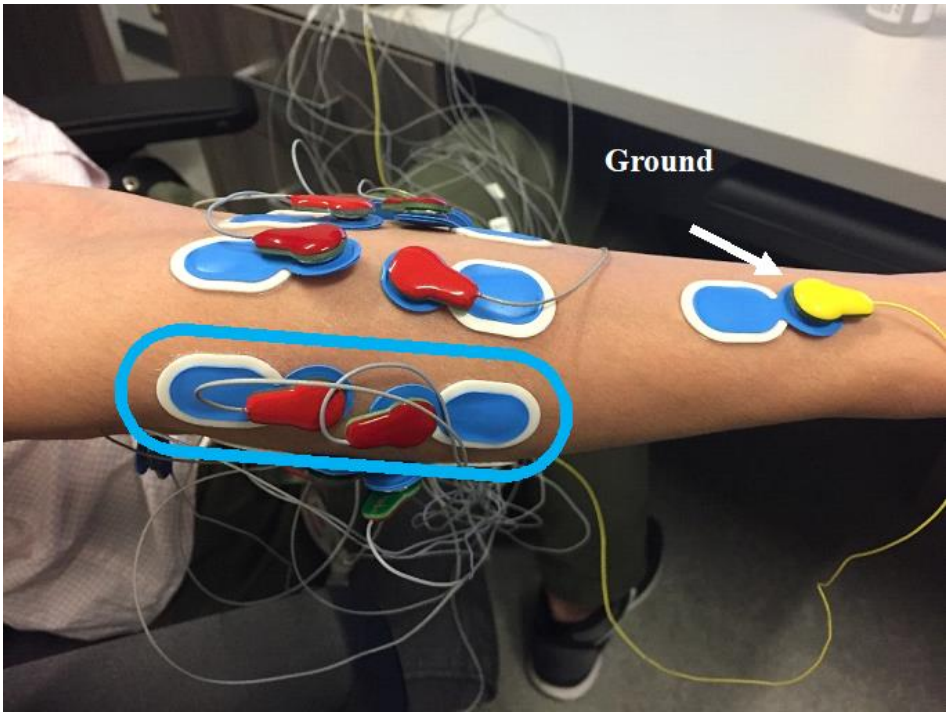

Figure 3.3, electrode placement

3.4.2. Gesture Tasks and Data Collection

Each participant was required to perform 14 isometric contraction tasks involving their dominant hand. These contractions are shown in Figure 3.4. Participants were instructed to perform the gestures by either fully flexing or extending each finger depending on the gesture. Because of this, a total of 16 gestures plus rest can be performed with the four fingers. 14 gestures including rest were used, as three gestures were determined to be uncomfortable during pilot testing. These movements were chosen so that a variety of flexion/extensions positions of the four fingers were captured. The focus of this study was on recognizing the movement of the four fingers of the human hand, and therefore movements of the thumb and wrist were omitted to simplify the proposed system. Should this system prove effective for classifying trained and novel discrete digit positions, the classification of additional DoFs from the thumb and wrist would be added in future research. The participants were instructed to hold their thumb and wrist in a constant position for all contractions. The contractions were to be held at an intensity that the user could maintain for an extended period without discomfort or tremors in their hand.

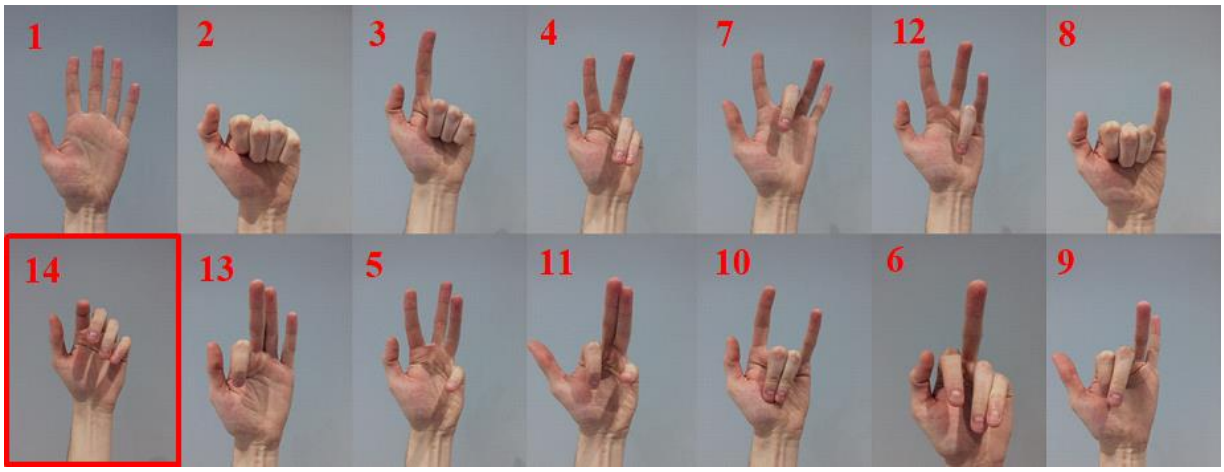

Figure 3.4, 14 gestures collected for this study. The rest gesture is outlined in red.

There were two rounds of data collection where each round consisted of the same 14 gestures. Twenty seconds of data [31] was collected for each gesture in each round. The order in which the gestures were presented to the participant was randomized by the collection interface. In total, nine minutes and 20 seconds of data was collected from each subject. The number of collection rounds used in data collection varies in other studies, from as few as one in [31], and as many as 6 in [51]. The length of the data collection periods for each isometric contraction employed in other studies also varies widely, with some studies using only three seconds per round [2], and others such as [31], collecting 20 seconds per round.

For the proposed study, in between each 20 second isometric contraction, the participant was given a 10 second break [52]. After the first round, each participant was given a 10-minute break. The 10-minute inter-round relaxation period was chosen from pilot testing and was determined with anecdotal and qualitative reporting of the degree of discomfort felt by the pilot subject, and the rest time required by said pilot test subject to recover. No fatigue was reported by any participants at the end of the collection rounds. A total of 672,000 sEMG samples per channel were collected, for a total of 5,376,000 samples across all eight channels for each participant.

Since the algorithms featured in the study are relevant for the development of commercial prosthetic arm control systems, the data collection procedure should mirror the procedure used in comparable commercially available prosthetic devices, while still providing the algorithms with a suitable volume of data for training and testing. For example, the Coapt Complete Control system uses two sets of data collection for each gesture, each lasting three seconds [20]. One study by Englehart et al, used two rounds of collection for each gesture with five seconds of data collected per gesture to provide both training and testing data for an LDA gesture classifier [2]. Considering these parameters of two rounds of collection for isometric hand gestures is seen in both academic

research, and commercial prosthetics, this indicates that it is suitable for the proposed experiment, which is meant to serve as both an academic study, and the foundation for the development of control algorithms with industry facing applications.

3.4.3. Data Collection Interface

A MATLAB interface was used to record the data for each of the 14 gestures. The MATLAB interface prompted the participant to perform the correct gesture with an image of the corresponding gesture. The experimenter was present during data collection to provide additional guidance when needed. The interface prompted the user when they were required to relax or perform an isometric contraction. A green progression bar at the bottom of the screen, as shown in (Figure 3.5) indicated the non-recording period, where the participant could relax. A red progression bar (Figure 3.6) indicated when the participant was required to perform the displayed isometric contraction, or rest gesture. A count-down timer was also displayed on screen (Figure 3.5 and Figure 3.6) along with the progression bar during the rest period and during the active period, so that the participant knew when to start and stop a contraction.

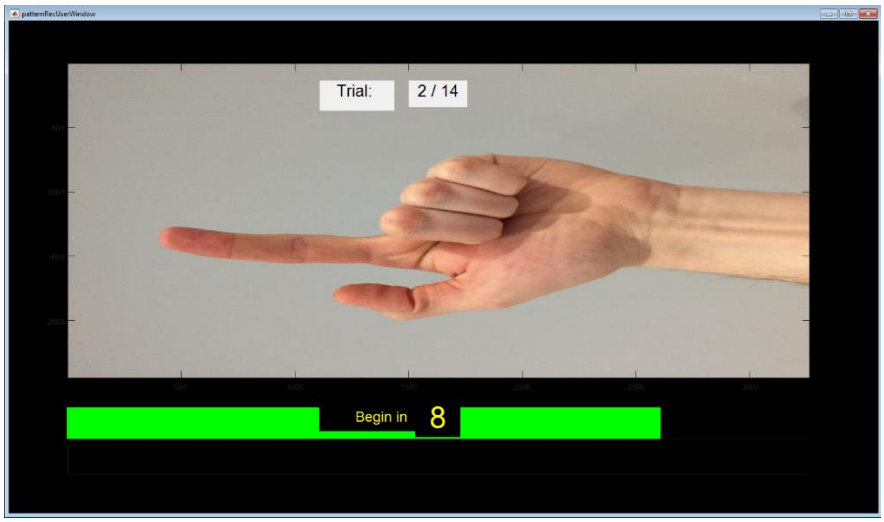

Figure 3.5, relax period before isometric contraction

Figure 3.6, Isometric Contraction Period

3.4.4. Data Segmentation and Preprocessing

The machine learning algorithms that were implemented in this study were a Stacked Sparse Auto-Encoder (SSAE), a Feed Forward Multi-Layer Perceptron (FFMLP) network, and a Convolution Neural Network (CNN). These networks were constructed using the Deep Learning Toolbox in MATLAB 2018b which provided an acceptable degree of customization for the networks in this study, with a very shallow learning curve. MATLAB was also used for the development of artificial neural networks in studies such as [53].

As described in [2], the method with which data is segmented or tiled is dependent on the features used for classification. For instance, methods such as Short-Term Fourier Transform (STFT) utilize a fixed tiling or window size, whereas the Wavelet Transform (WT) uses variable tiling, which changes the aspect ratio of the cells in order to ensure that the frequency resolution is proportional to the median frequency. The studies outlined in section 2.5 used a fixed window size, therefore the segmentation pre-processing also used a fixed window, which was determined through pilot data testing.

The use of automated feature extraction to isolate useful information from raw sEMG data effectively turns the feature extraction sub system into a “black box”. In other words, it is not clear exactly what features the algorithms are extracting from the signal and using for classification. Therefore, certain aspects of data segmentation such as the optimal degree of overlap between windows and the size of the windows were determined using pilot data testing. The window step sizes, and window size were determined through pilot data testing for this study. Majority voting was also considered for output post-processing as explored by [2]. However, although this has been proven to reduce the variation in the output decision stream, improving the applicability for prosthetic control, this does not allow us to evaluate the raw output of the algorithms to determine the network classification accuracy without post-processing. Therefore, majority voting will not be used in data post processing but can be applied in the future to improve the decision accuracy of the output decision stream for prosthetic control applications.

3.4.4.1. Data Segmentation for Deep MLP and SSAE

It was determined through testing that a window size of 25ms was suitable for the MLP and Autoencoders, which results a 240x1 input vector to both networks. Step sizes between 1ms and 25 ms were tested to determine the optimal step size. The step used to generate the input data was 5 ms. The windows were tested from 10-200 ms. Larger window sizes were tested using pilot data, but larger window sizes demonstrated increasingly poor performance. Pilot results for different tested window sizes from subject 08 for gesture combination 14 can be seen in Table 7.2 in section 8.

3.4.4.2. Data Segmentation for CNN

The raw data from the g-tec USB Bio Signal Amplifier is processed using MATLAB. The data was first segmented into windows of dimension 240x8 samples. The window is 200 ms of data

(240 data points) from each of the eight differential SEMG channels. The window step size is five ms, or a six data point step per channel. The dimension for these windows and the step was determined through pilot data testing.

3.4.5. SEMG Data Classification

3.4.5.1. Output Architecture and Labelling

The three algorithms tested for this study all utilized the same output design. This design was inspired by several studies [26][25]. Au et al was focused on using EMG signals to classify shoulder movements using a feedforward MLP with an output design where each output node represented the angle of a single DoF in the shoulder/arm. The study by Jiang et al. used an output design, where each digit activity was represented using two output nodes each. One output node represented the flexion state of a digit, and the other represented the extension state of the digit, for eight outputs in total [26]. A different architecture was adopted compared to the two previously discussed studies [26][25]. The output in this study uses two nodes per digit. The 'Active' node represents the finger position output, where one is digit extension, and zero is digit flexion. The second node is the 'Rest' node, with a label of one indicating that there is no activity for this digit. The rest node is necessary for each digit, as the 'Active' node is always in some active state, either zero, or one. Without an additional output in the form of the 'Rest' node to tell a digit in a prosthesis to do nothing, the digit will only be taking commands from the active node. In other words, these 'active' nodes produce a constant stream of decisions, and there needs to be some way of gating the output. This was achieved in this algorithm through the inclusion of a 'Rest' node for each digit. This output design is as vital as classification accuracy for the system to be usable for prosthetic control [2]. The diagram in Figure 3.7 depicts the outputs for the open hand gesture, where all four fingers are extended.

The architecture used in this study allows the rest state to be classified separately from the active state. Whereas with the architecture proposed in [25], the classification of a rest state would be directly dependant on the state of the flexion and extension nodes. This ability to gate the active node output with the rest node output is beneficial for prosthetic control. If a user wants to initiate a movement with their prosthetic hand, and the classifier outputs a rest command, it is easy for the user to perform the gesture again. Alternatively, if the user wants their prosthesis to hold a resting or neutral position while their prosthetic arm is holding an object, and the resulting classification is an active motion, the prosthetic may drop the object [44]. An architecture with a rest node that overrides the active node reduces that likelihood that the latter scenario will occur, as the system can be made to override any active node outputs, if the rest node output is also high.

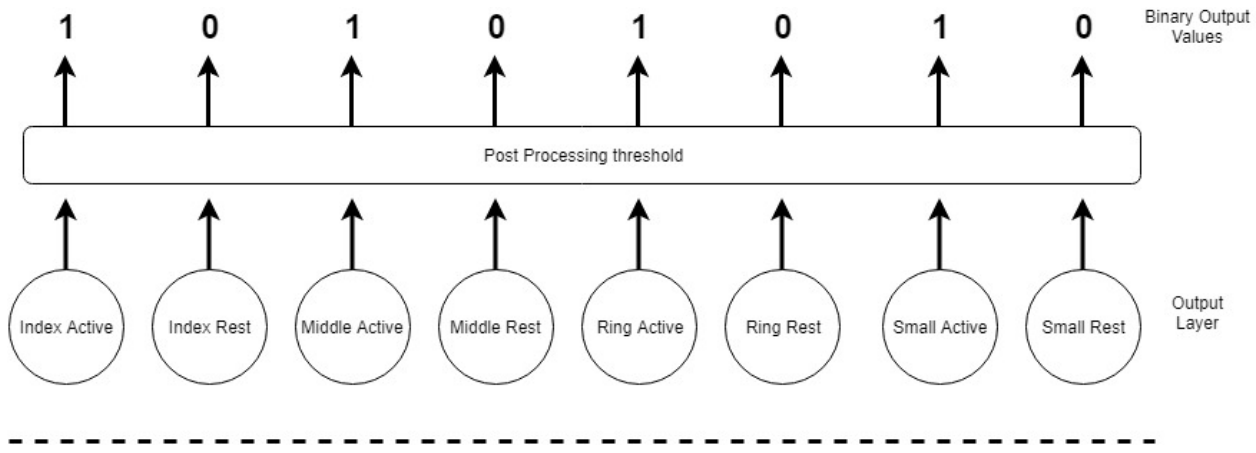

Figure 3.7, depicted here is the output architecture used across all three types of networks investigated in this study

The values produced by each output node in the network are not necessarily binary values, but instead are continuous. Therefore, to transform each of the raw outputs from the network into a binary value, a threshold was applied at the output for each node, where an output below 0.5 constituted a zero, and any value above 0.5 was a one, seen in Figure 3.7. This output architecture allowed the network to classify sEMG signals from gestures that the system had not been trained

on, as the network theoretically has the potential to produce novel combinations of finger movements in response to novel patterns of sEMG signals.

3.4.5.2. Gesture Combinations

The objective of this study was to determine if the networks could correctly classify gestures it had never been trained on before, and therefore not all gestures were included in the training data gesture combination. With a total of 14 gestures including rest, the number of possible training/testing configurations is 2^{14} unique gesture combinations. Therefore, it would have been impractical to test all the possible gesture combinations. This is because each gesture occupies one of two states, these being the novel gestures or untrained gestures. Trained gestures were used to train and test the network, whereas novel gestures were only used to test the network.

The gestures combinations that were used can be described as 'balanced' gestures combinations. This means the gesture combinations all contained an equal number of flexion and extension states for each finger in the training subset within each combination. For example, there were a total of six unique gestures that had the index finger in a flexion position, and seven where the index was extended. In any distribution of these combinations between novel and trained gestures, there must be an equal number of gestures with the index extended and index flexed in the training set. This method was used to reduce the chance that the trained network might be biased towards either flexion or extension labels for a given digit. Each gesture combination contains six gestures plus rest for training and testing, and the seven other gestures for testing only. This number of gestures was determined as the minimum in accordance with other papers such as [12] which trained six gestures plus a rest. Therefore, a total of 20 gesture combinations were tested for each participant and algorithm, and the composition of these gesture combinations can be seen in Table 7.1 in section 8.

