Title
TypEMG: A Framework for Acquisition, Processing and Classification of EMG Signals

Author
Eren, Deniz Orkun

Publication Date
2023

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA
Los Angeles

TypEMG: A Framework for Acquisition, Processing and Classification of EMG Signals

A thesis submitted in partial satisfaction
of the requirements for the degree
Master of Science in Electrical and Computer Engineering

by

Deniz Orkun Eren

2023

ABSTRACT OF THE THESIS

TypEMG: A Framework for Acquisition, Processing and Classification of EMG Signals

by

Deniz Orkun Eren
Master of Science in Electrical and Computer Engineering
University of California, Los Angeles, 2023
Professor Jonathan Kao, Chair

Traditional input methods to interface with computer systems prove to be challenging for individuals with amputations or paralysis. Although several brain-machine interfaces were developed to address this problem, their invasive nature prevents widespread adoption. Alternatively, developing interfaces using non-invasive signals has been shown to be effective but they require large, non-intuitive gestures to function. In this work, we propose a framework to decode the subtle finger movements that occur naturally during typing via analyzing non-invasive EMG signals. Here, we establish synchronized communication with an amplifier to get signal recordings, perform signal preprocessing and utilize deep learning architectures for feature extraction and classification. Our approach achieves a within-session accuracy of up to 89.23% in detecting individual finger movements during a randomized typing task, with an average accuracy of 77.64% across all sessions. The time needed for classification is 4.16 ms per sample, making our framework suitable for real-time operation. Our framework demonstrates the possibility of identifying finger movements during typing in real-time using non-invasive EMG signals and provides a starting point for future work to allow individuals with amputations or disabilities to communicate effectively with computers.

The thesis of Deniz Orkun Eren is approved.

Ying Nian Wu

Achuta Kadambi

Jonathan Kao, Committee Chair

University of California, Los Angeles

2023

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would first like to thank my advisor, Professor Jonathan Kao. His expertise and guidance have been invaluable not only for this thesis but also for all aspects of my graduate studies.

I would also like to thank the rest of my committee members, Professor Achuta Kadambi and Professor Ying Nian Wu for their support during this thesis. I am grateful for their time and advice.

Special thanks should also go to my team members on this project: Rakshith Ravindra Gore, Johannes Lee, Abhishek Mishra, John Zhou and Nevin Liang. Working with them has been a great learning experience for me and I truly appreciate all the efforts they made towards the project. It is through their contributions that this thesis was made possible.

Lastly, I would like to thank my family, friends and colleagues. Their support and words of encouragement have been a constant source of strength for me during these two years.

CHAPTER 1

Introduction

Over the past few decades, the importance of accessing information has become vital for the personal and professional lives of many individuals. This development makes interfacing with computers essential, with the keyboard or a touchscreen being the conventional ways of providing input. However, for individuals with tetraplegia or loss of limbs, such means pose significant barriers to accessing and using these technologies due to limited mobility and dexterity. To overcome this barrier, recent progress on Brain-Computer Interfaces (BCIs) offers a viable alternative for such individuals, by allowing them to interface with computers via cursor movement and typing [3, 4, 5]. However, the need for surgery for these applications mitigates the potential of widespread usage while posing additional risks to the user [6].

As an alternative to BCIs to decode user intention, detection and classification of movement through the analysis of non-invasive signal recordings has been shown to be a viable approach. One such example of this kind of signal is electromyography (EMG), which is obtained by recording the electrical activity generated by the motor neurons during muscle contraction [7]. There have been several works throughout the years that demonstrate the feasibility of using EMG signals to decode finger movements of users [8, 9, 10]. Although these methods have been shown to be effective, their reliance on manual feature extraction prevents extending them to multiple use cases, as such features are found to be inconsistent across applications [11]. To overcome this disadvantage, recent work in classifying EMG signals focuses on utilizing deep learning techniques to extract the needed features automatically [12, 13]. These results show that it is possible to get state-of-the-art performance for

EMG signal classification without the need for manually defining features.

In addition to their usage in healthy subjects, EMG signals have been shown to be applicable for detecting movement intention in individuals with amputations [14, 15]. The work of Tenore et al. shows that it is possible to decode finger extension/flexion in amputees with 90% accuracy by utilizing surface EMG (sEMG) signals [15]. More surprisingly, the work of Ting et al. shows that sEMG signals can be detected and classified with deep learning methods for individuals with tetraplegia, where an individual has no control of any limb [6]. These results show that using EMG signals is a viable approach for determining the intended movement for disabled individuals.

Furthermore, using EMG signals has been shown to be effective in developing non-conventional means of interacting with computers. Recent demonstrations by CTRL-labs prove that it is possible to build real-time computer interfaces by determining hand movements and forces using EMG signals obtained from a simple wristband sensor attached to the forearm of a subject [16]. In addition, the recent work done by Crouch et al. at Bell Labs shows the feasibility of building a real-time typing system by decoding forearm muscle activity from sEMG signals [17].