3.4.5.3. Feed-Forward Multi-Layer Perceptron

The number of hidden layers tested varied between 1-5 hidden layers in the network, and 10-100 nodes for each hidden layer. The optimal number of layers, as well as the number of nodes per layer was determined through pilot data testing. The type of activation function for the layers was also determined through testing, where the two functions tested were a Sigmoid Function and a Hyperbolic Tangent Sigmoid Function (HTSF). The HTSF performed better than the network using Sigmoid.

For training, resilient back propagation was confirmed to have the best performance when compared to scaled conjugate gradient, which was used in [25], and gradient descent, which was compared to resilient back propagation in [54]. Compared to gradient descent, resilient back propagation is beneficial as it is less susceptible to local minima, as only the sign of the error gradient is considered when adjusting the weights, while gradient descent considers the sign and value of this gradient error. [54] states that resilient back propagation can reach a convergence faster than gradient descent, making it more applicable for applications in commercial prosthetics, where longer training times can interfere with a users' operation of their prosthesis.

A detailed description of the Deep MLP architecture used can be seen in Table 2.1. All other parameters were left in the default settings for the 'feedforwardnet' function in MATLAB 2018b.

Table 3.1, this table depicts the architecture, input window size, and training/testing data distribution for the FFMLP.

Feed-Forward Deep Multi-Layer Perceptron	
Parameter	**Value**
Training Algorithm	Resilient Backpropagation
Outputs	8 nodes
Hidden Layer Activation Function	Hyperbolic Tangent Sigmoid
Hidden Layer Architecture	1. 100 nodes 2. 60 nodes 3. 40 nodes
Output Activation Function	Logarithmic Sigmoid
Input Size (milliseconds, samples)	25ms, 240 samples
Number of Epochs	500
Number of Validation Checks	10
Data Distribution	Training: 70% Validation: 15% Testing: 15% [30]

3.4.5.4. Stacked Sparse Auto-Encoder

The time delay input for the Stacked Sparse Autoencoder (SSAE) is identical to the input vector used in the FFMLP. The initial process of training each layer in the SSAE was performed without the need to provide the network with corresponding labels for training data, and was unsupervised [55]. Each layer was trained as a shallow autoencoder, with the inputs and outputs of each autoencoder being equal to the outputs of the previously trained encoder layer. The trained encoder layers of each autoencoder were then stacked into a feedforward MLP, and fine tuned using backpropagation [12].

The stacked sparse autoencoder used in this study had the same number of layers and the same number of hidden nodes per layer as the FFMLP constructed for this study, discussed in the previous section. This method was chosen to compare the semi-supervised method of the SSAE

with the supervised training of the MLP, within the constraints of MATLABs Deep Learning Toolbox. A detailed description of the SSAE used for this study can be seen in Table 2.2 and Table 2.3. The other parameters used, including the L2 Regularization, Sparsity Regularization, and sparsity Proportion were all applied in accordance with [12].

Table 3.2, this table shows the network architecture and hyperparameters used for the individual autoencoders. Each encoder layer number depicted in the table below is the hidden layer size for each of the four autoencoders, and the input/output sizes are the number of inputs/outputs that were used to train each of the respective encoder layers.

Layer-wise Autoencoder	
Parameter	**Value**
Training Algorithm	Scaled Conjugate Gradient
Encoder Activation Function	Logarithmic Sigmoid [12]
Encoder Layers	100 nodes 60 nodes 40 nodes 8 nodes
Decoder Activation Function	Pure line function [12]
Input/Output Sizes	Autoencoder 1: 240 Autoencoder 2: 100 Autoencoder 3: 60 Autoencoder 4: 40
Number of Epochs	500 [12]
L2 Regularization	0.0001 [12]
Sparsity Regularization	0.01 [12]
Sparsity Proportion	0.5 [12]
Data Distribution	Training: 70%

Table 3.3, the architecture for the stacked autoencoder. The hidden layers of the stacked network are the trained encoder layers depicted in the previous table. While the training process for the individual autoencoders was unsupervised, the training process for the stacked autoencoder was supervised. Therefore, the training/testing data distribution hyperparameters are provided in this table as well.

Stacked Sparse Autoencoder	
Parameter	**Value**
Training Algorithm	Scaled Conjugate Gradient
Outputs	8 nodes
Hidden Layer Activation Function	Logarithmic Sigmoid
Hidden Layer Architecture	1. 100 nodes 2. 60 nodes 3. 40 nodes
Output Activation Function	Logarithmic Sigmoid
Input Size (milliseconds, samples)	25ms, 240 samples
Number of Epochs for stacked training (supervised)	500 [12]
Data Distribution	Training: 70% Testing: 15%

3.4.5.5. Convolution Neural Net Architecture

Papers such as [40] and [31] focus on the applications of CNNs for the classification of EMG signals based on discrete gesture classes. However, for this type of algorithm to be able to classify every combination of discrete flexed/extended finger position, it would require each combination to be manually programmed into the network as an output label. Within the limits of the authors literature review, it did not appear that any paper had been published which uses the proposed output architecture in a CNN, nor had one been developed to test new untrained gestures.

The number of fully connected layers was taken from [40], though smaller fully connected layer architectures were also tested, but did not improve performance. Both 32 filters and 64 filters both of size 3x3 from [12] and [40] respectively were both tested. From the tests performed, there was no significant difference between the two filter sizes, so 32 filters were used in both layers to reduce the training time. After each convolution layer, a ReLu layer, batch normalization layer,

and max pooling layer are included in accordance with [12]. The fully connected layers were derived from [14], though as mentioned previously, a simpler set of fully-connected layers was tested with not significant improvements in performance.

The two options for outputs in the MATLAB Deep Learning Toolbox are; a Regression layer, or a Softmax Layer. The regression layer was selected as the output as it allows for outputs to be active simultaneous. A detailed description of this CNN can be seen in Table 2.4.

Table 3.4, this table depicts the final architecture, input window size, and training/testing data distribution for the Convolution Neural Network

Convolution Neural Network	
Parameter	**Value**
Training Algorithm	SGDM (Stochastic Gradient Descent with Momentum)
Outputs	8 nodes
Hidden Layer Architecture	1. Convolution Layer 1 (32, 3x3 filters) 2. ReLu Layer 3. Batch normalization 4. Max Pooling Layer 1 5. Convolution Layer 2 (32, 3x3 filters) 6. ReLu Layer 7. Batch Normalization 8. Pooling Layer 2 9. Fully Connected Layers a. 516 nodes b. 516 nodes c. 128 nodes d. 8 nodes 10. Regression Layer
Input Size (milliseconds, samples)	200ms, 240x8 samples
Batch Size	256
L2 Regularization	0.0001
Momentum	0.95
Segmentation	70% Training, 15% Testing

These networks were chosen for their demonstrated ability in other studies [40][12][25], to be used for automated feature extraction from raw sEMG data as discussed in section 2.5. For the CNN, other parameters were also tested including the size of the fully connected layers, introducing dropout into the fully connected layers, the number of kernels or filters in the convolution layers, and varying the batch size were also tested.

3.5. Performance Evaluation

Performance validation methods varied in the papers explored depending on the output architecture of the networks used to classify SEMG signals. Networks with continuous outputs such as the one in [25], used root mean squared error (RMSE) as a measure of performance. In [40], the performance is determined based on the percentage of correct gesture decisions made by the network during testing [28].

3.5.1. All-or-Nothing Gesture Classification

Two different gesture classification analysis methods were applied to the classifier decisions for each algorithm. The All-or-Nothing Gesture Classification method (ANGC) works by reviewing all classification decisions for a given input window of raw sEMG data. If a single digit is misclassified with respect to the target digit positions for said input, then the entire gesture is misclassified, regardless of whether the other digits in the gesture were correctly classified. This method was meant to be comparable to the discrete label classification method often applied in machine learning classification problems. However, this ANGC method ignores whether the classification of the other digit positions was correct with respect to the target gesture, and therefore, is not truly representative of the actual performance of the classifier in predicting finger positions from the isometric gesture that the sEMG signals are representative of. Therefore, a second analysis method was envisioned that more accurately represents the applicability of the classifier for use in prosthetic control.

3.5.2. Weighted Digit Gesture Classification

The second method, referred to as Weighted Digit Gesture Classification (WDGC), takes the weighted sum of all correct digit classifications, where each digit is weighted based on its relative importance for manipulation tasks with a trans-radial prosthesis. This second classification

analysis method is more representative of the actual classification accuracy and usefulness of these algorithms for applications in prosthetic control and provides a more complete picture of the classification accuracy of a given algorithm. Weightings for each digit are multiplied by one or zero depending on whether the digit was correctly classified or not.

The weighting for each digit was determined based on whether a digit for a given gesture was involved in grasping or manipulation for a given gesture. If a digit was used for grasping or manipulation for a given gesture, then a vote would be assigned a value of one for the digit, for the gesture in question, otherwise the digit was assigned a value of zero. These values for each digit are summed across all gestures, then the weighting of each is determined based on the summed value for this gesture divided by the sum of the total number of values for all gestures and digits used in the analysis. The value of each digit for each gesture, as well as the final relative weighting for each digit is shown in section 8, Table 7.3. These weightings are 39.39%, 24.24% for the index and middle fingers, and 18.18% for both the ring and little fingers. These gestures were selected from the gestures available for the Bebionic 3 prosthesis developed by Ottobock.

The images of the 13 grip patterns used to determine the relative weights are depicted in Figure 8.1 in the Appendix in section 8, and a description of these grip patterns can be found in the Bebionic User Guide [17]. The results of the WDGC analysis method not only serves as a more realistic means of determining how effectively a classifier can recognize common sEMG patterns for particular movements across trained and novel gestures, but it also demonstrates why the method of classifying finger positions is superior to methods that only classify discrete gestures. This is because even if the algorithm misclassifies a gesture, at least with the CNN, the system can still be used for manipulation tasks, as often no more than 1 digit was misclassified as will be

discussed in section 4. This would make a prosthesis much more forgiving if the user makes an error in initiating a control signal with sEMG.

3.6. Statistical Analysis of Results

ANOVA mixed effects model was used to evaluate the effects of multiple variables on the collected results. The variables considered were the subject, gesture combination, and the algorithm used. The subject was considered a random factor, as was the gesture combination. The gesture combination was a random factor, as only 20 out of 16,384 possible combinations were tested for the purpose of this study. The algorithm was the fixed factor. The tests determine whether the gesture combination and the subject influenced the performance produced by each algorithm. The significance of performance between the algorithms were evaluated with a Tukey test. This methodology was applied to the results from ANGC and WDGC analysis methods for both trained and untrained gestures. One-way ANOVA was used to evaluate the significance of the performance differences between the selected classification algorithms.

4. Results

This section outlines the results for all three algorithms for the classification of individual digits, as well as the performance for both the ANGC and WDGC performance analysis methods. This section is divided into sub sections based on the algorithm in question, and each sub section is further divided into sections for digit, ANGC or WDGC results. The average performance across all combinations and participants, the performance for GC 14 across all participants, and the performance of the top performing participant for GC 14 will all be presented. GC 14 was chosen as this was the best performing gesture combination for the CNN for untrained gestures. The statistical results for the influence of participants and gesture combinations are discussed last. The graphs and figures for the performance of certain individual participants for each algorithm are included in the Appendi.

4.1. Participant Data

Table 4.1, participant demographics

Mean Reported Age	**Male**	**Female**	**Lefthanded**	**Righthanded**
27.7 years	8	2	2	8

4.2. FFMLP Results

4.2.1. FFMLP Performance for Digit Classification

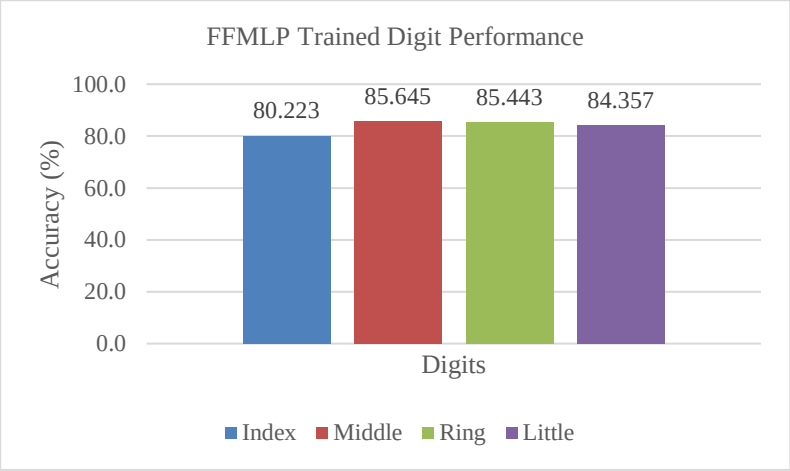

Figure 4.1, this figure depicts the FFMLPs classification accuracy for the discrete position of each finger for trained gestures

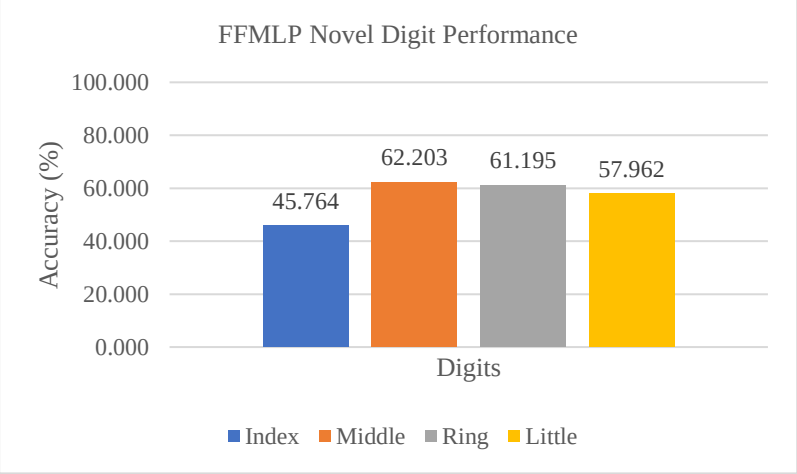

Figure 4.2, this figure depicts the FFMLPs classification accuracy for the discrete position of each finger for novel gestures

The FFMLP was able to predict the discrete position of individual fingers, shown in Figure 4.1 across all participants and gesture combinations for trained gestures with reasonable accuracy. The FFMLP was able to achieve performance of greater than 80% for each of the four digits. The middle and ring fingers were the top performing digits for novel gestures. For the classification of

finger positions from novel gestures in Figure 4.2, the index finger achieved sub 50% performance, and the other digits achieved low classification.

4.2.2. FFMLP Performance with ANGC Method

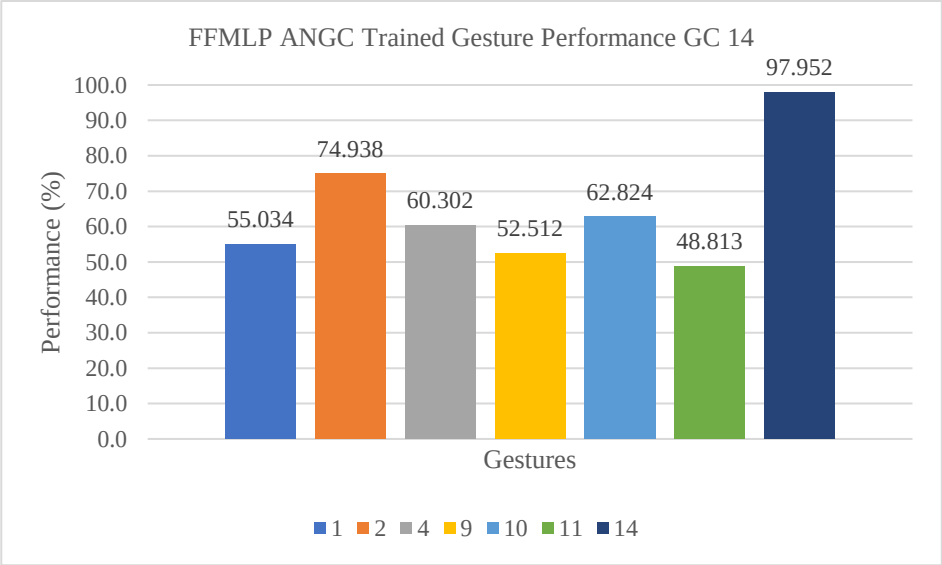

Figure 4.3, this figure depicts the FFMLPs classification performance for trained gesture combination 16 averages across all participants using the ANGC method

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	55.04	1.63	6.60	6.13	5.05	9.93	**1111**
	2	2.25	77.42	3.78	2.61	3.63	2.86	**0000**
	4	6.58	3.75	60.33	3.05	4.04	6.57	**1100**
	9	5.42	1.66	2.20	52.52	8.28	10.66	**0011**
	10	5.28	3.70	4.47	8.19	62.92	3.38	**1001**
	11	7.21	1.88	7.14	9.73	1.28	48.83	**0110**
	13	0.45	2.43	0.26	2.86	3.51	0.77	**0111**
	6	0.51	1.90	2.23	0.65	0.26	1.80	**0100**
	13	4.70	0.40	1.20	5.85	0.70	5.22	**0111**
	3	0.73	2.16	1.74	0.28	2.15	0.39	**1000**
	7	2.73	0.29	0.37	3.66	4.12	0.78	**1011**
	12	3.07	0.45	3.34	0.64	3.02	0.74	**1101**
	5	5.53	0.49	5.76	1.41	0.58	5.53	**1110**
	N	0.49	1.82	0.56	2.37	0.43	2.50	**0010**
	N	0.00	0.01	0.00	0.03	0.01	0.03	**0101**
	N	0.01	0.01	0.02	0.01	0.01	0.02	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 4.4, the confusion matrix for gesture combination C14 for the active nodes in the FFMLP output averaged across all participants for trained gesture classification. The base-10 numbers along the top and left sides correspond to the gestures in Figure 3.4. The binary numbers on the right side and bottom of the matrix represent the position of each finger in the gesture, with 1 referring to finger extended, and 0 being finger flexed. Reading from left to right, the order is index, middle, ring and little finger. The green cells contain the classification accuracy for correctly classified gestures in accordance with the ANGC method. The other cells represent the percentage of times that one or more digits have been misclassified.

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	4.18	0.79	1.34	0.92	2.28	2.77	0.44	**1000**
	5	1.98	5.60	6.02	2.29	0.49	4.83	5.45	**1110**
	6	1.60	1.08	3.16	1.14	0.59	2.12	0.89	**0100**
	7	0.72	2.03	0.86	3.69	4.38	1.72	2.17	**1011**
	8	1.68	0.93	0.94	2.39	4.77	1.52	1.28	**0001**
	12	4.60	2.54	3.39	1.76	1.44	3.93	1.39	**1101**
	13	1.21	5.87	2.00	4.30	0.99	2.19	7.10	**0111**
	2	14.22	2.54	5.47	3.14	7.36	5.89	2.40	**0000**
	9	5.81	11.52	5.93	35.07	24.21	8.33	20.40	**0011**
	11	4.06	23.62	17.88	14.08	2.51	10.02	19.61	**0110**
	10	19.57	7.94	5.74	13.80	41.53	16.42	5.50	**1001**
	4	33.28	11.22	37.30	7.92	4.50	29.84	6.60	**1100**
	1	5.50	22.46	8.28	7.14	3.33	9.01	24.85	**1111**
	N	1.55	1.82	1.65	2.31	1.59	1.38	1.88	**0010**
	N	0.02	0.03	0.02	0.03	0.01	0.03	0.03	**0101**
	N	0.02	0.01	0.00	0.02	0.00	0.02	0.01	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 4.5, the confusion matrix for gesture combination C14 for the active nodes in the FFMLP output averaged across all participants for novel gesture classification.

Using the ANGC method, the FFMLP demonstrated an ability to classify trained gestures, whose performance was significantly different from random chance. This can be seen in Figure 4.3, which shows the performance for each gesture in the top performing gesture combination, GC 14. For trained gestures we can observe that there were many instances where multiple digits were simultaneously misclassified, (Figure 4.4) though with fewer overall misclassifications when compared to the novel gestures. For instance, with target gesture 11, both the index and little fingers were misclassified 9.93%. For target gesture 1, the multi digit misclassifications occurred with ring/little fingers, index/middle, and middle/ring at rates of 6.58%, 5.42%, and 5.42% respectively.

The highest novel gesture classification performance achieved by the FFMLP was for subject 03, GC 7, gesture 9, which was 25.5%. While this novel gesture performed better than random chance, it is not sufficient for prosthetic control applications.

From Figure 4.5, which depicts the FFMLP output decisions for novel gestures, we can see that when a novel gesture was misclassified as per the ANGC method, it appears that the index, middle and little fingers contributed to the majority of misclassifications. For target output for gesture 3, only one digit was misclassified. 33.28% misclassification was due to the middle finger, 19.57% due to the little finger, and 14.22% was due to the index finger misclassification.

What can also be observed is that the classifications of the active nodes account for all misclassifications for the novel gestures, meaning any misclassification of the rest nodes directly overlapped with the misclassification of the active nodes.

The classification results for the subject 8 are shown in Figure 8.3, in Appendi 8.1. For the performance subject 08 with GC 14, a high volume of the misclassifications resulted from multiple digits being simultaneously misclassified, as with the average performance across all participants discussed previously. For example, the index/little fingers were misclassified for gesture 1 23.35% of the time. Gesture 11 also had a majority of the misclassifications for this gesture result from simultaneous digit position errors as well, with index and little being misclassified simultaneously at a rate of 26.69%.

4.2.3. FFMLP Performance for Gestures with WDGC Method

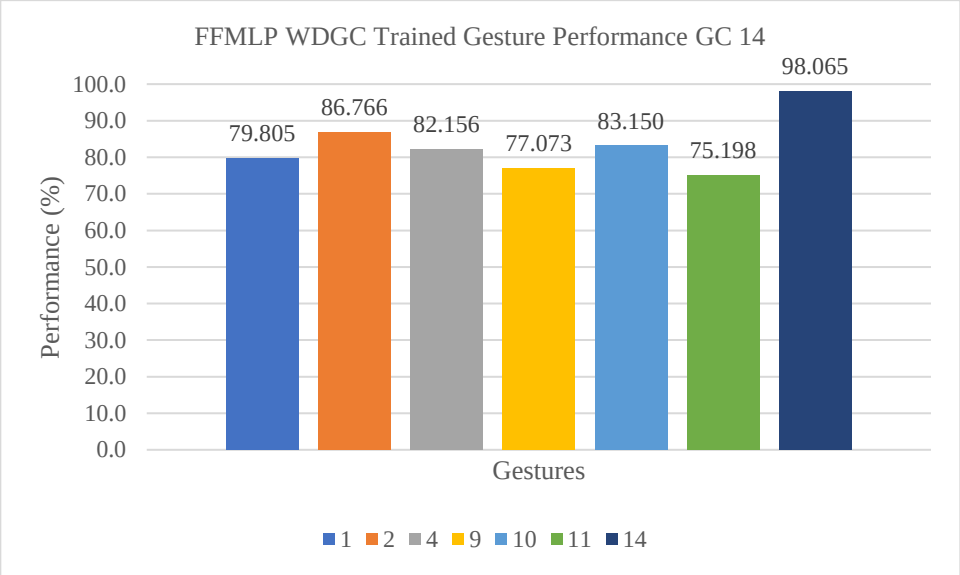

Figure 4.6, this figure depicts the FFMLPs classification performance for trained gesture combination 14 averages across all participants using the WDGC method

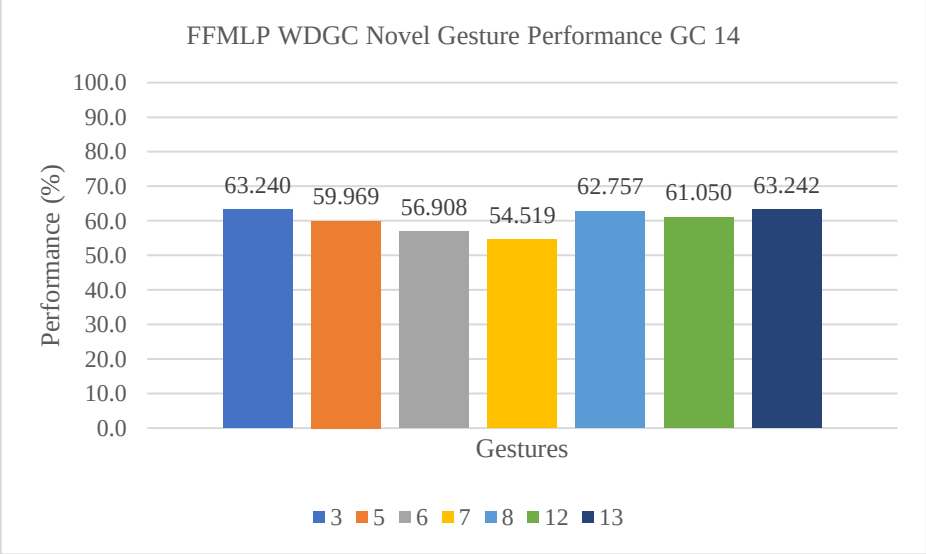

Figure 4.7, this figure depicts the FFMLPs classification performance for novel gesture combination 14 averages across all participants using the WDGC method

Both trained and novel gestures performance from the WDGC method demonstrated a significant improvement in performance over the trained and novel gesture performance from the ANGC

method. The average performance for trained gestures in gesture combination 14 for the WDGC method can be seen in Figure 4.6. The results for novel gesture classification using WDGC method can be seen in Figure 4.7.

The confusion matrices for this combination in Figure 8.3 and Figure 8.4 show the degree to which these gestures were misclassified, as well as the specific digits that contributed to misclassifications.

The performance for trained and novel gestures in GC 14 for subject 8 is shown in Figure 8.5 and Figure 8.6. The top performing trained gesture in this combination, gesture three, had a performance of 79.75%. This gesture corresponds to index extended. For trained gesture 7, the performance was the lowest for subject 8, GC 14 for WDGC method. The source of this low classification rate becomes apparent when reviewing Figure 8.4. Because the index finger has the highest weighting, 39.39%, any misclassification of the index finger has a greater impact on the WDGC performance than the misclassification of any other finger. Therefore, it can be concluded that the reason for the low performance of gesture 7 is because the index finger was involved in most misclassifications for this gesture. For example, the index finger was involved in 37.81%, and the index and little fingers together were involved in 25.58% of misclassifications. Most of the misclassifications for gesture 3 were due to the little finger being misclassified 67.75% of the time, and the middle finger being misclassified 23.31% of the time. This resulted in a higher overall performance of 79.75%, as the middle and little fingers were determined to be less critical for prosthetic use when compared to the index.

4.3. SSAE Results

4.3.1. SSAE Performance for Digit Classification

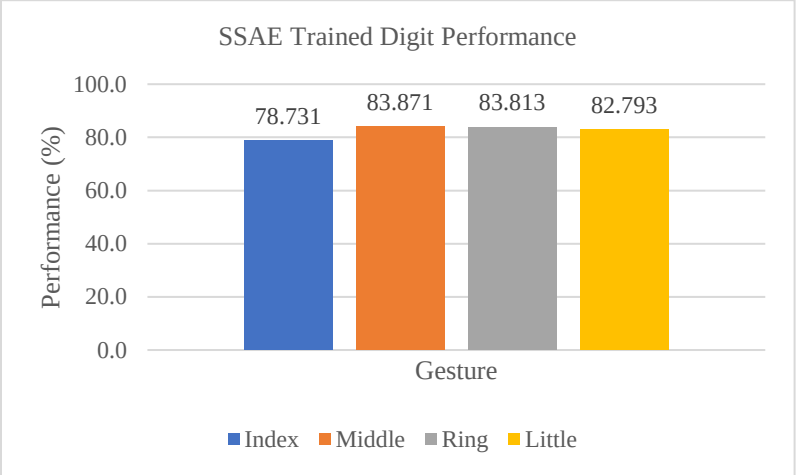

Figure 4.8, this figure depicts the SSAE classification accuracy for the discrete position of each finger for trained gestures

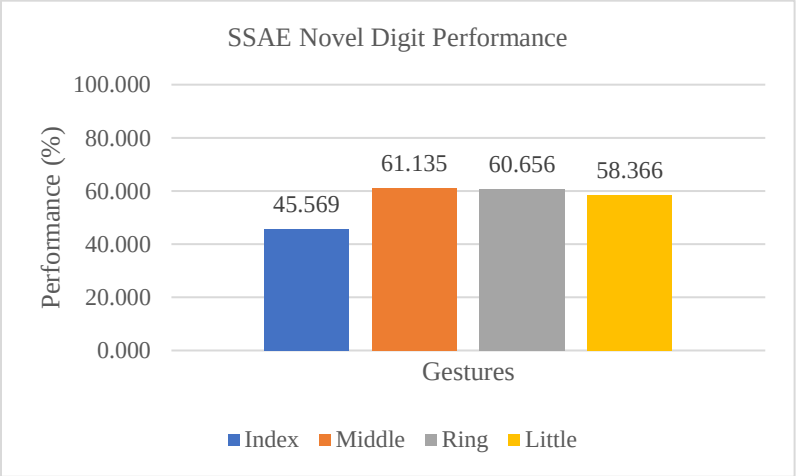

Figure 4.9, this figure depicts the SSAEs classification accuracy for the discrete position of each finger for novel gestures

Figure 4.8 and Figure 4.9 show the overall average classification accuracy for each of the four digits for trained and novel gestures respectively. The index finger was the lowest performing digit for trained gestures, with a position classification accuracy of 78.7%. As with the FFMLP, the

index finger for the novel gesture classification results generated by the SSAE were below 50% for novel gestures.

4.3.2. SSAE Performance with ANGC Method

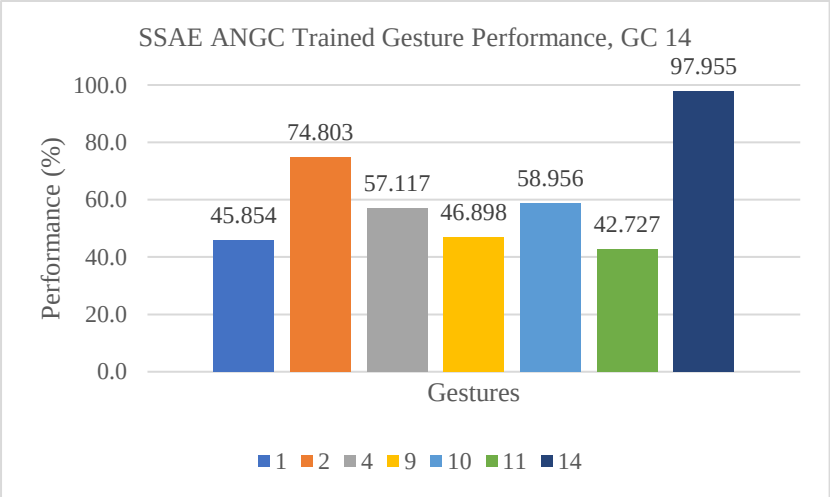

Figure 4.10, this figure depicts the SSAEs classification accuracy for the discrete position of each finger for trained gestures

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	45.85	1.73	7.27	7.69	4.96	11.63	**1111**
	2	2.61	76.29	4.97	2.46	3.78	2.23	**0000**
	4	8.14	6.02	57.12	2.81	5.58	7.91	**1100**
	9	7.78	1.79	2.13	46.92	10.67	12.18	**0011**
	10	5.92	3.76	5.50	10.08	58.99	4.11	**1001**
	11	7.93	1.98	8.03	11.43	2.02	42.73	**0110**
	13	0.79	1.84	0.36	2.28	3.02	0.63	**0111**
	6	0.56	1.87	1.99	0.35	0.23	1.26	**0100**
	13	7.68	0.28	0.94	7.98	0.74	8.29	**0111**
	3	0.48	1.68	1.43	0.26	1.38	0.19	**1000**
	7	3.12	0.48	0.52	4.18	4.41	1.09	**1011**
	12	3.44	0.75	3.45	0.62	3.16	0.68	**1101**
	5	5.09	0.56	5.85	1.24	0.62	5.14	**1110**
	N	0.60	0.98	0.41	1.71	0.43	1.92	**0010**
	N	0.01	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.01	0.00	0.01	0.00	0.02	0.00	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 4.11, the confusion matrix for GC 14 for the active nodes in the SSAE for trained gestures across all participants

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	3.93	0.42	1.20	0.34	1.53	1.82	0.33	**1000**
	5	1.71	6.53	6.83	1.64	0.54	5.61	4.31	**1110**
	6	1.18	0.95	2.18	0.51	0.26	1.03	0.76	**0100**
	7	1.14	2.18	0.97	4.28	4.39	2.53	2.87	**1011**
	8	1.35	0.68	0.76	2.20	3.79	1.13	1.06	**0001**
	12	4.16	2.09	2.96	1.46	1.60	3.67	1.82	**1101**
	13	0.96	8.23	2.10	7.00	1.80	2.42	10.69	**0111**
	2	23.00	2.43	4.75	2.59	6.66	5.05	2.55	**0000**
	9	2.74	10.58	5.51	35.31	25.59	7.97	18.79	**0011**
	11	2.87	25.03	15.22	12.80	2.90	9.18	16.13	**0110**
	10	14.81	8.03	8.98	15.43	41.04	20.04	7.18	**1001**
	4	35.42	12.60	38.74	5.79	4.99	26.77	6.69	**1100**
	1	6.15	18.98	8.83	8.99	4.03	12.05	25.37	**1111**
	N	0.57	1.25	0.97	1.66	0.88	0.72	1.44	**0010**
	N	0.01	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.01	0.00	0.01	0.01	0.00	0.01	0.00	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 4.12, the confusion matrix for GC 14 for the active nodes in the SSAE for novel gestures across all participants

The trained gesture classification performance with the ANGC method, as expected, exceeded the performance of the untrained gestures and is comparable to the trained gesture classification performance of the FFMLP. However, when conducting one-way ANOVA to compare the performance of the SSAE and FFMLP, the FFMLP appeared to have a significantly higher performance than the SSAE for the trained ANGC results, as indicated with a p-value of 0.001.

The confusion matrices in Figure 4.11 and Figure 4.12 demonstrate that for the SSAE, when a particular gesture was considered misclassified through the ANGC method, more than one digit was frequently misclassified, which is comparable to the FFMLP. For the SSAE, the confusion matrix for the highest performing untrained gesture combination averaged across all participants reveals that when a misclassification occurred, the index and little finger positions were more frequently misclassified simultaneously, compared to the misclassifications of the other two digits.

The top performing participant for gesture combination 14 was subject 7. The performance for subject 7 for trained gestures for GC 14 is shown in Figure 4.3. Subject 7 data had improved performance with the SSAE when compared to the mean performance for gesture combinations 14.

4.3.3. SSAE Performance with WDGC Method

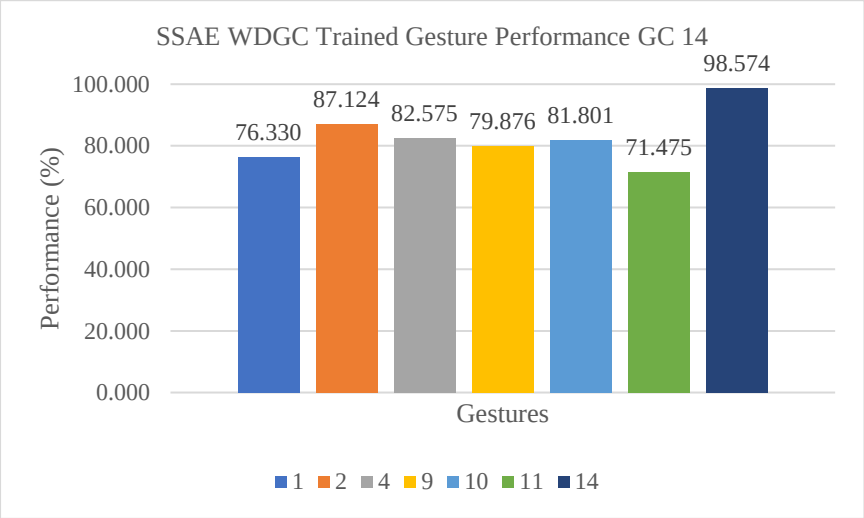

Figure 4.13, this graph depicts the SSAEs trained gesture classification accuracy for the WDGC method for GC 14 across all participants

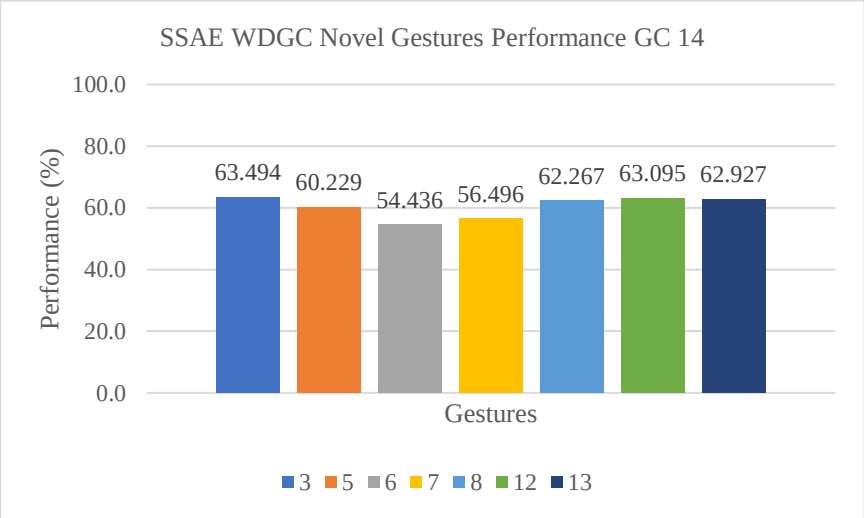

Figure 4.14, this graph depicts the SSAEs novel gesture classification accuracy for the WDGC method for GC 14 across all participants

As with the WDGC performance for the FFMLP, the results for the SSAE showed considerable improvements over the ANGC method. The average performance for GC 14 for both trained and novel gestures can be seen in Figure 4.13 and Figure 4.14. For the trained WDGC method, there was a significant difference between the performance of the FFMLP and SSAE with a p-value of 0.000. The performance between these two algorithms was not significantly different for the novel WDGC results, with a p-value of 0.875.

The data collected from subject 7 had the highest classification accuracy for SSAE WDGC method for GC 14. The performance for each gesture in this combination for subject 7 is shown in Figure 8.10 and Figure 8.11 for trained and novel gestures respectively. The top performing trained gesture in this combination, gesture 2, hand closed, had a performance of 75.66%. For the WDGC results for the trained gestures, the average performance for subject 7, gesture combination 14 was 85.18%. Aside from rest, the highest performing WDGC trained gesture was gesture 4, with a classification accuracy of 90.09%. Gesture 4 corresponds with index and middle extended.

4.4. CNN Results

4.4.1. CNN Performance for Gestures Digit Classification

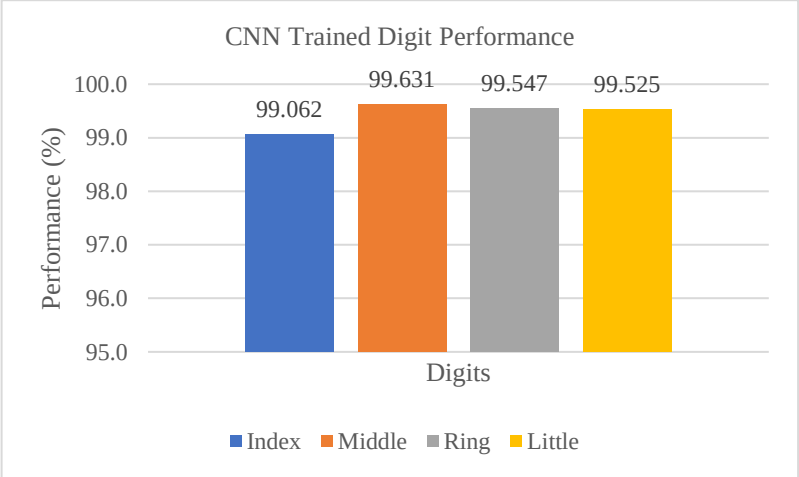

Figure 4.15, this graph shows the digit classification accuracy for trained gestures across all gesture combinations and participants

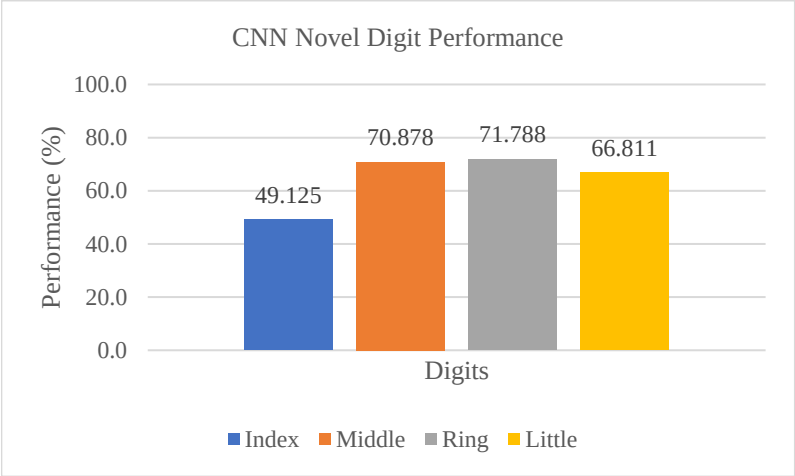

Figure 4.16, this graph shows the digit classification accuracy for novel gestures for across all gesture combinations and participants

The digit classification performance for trained gestures was very high, with the CNN generating digit classification accuracies above 99% for all digits for trained gestures as shown in Figure 4.15. For novel gestures, the relative ranking of digit classification performance is similar across all algorithms, with the ring and middle fingers achieving the highest performance, followed by the

little and index fingers. However, the CNN obtained higher classification accuracies for all digits when compared to the FFMLP and SSAE except for the index finger. The index finger performance for novel gestures was on average less than 50% as shown in Figure 4.16.

4.4.2. CNN Performance for Gestures with ANGC Method

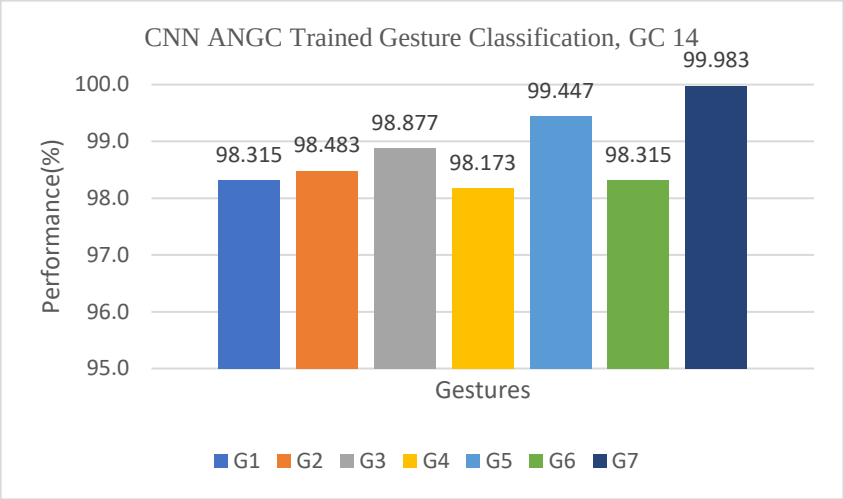

Figure 4.17, this graph shows the classification performance for all trained gestures in GC 14 across all participants for the ANGC method

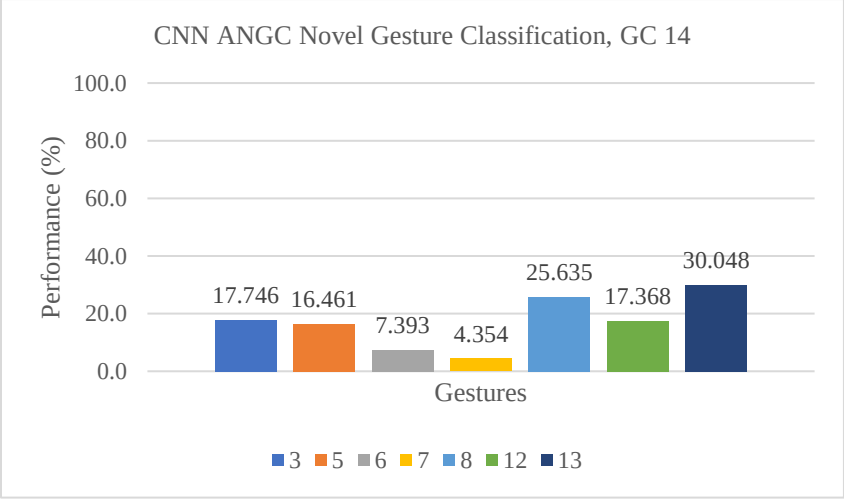

Figure 4.18, this graph shows the classification performance for all novel gestures in GC 14 across all participants for the ANGC method

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	98.32	0.00	0.08	0.03	0.00	0.06	**1111**
	2	0.08	99.77	0.11	0.15	0.00	0.14	**0000**
	4	0.14	0.01	98.88	0.03	0.00	0.07	**1100**
	9	0.03	0.00	0.00	98.17	0.02	0.11	**0011**
	10	0.03	0.01	0.00	0.13	99.45	0.00	**1001**
	11	0.08	0.02	0.00	0.09	0.00	98.32	**0110**
	13	0.00	0.07	0.00	0.36	0.39	0.00	**0111**
	6	0.00	0.01	0.04	0.03	0.00	0.18	**0100**
	13	0.38	0.00	0.00	0.23	0.00	0.81	**0111**
	3	0.11	0.13	0.02	0.00	0.02	0.00	**1000**
	7	0.18	0.00	0.00	0.62	0.13	0.00	**1011**
	12	0.49	0.00	0.68	0.00	0.00	0.00	**1101**
	5	0.14	0.00	0.19	0.00	0.00	0.22	**1110**
	N	0.03	0.00	0.00	0.16	0.00	0.09	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 4.19, the confusion matrix for GC 14 for the active nodes in the SSAE for trained gestures across all participants

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	17.75	0.06	0.97	0.16	2.51	7.00	0.30	**1000**
	5	0.01	16.46	5.23	0.66	0.00	1.08	2.63	**1110**
	6	0.20	0.63	7.46	0.09	0.06	0.12	0.24	**0100**
	7	2.11	0.52	0.18	4.35	0.72	1.28	1.06	**1011**
	8	3.56	0.06	0.00	4.14	25.64	0.48	0.14	**0001**
	12	4.87	8.61	1.37	0.97	0.02	17.37	0.67	**1101**
	13	0.00	5.39	0.01	3.37	0.00	0.34	30.05	**0111**
	2	23.92	0.19	0.11	0.23	6.39	0.22	0.24	**0000**
	9	0.00	1.00	0.01	47.28	14.81	5.04	17.93	**0011**
	11	0.01	42.80	21.21	10.93	0.00	2.10	22.67	**0110**
	10	14.70	2.10	1.12	18.64	49.64	21.86	1.07	**1001**
	4	26.97	9.43	61.86	4.27	0.03	38.11	1.71	**1100**
	1	5.89	12.63	0.46	1.18	0.00	4.59	18.65	**1111**
	N	0.00	0.12	0.01	3.72	0.18	0.43	2.65	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 4.20, the confusion matrix for GC 14 for the active nodes in the SSAE for novel gestures across all participants

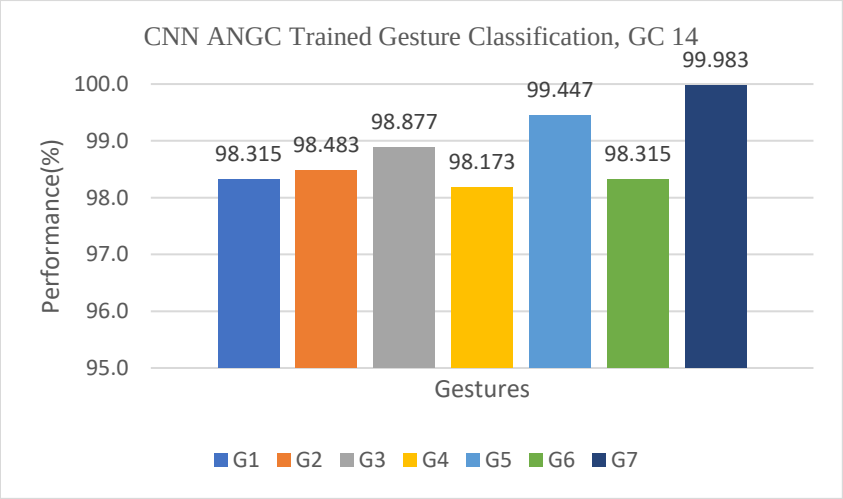

Figure 4.17 and Figure 4.18 show the average performance for the top gesture combination, GC 14 across all participants for trained and novel gestures respectively for the ANGC analysis method. While the novel gesture performance shows that some gestures performed better than random chance, the performance is still low and not suitable for prosthetic control as it relates to

classifying novel gestures but is suitable for classifying trained gestures. The CNN achieved a high classification accuracy for trained gestures, with almost no misclassifications. All ANGC gestures achieved classification accuracies of 98% or above across all participants for GC 14. The CNN outperformed both the FFMLP and SSAE for ANGC trained results, with p-values of 0.000 for both CNN-FFMLP, and CNN-SSAE.

Subject 9 data was the highest performing with the CNN, and Figure 8.14 in Appendix 8.5 for trained gestures in GC 14, subject 09, shows that with the exception of gestures 9 and 11, all trained gestures were classified with 100% accuracy.

The average performance for novel gestures across all participants was low, however Figure 8.13 shows that the CNN was able to predict some gestures with higher performance using the ANGC method for some participants. Novel gesture 8 was correctly classified 93.23% of the time, with only the index finger being misclassified 4.04% of the time. This gesture corresponds to index, ring and little finger extended. Figure 8.12 depicts the trained gesture performance for subject 9.

Promising classification accuracies were generated by other subjects and gesture combinations as well. The CNN for subject 9, GC 9 classified untrained gestures 1 and 2 with accuracies of 83.4% and 80.5% respectively. The CNN trained on data from GC 7, subject 4 was able to classify untrained gesture ‘1’ with 71% accuracy, and ‘9’ with 51% accuracy, which corresponds to open hand, and ring/little extended. The CNN for subject 3, GC 7 classified untrained gesture 10 with an accuracy of 80.5%, and a CNN for GC 4 classified untrained gesture 10 with 88.2% accuracy. Gesture 10 is index/little extended. These results indicate that for certain gestures and subjects, the network can generalize relationships that correspond to individual finger positions for all digits simultaneously.

When reviewing the misclassification rates for each novel gesture in GC 14 from Figure 4.20, we can see that for most misclassifications, only a single digit was incorrectly classified. The index finger appears to be responsible for the majority of misclassification events for the novel gestures. There were some instances where multiple digits were misclassified, for instance 10.93% of misclassifications were due to the incorrect prediction of both index, middle, and little finger positions for gesture 7. However, 47.28% of misclassifications for gesture 7 were due to index finger alone.

For GC 14, the data from subject 9 demonstrates improved results with respect to the mean for GC 14, and one novel gesture produced high performance in accordance with the ANGC rules at an accuracy of 93.23%.

4.4.3. CNN Performance with WDGC Method

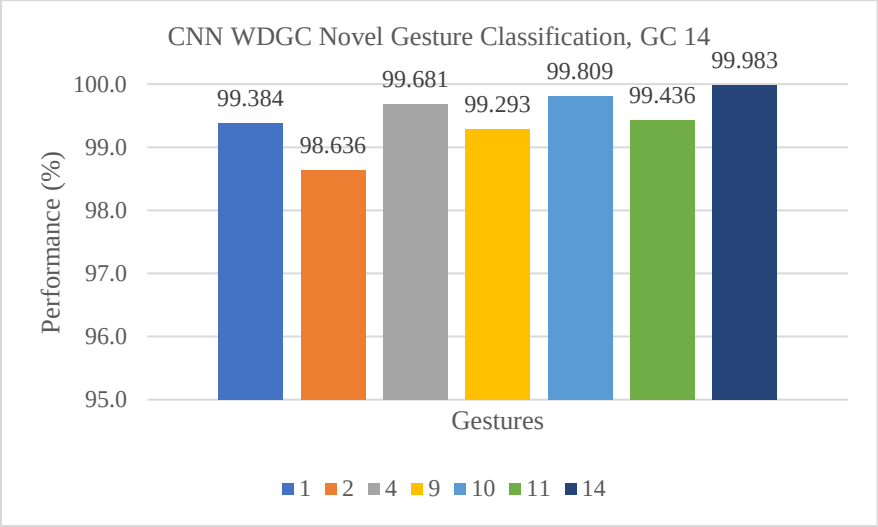

Figure 4.21 this graph shows the classification performance for all novel gestures in GC 14 across all participants for the WDGC method

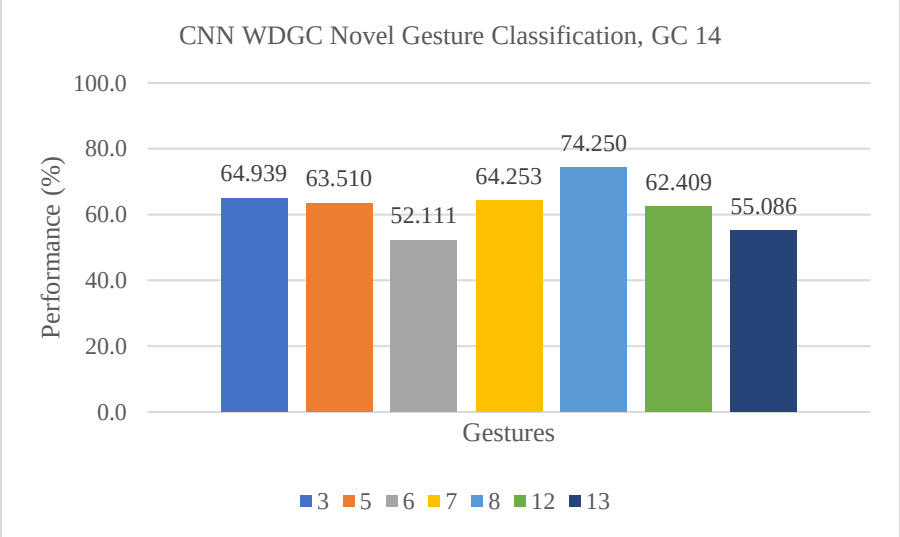

Figure 4.22, this graph shows the classification performance for all novel gestures in GC 14 across all participants for the WDGC method

The WDGC method provides a more accurate insight into the potential usefulness of a classifier for applications in prosthetic control. For the trained gestures processed via the WDGC method, the performance is similar to the performance of trained gestures for the ANGC method but demonstrates higher performance than the trained WDGC results for the FFMLP and SSAE. The

p-value for the CNN in comparison to both the FFMLP and SSAE novel and trained WDGC results was 0.000.

The performance for new gestures for the WDGC method for GC 14 is shown in Figure 4.22. The performance shows some improvement when compared to the performance for GC 14 with the FFMLP and SSAE. Like the SSAE and FFMLP WDGC results, the reason for the low performance for untrained gestures averages across all participants is likely due to the misclassification of the index finger, due to its relatively high weighting.

Figure 8.17 shows the novel classification performance for WDGC for the subject with the top performing data, subject 9. Here we can see a higher performance, with only 2 of the 7 novel gestures having produced a classification accuracy of less than 80%, and the highest performance of 97.76% produced by gesture 5. The matrices in Figure 8.14 and Figure 8.15 in Appendi 8.6, show the performance for subject 9 for trained and novel gestures respectively. For subject 9, the index finger was responsible for a lower number of misclassifications when compared to the average results for GC 14 previously discussed in section 4.4.2. For gesture 13, the index finger was misclassified 18.65%, the little finger was misclassified at a rate of 22.67%, and the middle finger at 17.93%. For gesture 13, 64.99% of misclassifications were the result of little finger, while the index finger was only misclassified 2.45%. For gesture 12, the middle and index fingers were responsible for 74.48% and 0.14% of the classification errors respectively.

What this tells us that the trained CNN model is more useful for prosthetic control for subject 9 when compared to the average across all participants, as the misclassification of the little or ring fingers would have a less detrimental impact on the systems usefulness for the users due to the lower weighting of these digits.

4.4.4. Gesture Combinations

Statistical analysis results indicate that there is a statistically significant difference between the performance of gesture combinations. For trained ANGC, trained WDGC and novel WDGC the p-values were 0.023, 0.016, and 0.003 respectively, indicating that the performance differences between different gesture combination is significant between at least two combinations for all analysis methods. Therefore, the null hypothesis can be rejected, and it can be stated that the different gesture combinations produce significantly different classification results.

4.4.5. Participants

The performance was also dependant on the participant. For trained ANGC, trained WDGC and novel WDGC the p-values were 0.021, 0.02, and 0.019 respectively. Possible explanations for the impact of the participant on the performance beyond random variability will be discussed in the next section. This indicated that the performance differences between subjects is significant between at least two subjects. Therefore, the null hypothesis can be rejected, and it can be stated that the different subjects do achieve significantly different classification results.

5. Discussion

5.1. Summary of Key Findings

From the results, we have determined the following:

1. The SSAE, FFMLP and CNN all produced low classifications accuracies for novel gestures based on the ANGC method, though there were some instances of high classification accuracy for select subjects and gestures for the CNN.
2. The SSAE, FFMLP and CNN all produced acceptable results for the classification of sEMG signals from trained gestures for the ANGC method.
3. The SSAE, FFMLP and CNN produced acceptable results for the classification of trained and for the WDGC analysis method, that may indicate suitability for prosthetic control for trained gesture classification. Across all participants the CNN demonstrates the strongest performance, and may be applicable for prosthetic control in classifying both trained and novel finger positions for some participants.
4. There is no statistically significant difference in performance between the SSAE and FFMLP for trained ANGC and novel WDGC results. The FFMLP produced statistically significant results when compared to the SSAE for the trained WDGC.
5. The CNN results are significantly different from the SSAE and FFMLP for trained ANGC and, both trained and novel WDGC methods.
6. The classification accuracy differences between subjects are statistically significant between at least two participants. The classification accuracy differences between gesture combinations are statistically significant between at least two combinations.

The discussion is broken into two parts. The first part is divided into sub-sections that focus on each neural network algorithm. The second part of the discussion identifies possible sources of error, considerations for methodological improvements based on the possible sources of error and discusses future research directions that will build off of the results of this study.

5.2. Feed Forward Multi-Layer Perceptron

The FFMLP has, in other research demonstrated promising applications in classifying isometric gestures and other movements from both raw and post processed sEMG signals as discussed in section 2.5.1. The question that was posed in this thesis was whether this algorithm could generalize relationships from raw, sEMG data that correspond to the discrete positions of individual fingers, and classify trained and novel gestures, as well as trained and novel digits positions.

The results indicate that raw data may not be effective for the FFMLP to be used in the classification of either trained or novel discrete finger positions, as it relates to the ANGC classification method. The instructions given to direct participants to perform the gestures, and the labelling method used for the sEMG data may be the greatest limitation in this case. It is possible that inconsistent performance of the intensity of flexion extension for digits across gestures, combined with the labelling method involving binary labelling of digit positions and the use of raw sEMG data lead to low performance. The use of raw data is more likely to be the problem, as studies such as [25], have demonstrated that a similar binary labelling method and output architecture can be used to classify discrete digit positions using wavelet transform of the signal for trained gestures. Due to the use of different input features between the network proposed in this thesis, and the system used in [25], it is difficult to determine if the differences in performance between these output architectures would be significant. The trained results from the classification

of individual fingers, and the results of trained gestures from GC 14 indicate that the FFMLP may be suitable for classification of digit positions from raw data for trained gestures.

We can see from Table 7.4 in Appendix 8.2, which depicts the raw FFMLP outputs for novel gesture 5, GC 14, that the mean decisions for all digits were close to the extend/flexion threshold of 0.5, indicating poor differentiation of digit positions across different gestures, and a poor ability to isolate digit-specific sEMG signal features.

5.3. Stacked-Sparse Auto-Encoder

In using the same number of layers as the FFMLP discussed in the previous section, the stacked autoencoder was used to evaluate how a different method of training a network with the same number of parameters might improve performance. Because training each layer as an autoencoder forces the network to create a compressed representation of the previous layers' outputs, it was hypothesised that the network might be able to isolate the signal features that are related to the positions of the individual fingers. However as discussed in section 4, there was either no significant difference between the FFMLP and SSAE, or the FFMLP produced significantly better results than the SSAE.

5.4. Convolution Neural Network

The CNN demonstrated the highest performance capabilities compared to the SSAE and FFMLP when classifying both trained and novel sEMG signals for the WDGC results, as well as trained gestures for the ANGC results. As discussed in section 4, there were certain instances where the CNN was able to classify select gestures for some participants with acceptable accuracy, but overall the performance was not significant. On average, the CNN misclassified at least 1 digit for most novel gestures. For many misclassifications, this was the index finger, which is, according

the weights derived from the commercial prostheses grasp patterns, is the most important digit in the daily life of trans-radial prosthesis users. Table 7.5 in Appendix 8.6 shows that the raw output values for the index finger active node are very close to the threshold of 0.5 used to produce the binary output to represent each discrete digit position. This indicates that the network may not have been able to assign any definitive signal feature to the position of the index finger, or that the index finger position may have been related to multiple contradictory features.

This is evident when reviewing the raw classification outputs of other digits within the same gesture. For the middle finger, the subset of raw outputs shown are classified close to the target, with the lowest value in the samples presented in Table 7.5 in Appendix 8.6 is 0.8953. The target for this digit for gesture 5 is one (digit extended). All other node outputs, including the rest nodes, show a similar pattern, where the raw classification output was close to the value of the target, and therefore, after being passed through the threshold filter, were correctly classified.

This indicated that the CNN may be relating the position of the index finger to signal patterns associated with other digits. For instance, this problem could arise if the CNN is determining through the training process that the index finger should be flexed when the ring or little fingers are flexed and extended in other conditions.

As for subject 9, the incorrectly classified digit was most often the little finger. This digit was misclassified nearly half the time, indicating that overall, the system was unable to extract meaningful relationships from the raw data that could be used to reliably classify. Instead the network achieved performance similar to random guess of finger position.

There is considerable difference in performance between the trained and novel gestures. It is possible that the CNN was overfitting to the trained gestures. Changing the size of the fully-

connected layers, introducing dropout layers into the fully-connected layers, and varying the batch size were also tested to improve generalization, with no significant improvements as described in section 3.4.5.5. Considering that the CNN was able to produce reasonable classification accuracies for most digits on average with the exception of the index finger, it is more likely that some other underlying problem such as the network relating certain features to the positions of multiple digits, or the labelling method coupled with inconsistent flexion/extension of digits between gestures, was affecting the performance. Due to the 'black box' nature of deep learning algorithms, it is not clear what features the CNN was extracting from the raw sEMG signals, but it is possible that the CNN could have extracted muscle synergies or patterns of muscle signals that manifest across multiple gesture.

5.4.1. Gesture Combinations

The gesture combinations had a statistically significant influence on performance across all networks for the WDGC method, as well as trained gestures using the ANGC method. Therefore, it may be possible that certain combinations of gestures may be more optimal for use in training to allow the network to classify new gestures with these same learned properties. It is also possible that there is a more balanced distribution of gestures between training and novel subsets that are alike in the intensity of flexion/extension for each digit, leading to similar signal properties between trained and novel gestures.

5.4.2. Between Participants

At least two participants produced classification results that were significantly different from one another. This may simply due to random variability or may be the related to how consistently participants performed flexion/extension tasks across all gestures. Those that performed the

flexion/extension tasks for each digit consistently across all isometric gesture may have produced better performance than those who were less consistent.

5.4.3. WDGC Method

For the WDGC method, we can see that the reason for the sub 60% performance in most cases is due to the misclassification of only the index finger. While only one finger is misclassified in these scenarios, the index finger is weighted very highly in accordance with the parameters described in section 3, and as a result, misclassification of the index finger results in significantly reduced performance for a given gesture, despite the fact that the other digits were classified correctly. The WDGC method can be viewed as a classification accuracy analysis method that serves as a more effective indicator of the usefulness of the system for real-world prosthetic control applications, when compared to the ANGC method.

5.4.4. ANGC Method

The ANGC method demonstrated unsuitable classification performance accuracies for novel gestures for all networks, though there were select instances for the CNN where the ANGC produced high classification accuracies that would be suitable for prosthetic control for novel gestures. The ANGC method is more comparable to the discrete gesture classification method that is conventionally used in other sEMG classification studies. However, the ANGC method of analysis is not as suitable when classifying individual digits, as it ignores correct outputs for all other digits, should a single digit be misclassified. Therefore, The ANGC method is not an effective measure for the true performance of the network as it relates to the systems ability to generalize and classify novel sEMG data to predict individual digit positions.

Regarding other studies conducted using a CNN to classify trained gestures, the ANGC method performed well on average for gesture combination 14. For example, overall the average performance for GC 14 was 98.63% across all gestures and participants. Studies such as [12], which used a CNN to classify seven trained gestures using eight differential sEMG channels, showed a within-session classification accuracy of 97.74%. Therefore, the performance of the CNN proposed in this thesis demonstrates a marginal improvement in comparison the performance of the CNN architecture used in [12].

As discussed in section 2.5.2.7, the study in [40] used high density sEMG channels to classify instantaneous sEMG signals, and the CNN used in [40] achieved a performance of 89.3% before majority voting was applied, for the classification of eight isometric gestures. The proposed CNN used in this thesis outperformed the classification of the algorithm used in Geng et al before the use of majority voting but underperformed slightly in comparison to the post majority voting performance of 99.5%. Overall it can be stated that the proposed CNN in this thesis offers improved functionality over the CNNs used in these previous studies, because of the comparable ability of the CNN used in this thesis to classify novel gestures, coupled with the ability to classify the discrete digit positions from novel gestures to some degree.

5.5. Methodological Considerations and Future Development

The main source of error in the methodology was likely the binary digit labelling coupled with inconsistent performance of digit flexion/extension tasks across gestures.. During the experiment, the participants were asked to either fully extend or fully flex digits, and it is possible that a lack of consistent force and/or positioning for each finger between isometric gestures may have been responsible for the classification errors. This is because there was no external apparatus such as a data glove, optical tracking system, or force gauge, used to ensure that the sEMG signals

consistently corresponded to the exact flexion/extension state of a digit. For example, in one gesture, a finger may have been fully extended, but for another gesture, this same finger may have only been extended with 70% force of the previous gesture. This digit for both gestures would have been assigned the same label of '1' to represent an extended digit, despite the activations of this digit not being consistent between these gestures [22]. This problem has been documented to affect the performance of sEMG signal classification, as discussed in section 2.4.

To circumvent this limitation in the future, transfer learning could be used. A paper by [31] described using transfer learning in a CNN to train a general EMG model, then fine tune this model for particular users. This method could be employed such a way where the initial model would be trained with data collected from a large set of able-bodied subjects, with their joint kinematics being recorded simultaneously with the sEMG recordings. The classifier would then be trained on this data from all able-bodied participants and can be fine tuned on a per user basis. The fine-tuning process would have the user perform a few gestures with discrete finger positions similar to those performed for this study, without capturing the hand kinematics, and the same labelling method as proposed and explored in this thesis could be used to provide supervised training targets for the users sEMG data.

During the data collection experiment, care was taken to ensure that the electrode locations between participants were similar, though it is possible that inconsistency in electrode positions between participants may account for differences in classification accuracy between participants. Using an electrode array design similar to the Myo armband used in studies like [12], would make consistent electrode placement more reliable between participants.

To explain the reason why the majority of misclassification were due to only one digit for the CNN, it is possible that the training process for the CNN is relating certain signal features to the positions of multiple digits, which therefore results in high classification for trained gestures, but lower classification for novel gestures. A possible solution for this issue for the CNN, would be to explore algorithms that introduce sparsity into the weights of the fully connected layers of the network. More specifically this type of algorithm would ensure that each feature extracted from the raw sEMG is related to the position of one or two digits, and any features related to multiple digits are penalized. This method could make the classification of digits entirely independent from one another, and in theory would improve the networks generalization capabilities to digit positions from novel gestures.

It will also be important to investigate the use of training sets that include more gestures, and therefore include fewer gestures in the novel sub-sets. This could assist in determining if there is an ideal number of gestures that should be included in training, and to evaluate the point at which increasing the number of training gestures has diminishing improvements in performance of novel gestures or has adverse effects on novel classification performance.

Frequency domain features have been demonstrated to be effective for the classification of discrete gestures using CNNs [44]. The application of both phase and power spectrum features extracted from the raw sEMG data to determine if these features are more effective in classifying novel gestures can also be investigated. The use of a separate neural networks for each DOF can also be explored, as this has been demonstrated to produce better results than single network for multiple DOFs [3].

The MATLAB toolbox for neural networks has limitations regarding what activation functions and backpropagation algorithms can be used with different types of network architectures. For example, the Tanh function that is available to use with feedforward FFMLPs, but not for autoencoders. Therefore, future development will be conducted using tools better suited for building networks that require a high degree of customizability such as Python and the related machine learning toolboxes like Tensorflow.

6. Conclusions

The objectives of this study were as follows:

1. Evaluate the performance FFMLP, SSAE and CNN with classifying isometric hand gestures from raw sEMG data for trained gestures and novel gestures.
2. Evaluate whether the gesture combinations had an impact on the classification performance.
3. Classify digit positions without training the network with kinematic data from the participants hand.

The classification accuracy for novel finger positions for the CNN was statistically significant for 3 of the 4 digits. The FFMLP and SSAE do not appear to be well suited for applications in novel and trained gesture classification when used with raw sEMG data, the proposed output architecture, and labelling methodology applied in this study.

The results indicate that the gesture combination plays a role in classification accuracy, which may indicate that certain gesture combinations are more effective at providing a wide enough range of information to allow a classifier to predict the positions of individual digits from both trained and novel gestures. The statistical results also indicate that there was a significant difference in performance between subjects, which may be due to degree of consistency when performing the isometric hand gestures, and therefore should be more carefully controlled in future experiments. The results for certain participants for the WDGC method indicate that the CNN proposed in this study may have applications in predicting finger positions from novel sEMG patterns for prosthetic control applications, though there are some significant classification problems that need to be resolved before this can be seen as an applicable system for prosthetic control. Based on the

findings of this study, CNNs warrant further investigation for applications in the classification of novel patterns of sEMG signals to predict digit positions.

7. References

[1] A B Ajiboye R F Weir, “Muscle synergies as a predictive framework for the EMG patterns of new hand postures,” vol. 6, no. 3, 2011.

[2] K. Englehart, B. Hudgins, P. A. Parker, and S. Member, “A Wavelet-Based Continuous Classification Scheme for Multifunction Myoelectric Control,” *IEEE*, vol. 48, no. 3, pp. 302–311, 2001.

[3] M. Ison and P. Artemiadis, “The role of muscle synergies in myoelectric control: Trends and challenges for simultaneous multifunction control,” *Journal of Neural Engineering*, vol. 11, no. 5. IOP Publishing, 2014.

[4] K. J. Zuo and J. L. Olson, “The evolution of functional hand replacement : From iron prostheses to hand transplantation,” vol. 22, no. 1, pp. 44–51, 2014.

[5] W. J. Gaine, C. Smart, and M. Bransby-Zachary, “Upper limb traumatic Amputees,” 1997.

[6] G. H. Kejlaa, “Consumer concerns and the functional value of prostheses to upper limb amputees,” pp. 157–163, 1993.

[7] H. Gottinger and C. Umali, “Organizational Entrepreneurship : A Historical Overview on Industry Alliances in Biotech and Pharmaceuticals Organizational Entrepreneurship : A Historical Overview on Industry Alliances in Biotech and Pharmaceuticals,” no. November, 2014.

[8] M. Asghari Oskoei and H. Hu, “Myoelectric control systems-A survey,” *Biomedical Signal Processing and Control*, vol. 2, no. 4. pp. 275–294, 2007.

[9] B. Hudgins, P. Parker, S. Member, N. Robert, and S. Member, “A New Strategy for Multifunction Myoelectric Control,” *IEEE Trans. Biomed. Eng.*, vol. 40, no. I, pp. 82–94, 1993.

[10] K. G. Keenan, “Influence of amplitude cancellation on the simulated surface electromyogram,” *J. Appl. Physiol.*, vol. 98, no. 1, pp. 120–131, 2004.

[11] M. G. B. Fonseca, A. G. S. Conceição, E. F. S. Filho, A. M. Tm, and A. Sensor, “ARTIFICIAL NEURAL NETWORKS APPLIED TO THE CLASSIFICATION OF HAND GESTURES USING ELETROMYOGRAPHIC SIGNALS,” 2017.

[12] M. Z. ur Rehman *et al.*, “Multiday EMG-Based classification of hand motions with deep learning techniques,” *Sensors (Switzerland)*, vol. 18, no. 8, pp. 1–16, 2018.

[13] A. Fougner, O. Stavdahl, P. J. Kyberd, Y. G. Losier, and P. A. Parker, “Control of upper limb prostheses: Terminology and proportional myoelectric controla review,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 20, no. 5, pp. 663–677, 2012.

[14] M. Hioki and H. Kawasaki, “Estimation of Finger Joint Angles from sEMG Using a Neural,” vol. 2012, 2012.

[15] V. C. K. Cheung, A. Avella, M. C. Tresch, and E. Bizzi, “Central and Sensory Contributions to the Activation and Organization of Muscle Synergies during Natural Motor Behaviors,” vol. 25, no. 27, pp. 6419–6434, 2005.

[16] J. T. Belter, J. L. Segil, A. M. Dollar, and R. F. Weir, “Mechanical design and performance specifications of anthropomorphic prosthetic hands: A review,” vol. 50, no. 5, pp. 599–618, 2013.

[17] Ottobock, “Bebionic User Guide.”

[18] M. Atzori and H. Müller, “Control Capabilities of Myoelectric Robotic Prostheses by Hand Amputees : A Scientific Research and Market Overview,” vol. 9, no. November, pp. 1–7, 2015.

[19] A. D. Bellingegni *et al.*, “NLR , MLP , SVM , and LDA : a comparative analysis on EMG data from people with trans-radial amputation,” pp. 1–16, 2017.

[20] Coapt LLC, *COMPLETE CONTROL Handbook*. 2018.

[21] R. Mahendran, “EMG Signal based Control of an Intelligent Wheelchair + IEEE,” *2014 Int. Conf. Commun. Signal Process.*, pp. 1267–1272.

[22] E. Scheme and K. Englehart, “Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use,” vol. 48, no. 6, pp. 643–660, 2011.

[23] C. Antfolk, C. Cipriani, M. Controzzi, M. C. Carrozza, and F. Sebelius, “Using EMG for Real-time Prediction of Joint Angles to Control a Prosthetic Hand Equipped with a Sensory Feedback System,” vol. 30, no. 6, pp. 399–406, 2010.

[24] R. J. Smith, F. Tenore, D. Huberdeau, R. Etienne-Cummings, and N. V. Thakor, “Continuous decoding of finger position from surface EMG signals for the control of powered prostheses,” *2008 30th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, pp. 197–200, 2008.

[25] A. T. C. Au and R. F. Kirsch, “EMG-based prediction of shoulder and elbow kinematics in able-bodied and spinal cord injured individuals,” *IEEE Trans. Rehabil. Eng.*, vol. 8, no. 4, pp. 471–480, 2000.

[26] M. Jiang, R. Wang, J. Wang, and D. Jin, “A method of recognizing finger motion using wavelet transform of surface EMG signal,” *Conf Proc IEEE Eng Med Biol Soc*, vol. 3. pp. 2672–2674, 2005.

[27] L. J. Hargrove, K. Englehart, and B. Hudgins, “A comparison of surface and intramuscular myoelectric signal classification,” *IEEE Trans. Biomed. Eng.*, vol. 54, no. 5, pp. 847–853, 2007.

[28] N. Nazmi, M. A. A. Rahman, S. I. Yamamoto, S. A. Ahmad, H. Zamzuri, and S. A. Mazlan, “A review of classification techniques of EMG signals during isotonic and isometric contractions,” *Sensors (Switzerland)*, vol. 16, no. 8, pp. 1–28, 2016.

[29] K. K. Jung, J. W. Kim, H. K. Lee, S. B. Chung, and K. H. Eom, “EMG pattern classification using spectral estimation and neural network,” *Proc. SICE Annu. Conf.*, pp. 1108–1111, 2007.

[30] E. H. Shroffe and P. Manimegalai, “Hand Gesture Recognition Based on Emg Signals Using Ann,” *IEEE Int. Conf. Consum. Electron. - Berlin, ICCE-Berlin*, vol. 2016-Octob, no. 3, pp. 174–178, 2016.

[31] U. Côté-Allard, C. L. Fall, A. Campeau-Lecoursy, C. Gosseliny, F. Laviolettez, and B. Gosselin, “Transfer learning for sEMG hand gestures recognition using convolutional neural networks,” *2017 IEEE Int. Conf. Syst. Man, Cybern. SMC 2017*, vol. 2017-Janua, pp. 1663–1668, 2017.

[32] J. C. Ives and J. K. Wigglesworth, “Sampling rate effects on surface EMG timing and amplitude measures,” vol. 18, pp. 543–552, 2003.

[33] A. Fougner, E. Scheme, S. Member, S. Member, K. Englehart, and S. Member, “Resolving the Limb Position Effect in Myoelectric,” pp. 1–8, 2011.

[34] C. Castellini *et al.*, “Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography,” *Frontiers in Neurorobotics*, vol. 8, no. AUG. pp. 1–17, 2014.

[35] Andrew Ng, “Multilayer Neural Network,” *Stanford*, 2017. .

[36] F. V. G. Tenore *et al.*, “Decoding of Individuated Finger Movements Using Surface Electromyography,” vol. 56, no. 5, pp. 1427–1434, 2009.

[37] A. Olsson, “sEMG Classification with Convolutional Neural Networks,” 2018.

[38] Andrew Ng, “Convolution Neural Networks,” *Stanford*, 2017. [Online]. Available: http://ufldl.stanford.edu/tutorial/supervised/ConvolutionalNeuralNetwork.

[39] L. Pan, D. Zhang, N. Jiang, X. Sheng, and X. Zhu, “Improving robustness against electrode shift of high density EMG for myoelectric control through common spatial patterns,” *J. Neuroeng. Rehabil.*, pp. 1–16, 2015.

[40] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, “Gesture recognition by instantaneous surface EMG images,” *Sci. Rep.*, vol. 6, no. November, pp. 6–13, 2016.

[41] Andrew Ng, “Autoencoder,” *Stanford*, 2017. [Online]. Available: http://ufldl.stanford.edu/tutorial/unsupervised/Autoencoders/.

[42] X. Glorot, A. Bordes, and Y. Bengio, “Deep Sparse Rectifier Neural Networks,” pp. 1–9, 2009.

[43] S. Santurkar, D. Tsipras, and A. Ilyas, “How Does Batch Normalization Help Optimization ?,” no. NeurIPS, 2018.

[44] U. Côtéallard *et al.*, “A Convolutional Neural Network for robotic arm guidance using sEMG based frequency-features,” *IEEE Int. Conf. Intell. Robot. Syst.*, vol. 2016-Novem, pp. 2464–2470, 2016.

[45] C. M. Light, P. H. Chappell, and P. J. Kyberd, “Establishing a Standardized Clinical Assessment Tool of Pathologic and Prosthetic Hand Function : Normative Data , Reliability , and Validity,” vol. 83, no. June, pp. 776–783, 2002.

[46] M. Zia ur Rehman *et al.*, “Stacked Sparse Autoencoders for EMG-Based Classification of Hand Motions: A Comparative Multi Day Analyses between Surface and Intramuscular EMG,” *Appl. Sci.*, vol. 8, no. 7, p. 1126, 2018.

[47] D. F. Ivan Vujaklija, Ernest N. Kamavuako, Vahid Shalchyan, Ning Jiang, “Online Mapping of EMG Signals into Kinematics by Autoencoding.”

[48] Mathworks, “trainAutoencoder Matlab Function.” [Online]. Available:

https://www.mathworks.com/help/deeplearning/ref/trainautoencoder.html#buyr01b-1.

[49] D. Datta *et al.*, “Myoelectric prostheses for below-elbow amputees : The Trent experience,” vol. 9147, 2009.

[50] J. Richards, *The Comprehensive Textbook of Clinical Biomechanics*, 2nd ed. 2018: Elsevier Ltd.

[51] K. Anam and A. Al-jumaily, “Real-time Classification of Finger Movements using Two-channel Surface Electromyography,” *Proc. Int. Congr. Neurotechnology, Electron. Informatics*, pp. 218–223, 2013.

[52] X. Wang, Y. Wang, Z. Wang, C. Wang, and Y. Li, “Hand gesture recognition using sparse autoencoder-based deep neural network based on electromyography measurements,” *Nano-, Bio-, Info-Tech Sensors, 3D Syst. II*, no. March, p. 42, 2018.

[53] D. Svoljšak, “Možjanca - novo bronastodobno najdišče v Alpah 1,” pp. 203–204, 2000.

[54] M. Riedmiller and H. Braun, “A direct adaptive method for faster backpropagation learning: The RPROP algorithm,” *IEEE Int. Conf. Neural Networks - Conf. Proc.*, vol. 1993-Janua, pp. 586–591, 1993.

[55] A. Ng, J. Ngiam, and C. Yu Foo, “UFLDL Standford Tutorial: Autoencoders,” 2013. [Online]. Available: http://ufldl.stanford.edu/wiki/index.php/Autoencoders_and_Sparsity.

8. Appendices

Table 8.1, Gesture Combinations used to train CNNs. C14 is in bold because it was the best performing gesture combination over all (See Section IIIA).

	Gesture Numbers													
Set	1	2	3	4	5	6	7	8	9	10	11	12	13	14
C1	U	U	U	U	T	T	T	T	U	T	T	U	U	T
C2	U	U	U	T	U	T	T	U	T	T	T	U	U	T
C3	U	U	U	T	T	U	U	T	T	T	T	U	U	T
C4	U	U	U	T	T	T	T	T	T	U	U	U	U	T
C5	U	U	T	U	U	T	T	U	U	T	T	U	T	T
C6	U	U	T	U	T	U	U	T	U	T	T	U	T	T
C7	U	U	T	U	T	T	T	T	U	U	U	U	T	T
C8	U	U	T	T	U	U	U	U	T	T	T	U	T	T
C9	U	U	T	T	U	T	T	U	T	U	U	U	T	T
C10	U	U	T	T	T	U	U	T	T	U	U	U	T	T
C11	T	T	U	U	U	T	T	U	U	T	T	U	U	T
C12	T	T	U	U	T	U	U	T	U	T	T	U	U	T
C13	T	T	U	U	T	T	T	T	U	U	U	U	U	T
C14	**T**	**T**	**U**	**T**	**U**	**U**	**U**	**U**	**T**	**T**	**T**	**U**	**U**	**T**
C15	T	T	U	T	U	T	T	U	T	U	U	U	U	T
C16	T	T	U	T	T	U	U	T	T	U	U	U	U	T
C17	T	T	T	U	U	U	U	U	U	T	T	U	T	T
C18	T	T	T	U	U	T	T	U	U	U	U	U	T	T
C19	T	T	T	U	T	U	U	T	U	U	U	U	T	T
C20	T	T	T	T	U	U	U	U	T	U	U	U	T	T

Table 8.2, this table shows data for GC 14 for pilot data used to determine window size. On average across both trained and untrained gestures, 25 millisecond window achieved the best performance.

	Trained Gestures for GC 14, Pilot Data							
Window Size	**1**	**2**	**4**	**9**	**10**	**11**	**14**	**Mean**
25 ms	47.29	85.49	87.24	75.56	82.74	34.53	99.33	**73.17**
50 ms	42.40	81.55	84.56	50.50	75.71	30.97	98.50	**66.31**
100 ms	42.56	74.25	80.85	55.27	71.99	25.08	97.66	**63.95**
200 ms	19.36	65.80	69.32	18.94	57.08	3.77	97.15	**47.35**
	Novel Gesture for GC 14, Pilot Data							
Window Size	**3**	**5**	**6**	**7**	**8**	**12**	**13**	**Mean**
25 ms	8.48	6.45	2.75	0.65	2.15	8.18	14.86	**6.22**
50 ms	2.89	1.30	0.93	2.30	2.45	2.92	21.91	**4.96**
100 ms	11.96	4.19	1.34	0.99	1.94	3.90	17.77	**6.01**
200 ms	2.30	0.39	0.11	1.04	3.48	4.04	40.40	**7.40**

Table 8.3, This figure depicts the importance. If a digit is used for grasping or interacting in a certain gesture, then it is assigned a value of 1, otherwise a digit is given a value of 0.

For instance, in 'index point', only the index finger is used to interact with objects, so it is given a value of 1, all other digits are given a value of '0'. For the active index however, the index, but the other digits are used to hold the object being used (like a squirt bottle or electric drill), therefore, all digits received a value of 1.

	Digits				
	Index	**Middle**	**Ring**	**Little**	**Grip Pattern**
Importance (1/0)	1	1	1	1	Power Grip
	1	0	0	0	Precision Open
	1	1	1	1	Hook Grip
	1	0	0	0	Precision Closed
	1	1	0	0	Tripod Grip
	1	1	1	1	Open Palm
	1	0	0	0	Index Point
	1	1	1	1	Mouse Grip
	1	1	1	1	Finger Adduction
	1	0	0	0	Pinch Grip
	1	1	0	0	Column
	1	0	0	0	Key Grip
	0	0	0	0	Relaxed
	1	1	1	1	Active Index
	13	8	6	6	**Total Sum**
	39.39	**24.24**	**18.18**	**18.18**	**Relative Importance (%)**

Figure 8.1, this figure depicts the 13 gestures available for use with the Bebionic V3 hand prosthesis developed by Ottobock. The "rest" gesture was omitted as it is not actively used for grasping or manipulation tasks

8.1. FFMLP ANGC

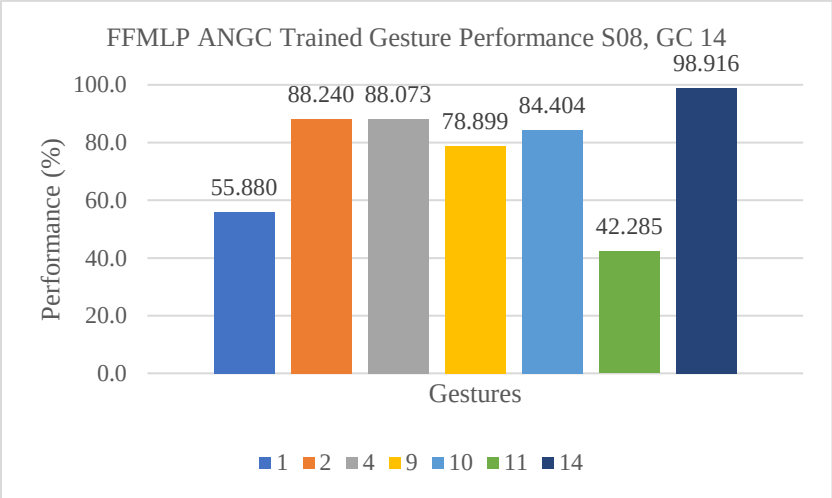

Figure 8.2, this figure depicts the MLPs classification performance for trained gesture combination 14 for subject 8

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	55.88	1.33	1.00	3.67	0.25	26.69	**1111**
	2	2.00	89.82	0.50	0.58	0.08	1.17	**0000**
	4	0.00	1.50	88.16	0.17	3.67	3.25	**1100**
	9	4.17	0.83	0.08	78.90	7.01	8.76	**0011**
	10	0.00	0.08	3.84	2.67	84.65	1.42	**1001**
	11	23.35	1.50	1.58	4.67	0.42	42.37	**0110**
	13	0.08	0.67	0.00	0.67	0.58	0.17	**0111**
	6	0.33	0.83	1.42	0.08	0.00	0.92	**0100**
	13	9.09	1.08	0.08	6.92	0.67	7.01	**0111**
	3	0.00	0.50	0.67	0.00	1.17	0.17	**1000**
	7	0.25	0.00	0.00	0.83	0.58	0.08	**1011**
	12	0.00	0.00	0.92	0.08	0.75	0.17	**1101**
	5	4.09	0.42	1.75	0.08	0.08	7.01	**1110**
	N	0.75	1.42	0.00	0.67	0.08	0.83	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 8.3, the confusion matrix for Subject 08, GC 14 for the active nodes in the MLP for trained gesture classification.

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	4.74	0.20	0.89	0.35	1.81	1.64	0.05	**1000**
	5	0.00	6.25	5.53	2.78	0.16	4.92	6.30	**1110**
	6	0.01	1.39	2.88	0.98	0.13	1.48	0.90	**0100**
	7	0.01	0.13	0.05	0.75	0.48	0.13	0.19	**1011**
	8	0.08	0.13	0.08	0.55	1.43	0.13	0.11	**0001**
	12	3.39	0.20	1.15	0.36	0.94	4.27	0.15	**1101**
	13	0.01	6.58	0.41	7.44	0.71	0.49	6.59	**0111**
	2	0.61	2.36	0.71	1.73	1.31	0.23	1.34	**0000**
	9	0.06	7.03	0.31	37.81	11.17	0.88	7.59	**0011**
	11	0.01	39.55	9.71	25.58	0.70	10.96	43.20	**0110**
	10	67.75	1.55	5.62	7.09	74.76	16.19	0.71	**1001**
	4	23.31	4.28	70.03	3.04	5.47	57.49	1.85	**1100**
	1	0.00	29.14	2.41	9.28	0.74	0.95	29.94	**1111**
	N	0.00	1.19	0.23	2.21	0.20	0.28	1.04	**0010**
	N	0.00	0.03	0.00	0.05	0.00	0.01	0.03	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 8.4, the confusion matrix for Subject 08, GC 14 for the active nodes in the MLP output for novel gesture classification.

8.2. FFMLP WDGC

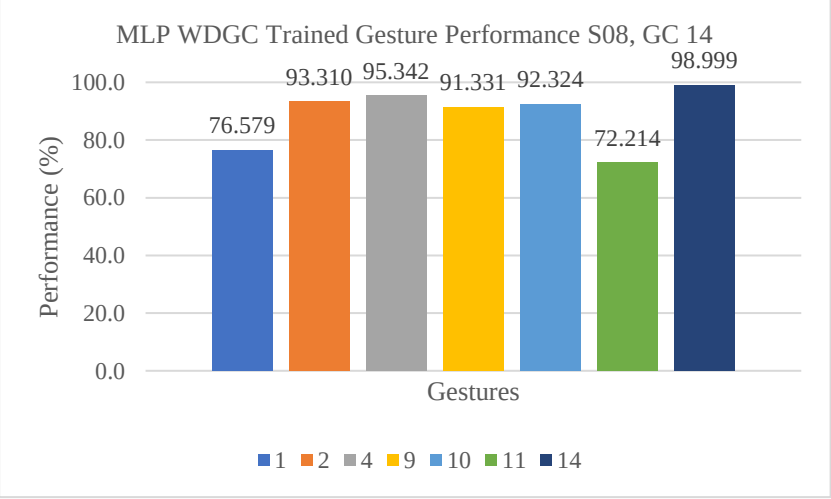

Figure 8.5, this figure depicts the MLPs classification performance for novel gesture combination 14 for subject 8 using the WDGC method

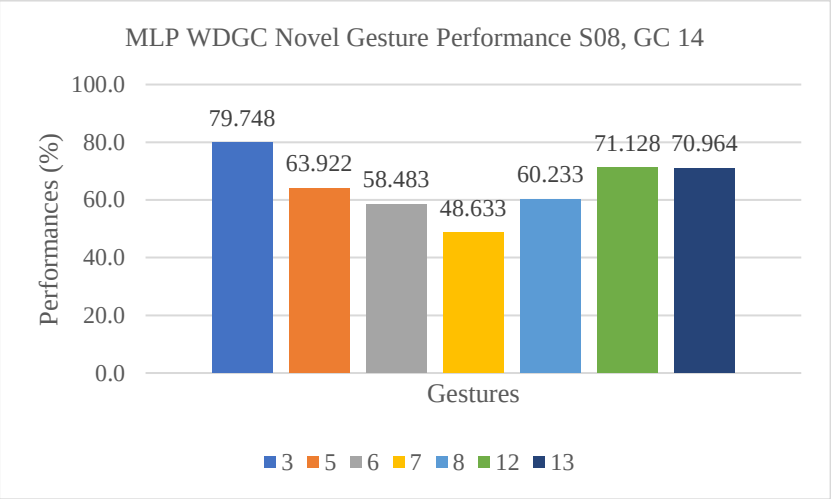

Figure 8.6, this figure depicts the MLPs classification performance for novel gesture combination 14 for subject 8 using the WDGC method

Table 8.4, This table depicts the raw output of the MLP novel gesture 5 from combination 14, before the threshold was applied, for each of the eight output nodes. Gesture 5 corresponds to index, middle, ring extended. In an ideal output, the index middle and ring fingers should produce values closer to one, and the little finger, as well as all rest nodes, should produce values close to zero.

	Subject 01/GC 14/Novel Gesture 5							
Target Output	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Output Nodes	**Index**	**Middle**	**Ring**	**Little**	**I Res**	**M Rest**	**R Rest**	**L Rest**
Raw CNN Outputs Samples	0.6367	0.6320	0.3506	0.4258	0.0000	0.0000	0.0000	0.0000
	0.3787	0.2813	0.5284	0.5610	0.0000	0.0000	0.0000	0.0000
	0.1845	0.3945	0.4018	0.2415	0.0000	0.0000	0.0000	0.0000
	0.8847	0.9561	0.8747	0.7313	0.0000	0.0000	0.0000	0.0000
	0.5848	0.6159	0.2974	0.3260	0.0000	0.0000	0.0000	0.0000
	0.1871	0.5312	0.8492	0.5750	0.0000	0.0000	0.0000	0.0000
	0.6418	0.8764	0.8553	0.5991	0.0000	0.0000	0.0000	0.0000
	0.2556	0.5601	0.7977	0.4973	0.0000	0.0000	0.0000	0.0000
	0.2975	0.5140	0.5951	0.3976	0.0000	0.0000	0.0000	0.0000
	0.6007	0.5450	0.6661	0.6801	0.0000	0.0000	0.0000	0.0000
	0.0702	0.4141	0.9871	0.4393	0.0000	0.0000	0.0000	0.0000
	0.8282	0.8478	0.1273	0.1286	0.0000	0.0000	0.0000	0.0000
	0.7896	0.8742	0.7503	0.6290	0.0000	0.0000	0.0000	0.0000
	0.5117	0.6269	0.4781	0.4010	0.0000	0.0000	0.0000	0.0000
	0.5463	0.2997	0.4971	0.6531	0.0000	0.0000	0.0000	0.0000
	0.5891	0.2228	0.4117	0.7091	0.0000	0.0000	0.0000	0.0000
	0.5166	0.6765	0.4818	0.3060	0.0000	0.0000	0.0000	0.0000
	0.1959	0.4260	0.2072	0.1279	0.0000	0.0000	0.0000	0.0000
	0.7971	0.6593	0.6360	0.7040	0.0000	0.0000	0.0000	0.0000
	0.3465	0.4637	0.6130	0.5041	0.0000	0.0000	0.0000	0.0000
	0.5421	0.1931	0.4545	0.8143	0.0000	0.0000	0.0000	0.0000
	0.1048	0.3854	0.4161	0.2369	0.0000	0.0000	0.0000	0.0000
Average	0.4768	0.5453	0.5580	0.4858	0.0000	0.0000	0.0000	0.0000

8.3. SSAE ANGC

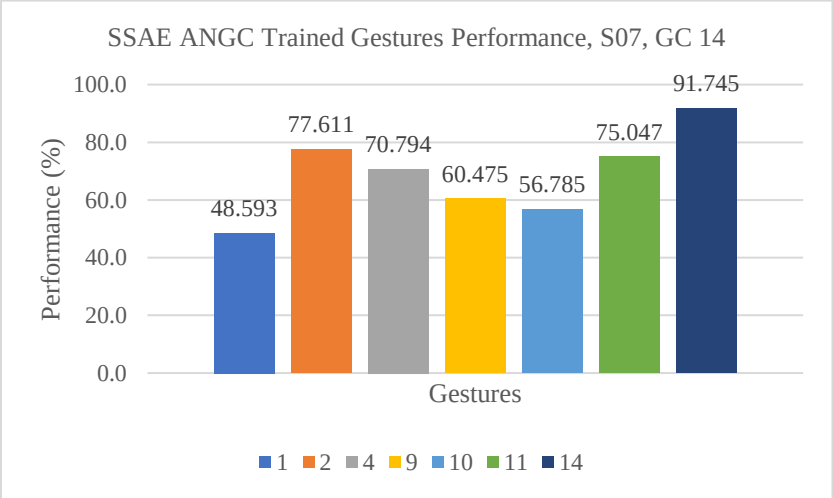

Figure 8.7, this figure depicts the SSAEs classification accuracy for the discrete position of each finger for novel gestures

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	48.59	0.50	12.63	2.13	6.63	10.32	**1111**
	2	1.25	88.24	2.31	1.69	3.81	0.13	**0000**
	4	13.26	2.19	70.79	0.25	1.56	2.31	**1100**
	9	4.57	0.44	0.06	60.48	14.20	3.50	**0011**
	10	5.75	2.38	2.13	18.20	56.79	0.75	**1001**
	11	8.44	0.50	0.50	2.44	0.31	75.05	**0110**
	13	0.44	2.19	0.06	3.19	3.75	0.00	**0111**
	6	0.38	0.56	0.56	0.06	0.00	0.31	**0100**
	13	3.69	0.06	0.06	2.81	0.25	3.50	**0111**
	3	0.38	1.63	0.94	0.06	0.81	0.06	**1000**
	7	3.50	0.19	0.38	7.94	7.88	1.06	**1011**
	12	5.69	0.25	5.50	0.19	3.75	0.25	**1101**
	5	3.88	0.13	3.94	0.00	0.13	2.25	**1110**
	N	0.13	0.75	0.13	0.56	0.13	0.50	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.06	0.00	0.00	0.00	0.00	0.00	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 8.8, the confusion matrix for GC 14 for the active nodes in the SSAE for trained gesture for subject 7

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	3.50	0.08	0.50	0.14	0.70	1.79	0.70	**1000**
	5	1.31	3.29	4.47	0.41	0.04	2.78	4.12	**1110**
	6	1.36	0.15	1.68	0.05	0.00	0.40	0.35	**0100**
	7	0.19	1.58	0.21	5.60	6.08	3.59	5.49	**1011**
	8	0.10	0.11	0.10	2.39	6.57	0.60	0.64	**0001**
	12	5.52	0.75	3.19	0.34	0.43	10.22	4.19	**1101**
	13	0.01	6.63	0.14	5.87	0.58	1.08	5.13	**0111**
	2	10.79	0.24	2.93	1.36	5.20	2.53	1.98	**0000**
	9	0.08	3.95	0.10	54.38	28.38	2.61	9.88	**0011**
	11	0.36	60.75	1.59	11.04	0.43	2.21	14.98	**0110**
	10	1.65	1.73	0.99	12.15	49.64	12.10	9.66	**1001**
	4	69.11	2.09	75.38	0.41	0.38	30.93	9.81	**1100**
	1	6.00	18.49	8.71	4.48	1.38	29.08	32.73	**1111**
	N	0.01	0.18	0.03	1.38	0.21	0.09	0.35	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 8.9, the confusion matrix for GC 14 for the active nodes in the SSAE for novel gesture for subject 7

8.4. SSAE WDGC

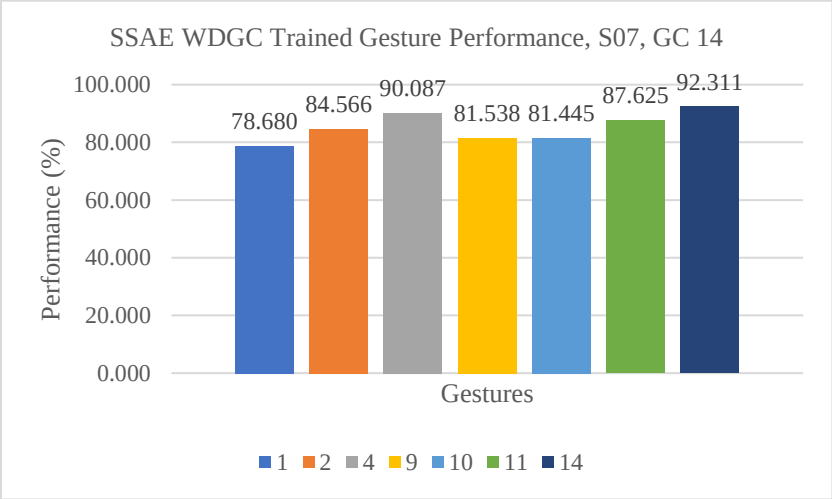

Figure 8.10, this figure depicts the SSAEs classification performance for trained gestures, gesture combination 14 for subject 7 using the WDGC method

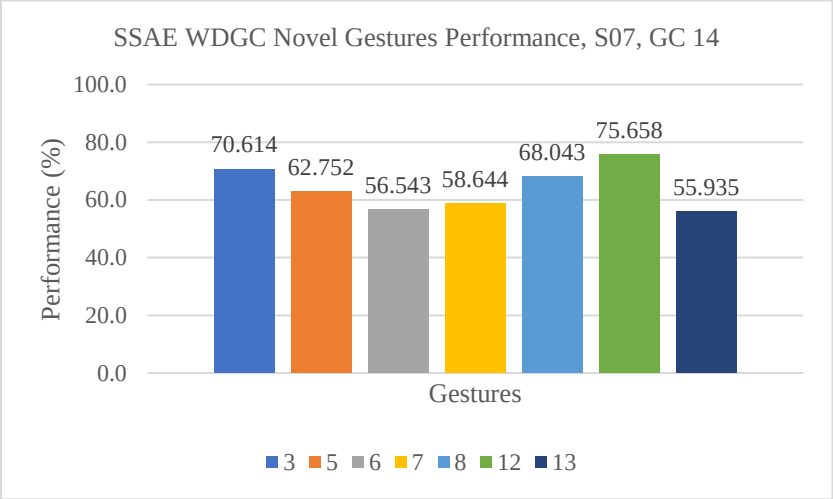

Figure 8.11, this figure depicts the SSAEs classification performance for novel gestures, gesture combination 14 for subject 7 using the WDGC method

8.5. CNN ANGC

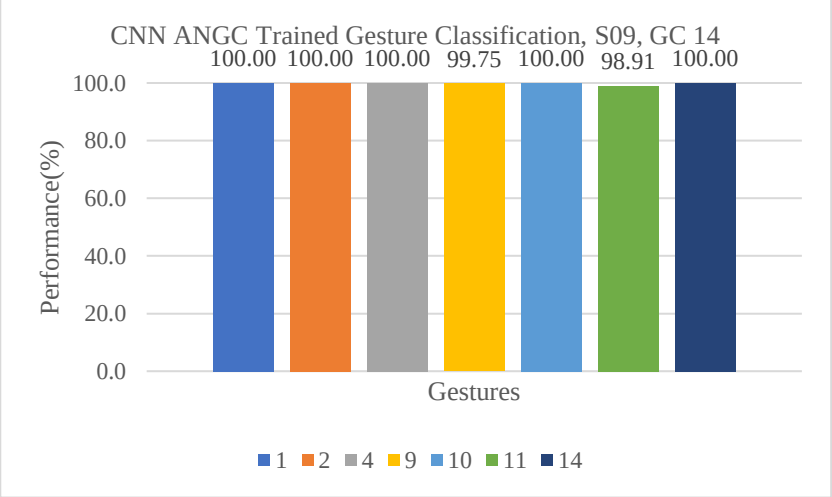

Figure 8.12, this figure depicts the CNNs classification performance for trained gestures, gesture combination 14 for subject 9 using the ANGC method

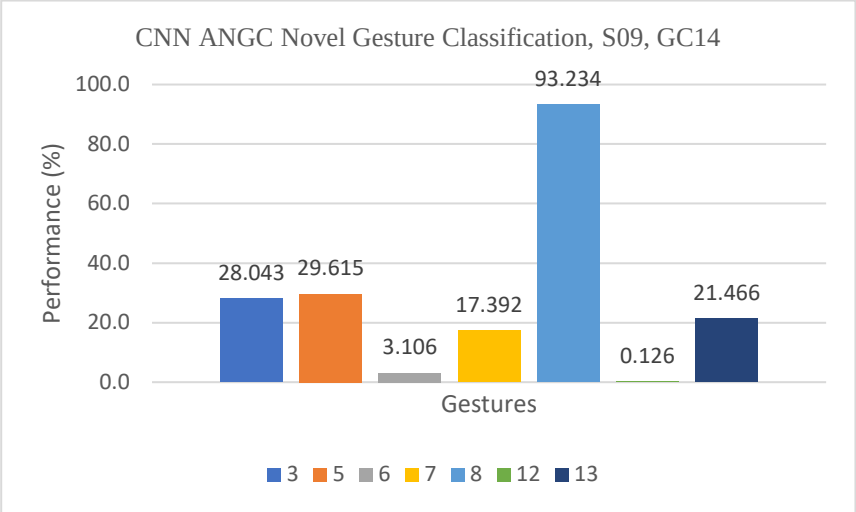

Figure 8.13, this figure depicts the CNNs classification performance for novel gestures, gesture combination 14 for subject 9 using the ANGC method

		Target Outputs						
		1	**2**	**4**	**9**	**10**	**11**	
Predicted Outputs	**1**	100.00	0.00	0.00	0.00	0.00	0.00	**1111**
	2	0.00	100.00	0.00	0.00	0.00	0.00	**0000**
	4	0.00	0.00	100.00	0.00	0.00	0.00	**1100**
	9	0.00	0.00	0.00	99.75	0.00	0.00	**0011**
	10	0.00	0.00	0.00	0.00	100.00	0.00	**1001**
	11	0.00	0.00	0.00	0.00	0.00	98.91	**0110**
	13	0.00	0.00	0.00	0.17	0.00	0.00	**0111**
	6	0.00	0.00	0.00	0.00	0.00	0.00	**0100**
	13	0.00	0.00	0.00	0.00	0.00	0.08	**0111**
	3	0.00	0.00	0.00	0.00	0.00	0.00	**1000**
	7	0.00	0.00	0.00	0.00	0.00	0.00	**1011**
	12	0.00	0.00	0.00	0.00	0.00	0.00	**1101**
	5	0.00	0.00	0.00	0.00	0.00	1.01	**1110**
	N	0.00	0.00	0.00	0.08	0.00	0.00	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1111	**0000**	**1100**	**0011**	**1001**	**0110**	

Figure 8.14, the confusion matrix for GC 14 for the active nodes in the CNN for trained gestures for subject 9

		Target Outputs							
		3	**5**	**6**	**7**	**8**	**12**	**13**	
Predicted Outputs	**3**	28.04	0.00	0.00	0.00	0.00	23.16	0.00	**1000**
	5	0.00	29.62	10.60	0.00	0.00	0.00	0.78	**1110**
	6	0.00	0.00	3.11	0.00	0.00	0.03	0.00	**0100**
	7	0.00	0.13	0.00	17.39	0.00	0.00	0.00	**1011**
	8	0.00	0.19	0.00	0.39	93.23	0.03	0.00	**0001**
	12	4.23	0.34	0.21	0.75	0.00	0.13	0.00	**1101**
	13	0.00	0.97	0.06	0.00	0.00	0.00	21.47	**0111**
	2	0.11	0.06	0.00	0.00	0.00	0.08	0.00	**0000**
	9	0.00	0.08	0.00	1.22	2.02	0.00	9.07	**0011**
	11	0.00	17.81	2.97	0.00	0.00	0.00	64.99	**0110**
	10	0.55	0.72	0.00	79.89	4.74	74.48	0.00	**1001**
	4	67.06	0.41	80.60	0.00	0.00	2.10	0.00	**1100**
	1	0.00	49.55	2.45	0.35	0.00	0.00	1.67	**1111**
	N	0.00	0.14	0.00	0.00	0.00	0.00	2.02	**0010**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**0101**
	N	0.00	0.00	0.00	0.00	0.00	0.00	0.00	**1010**
		1000	**1110**	**0100**	**1011**	**0001**	**1101**	**0111**	

Figure 8.15, the confusion matrix for GC 14 for the active nodes in the SSAE for novel gestures for subject 9

8.6. CNN WDGC

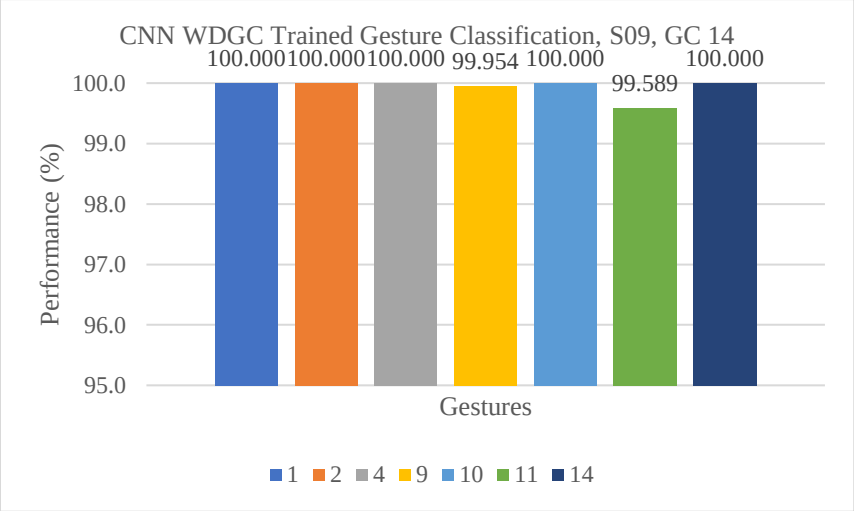

Figure 8.16, this figure depicts the CNNs classification performance for novel gestures, gesture combination 14 for subject 9 using the WDGC method

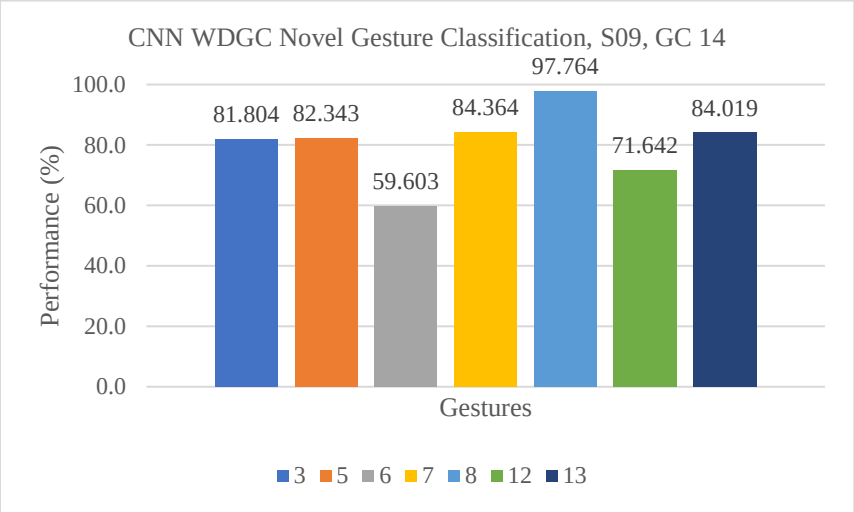

Figure 8.17, this figure depicts the CNNs classification performance for novel gestures, gesture combination 14 for subject 9 using the WDGC method

Table 8.5, This table depicts the raw output of the CNN novel gesture 5 from combination 14, before the threshold was applied, for each of the 8 nodes from. Gesture 5 corresponds to index, middle, ring extended. In an ideal output, the index middle and ring fingers should produce values closer to one, and the little finger, as well as all rest nodes, should produce values close to zero. This data is from subject 1 and appears to be more indicative of the average novel gesture 5 performance across all subjects, where the index finger is responsible for many misclassifications.

	Subject 01/GC 14/ Novel Gesture 5							
Target Outputs	**1**	**1**	**1**	**0**	**0**	**0**	**0**	**0**
Output Node	**Index**	**Middle**	**Ring**	**Little**	**I Rest**	**M Rest**	**R Rest**	**L Rest**
Raw CNN Outputs Samples	0.5178	0.8953	0.8955	0.0258	0.0152	-0.0153	-0.0150	-0.0151
	0.4934	0.9299	0.9305	0.0211	-0.0015	-0.0019	-0.0015	-0.0015
	0.5072	0.9112	0.9115	0.0315	-0.0157	-0.0158	-0.0155	-0.0159
	0.4909	0.9013	0.9012	0.0273	0.0012	0.0013	0.0014	0.0013
	0.5038	0.9196	0.9199	0.0254	-0.0122	-0.0120	-0.0124	-0.0127
	0.4995	0.9498	0.9496	0.0308	0.0068	0.0070	0.0066	0.0066
	0.4804	0.9759	0.9754	-0.0446	0.0085	0.0087	0.0086	0.0089
	0.4596	0.9707	0.9704	-0.0396	0.0049	0.0049	0.0052	0.0051
	0.4564	0.9778	0.9772	-0.0295	-0.0074	-0.0073	-0.0070	-0.0074
	0.4245	0.9985	0.9982	-0.0461	0.0027	0.0026	0.0029	0.0028
	0.4298	0.9662	0.9660	-0.0436	0.0014	0.0013	0.0012	0.0012
	0.4318	0.9726	0.9726	-0.0522	0.0120	0.0121	0.0119	0.0118
	0.4367	0.9500	0.9499	-0.0306	0.0076	0.0076	0.0076	0.0077
	0.4422	1.0046	1.0046	-0.0390	0.0094	0.0091	0.0093	0.0093
	0.4278	0.9906	0.9902	-0.0306	0.0081	0.0077	0.0079	0.0077
	0.4332	0.9860	0.9860	-0.0346	0.0106	0.0103	0.0105	0.0107
	0.4532	0.9868	0.9861	-0.0209	-0.0066	-0.0067	-0.0067	-0.0066
	0.4377	1.0227	1.0222	-0.0387	0.0100	0.0098	0.0099	0.0098
	0.4409	1.0034	1.0035	-0.0273	0.0178	0.0176	0.0176	0.0177
	0.4631	1.0239	1.0238	-0.0597	0.0176	0.0174	0.0172	0.0173
	0.4748	0.9707	0.9704	0.0228	0.0109	0.0109	0.0109	0.0109
	0.4768	1.0425	1.0423	-0.0385	0.0223	0.0226	0.0226	0.0226
Average	0.4628	0.9705	0.9703	-0.0178	0.0042	0.0042	0.0042	0.0042

8.7. Ethics Approval Confirmation

Ethics Clearance ORE # 32002

no-reply=kuali.co@mx3.kuali.co on behalf of Kuali Notifications <no-reply@kuali.co>

Fri 10/26/2018 9:28 AM

To:Erik Kolbjorn Lloyd <eklloyd@uwaterloo.ca>;

Dear Ning Jiang and other members of the research team:

Your application has been reviewed by Delegated Reviewers. We are pleased to inform you the Amendment application for 32002 Surface EMG Data Collection For the classification of individual finger movements has been given ethics clearance.

This research must be conducted in accordance with the most recent version of the application in the research ethics system and the most recent versions of all supporting materials.

Ethics clearance for this study is valid until Thursday, August 29th 2019.

The research team is responsible for obtaining any additional institutional approvals that might be required to complete this Expedited study.

University of Waterloo Research Ethics Committees operate in compliance with the institution's guidelines for research with human participants, the Tri-Council Policy Statement for the Ethical Conduct for Research Involving Humans (TCPS, 2nd edition), Internalization Conference on Harmonization: Good Clinical Practice (ICH-GCP), the Ontario Personal Health Information Protection Act (PHIPA), and the applicable laws and regulations of the province of Ontario. Both Committees are registered with the U.S. Department of Health and Human Services under the Federal Wide Assurance, FWA00021410, and IRB registration number IRB00002419 (Human Research Ethics Committee) and IRB00007409 (Clinical Research Ethics Committee).

Renewal: Multi-year research must be renewed at least once every 12 months unless a more frequent review has been specified on the notification of ethics clearance. This is a requirement as outlined in Article 6.14 of the Tri-Council Policy Statement for the Ethical Conduct for Research Involving Humans (TCPS2, 2014). The annual renewal report/application must receive ethics clearance before Wednesday, August 7th 2019. Failure to receive ethics clearance for a study renewal will result in suspension of ethics clearance and the researchers must cease conducting the study. Research Finance will be notified ethics clearance is no longer valid.

Amendment: Changes to this study are to be submitted by initiating the amendment procedure in the research ethics system and may only be implemented once the proposed changes have received ethics clearance.

Adverse event: Events that adversely affect a study participant must be reported as soon as possible, but no later than 24 hours following the event, by contacting the Director, Research Ethics. Submission of an adverse event form is to follow the next business day.

Deviation: Unanticipated deviations from the approved study protocol or approved documentation or procedures are to be reported within 7 days of the occurrence using a protocol deviation form.

Incidental finding: Anticipated or unanticipated incidental findings are to be reported as soon as possible by contacting the Director, Research Ethics. Submission of the incidental findings form is to follow within 3 days of learning of the finding. Participants may not be contacted regarding incidental findings until after clearance has been received from a Research Ethics Committee to contact participants to disclose these findings.

Study closure: Report the end of this study by submitting a study closure report through the research ethics system.

Initial application ethics clearance notification: Your clearance notification will be added to the record within 24 hours. Go to "View Admin Attachments" in the research ethics system (right-hand side) to print a copy of the initial application ethics clearance notification.

Best wishes for success with this study.

If you have any questions concerning this notification, please contact the Research Ethics Office or email researchethics@uwaterloo.ca.