In light of these developments, the main goal of our project dubbed TypEMG is to build a real-time typing interface for individuals with disabilities such as amputations or tetraplegia by decoding surface EMG signals using deep learning methods. Although there have been other attempts at using sEMG signals to allow people with amputations to perform typing [18], to our knowledge there has not been any work in building a real-time typing system for individuals with disabilities by decoding the distinct subtle finger movements that occur naturally during typing instead of relying on grand gestures.

This thesis will first provide a background for EMG technology and the foundations of the deep learning architecture utilized in this project. Then, the data collection, signal processing and classification pipeline will be further evaluated. Next, the results obtained so far will be conveyed and the next steps of the project will be outlined.

1.1 Contributions

The main contributions of this thesis can be summarized as follows:

- We introduce our TypEMG framework for acquisition, processing and classification of EMG signals.
- We show the validity of using a modified EEGNet, which is a deep learning architecture for EEG classification, to classify EMG signals and detect the individual subtle finger movements that occur during typing.
- We demonstrate that our system works well in a within-session classification scenario, with minimal operation time and training data requirements.
- We propose simple methods such as increasing the channel amount of the electrodes and changing the electrode orientation to improve classification accuracy and determine the effect of these changes on overall performance.

CHAPTER 2

Background

2.1 Electromyography

Electromyography (EMG) deals with the measure of the electrical activity in response to a nerve's stimulation of the muscle. During muscle contraction, the brain sends signals through a specific set of neurons in the motor cortex, which is located in the frontal lobe of the brain. These signals travel down through the spinal cord and out to the muscle via a long, thin nerve called an anterior horn cell, which is the name for a specific type of motor neuron that is responsible for ensuring connectivity between the spinal cord and muscle fibers [19]. The combination of the anterior horn cell, its axon and terminal branches, their neuromuscular junctions, and all the individual muscle fibers they innervate is referred to as a Motor Unit (MU) [1].

When an anterior horn cell is activated, all of the muscle fibers inside a motor unit are depolarized synchronously. This depolarization, accompanied by a movement of ions, generates an electric field near each muscle fiber, which is referred to as the motor unit activation potential (MUAP). This waveform is then recorded by the EMG electrodes [7].

There are two main types of EMG electrodes: surface electrodes and needle electrodes. Surface electrodes are made up of small adhesive patches that are placed on the surface of the skin overlying the muscle of interest. On the other hand, needle electrodes, also called intramuscular electrodes, are made up of thin, needle-like electrodes that are directly inserted to the muscle that is going to be examined. Needle electrodes are highly selective, capable

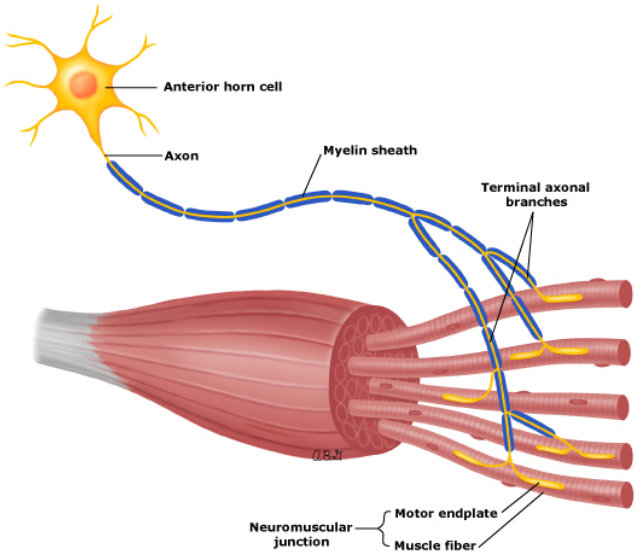


Figure 2.1: **An illustration of the motor unit [1].** The motor unit is comprised of the anterior horn cell, its axon and terminal branches, their neuromuscular junctions, and all the individual muscle fibers they innervate. The electrical activity generated during the activation of the anterior horn cell is what is detected in an EMG signal.

of getting single MUAPs with high signal-to-noise ratio. However, their invasive nature and difficulty of usage restricts them mostly to specialized clinical settings. On the contrary, surface electrodes are much easier to use, as they do not need to be inserted into the muscle and are non-invasive, which makes them safer and more convenient. However, the sEMG recordings obtained through such electrodes have low signal to noise ratio and offer low selectivity, as they contain the cross talk signal originating from surrounding muscles [20]. Nonetheless, their non-invasive nature makes them a popular choice for developing EMG applications.

For sEMG electrodes, the obtained signal can be modeled by the following equation [21, 22]:

$$\mathbf{y_i(n)} = \sum_{\mathbf{i=1}}^{\mathbf{K}} \sum_{\mathbf{l=0}}^{\mathbf{L-1}} \mathbf{c_{ij}(l)t_j(n-l)} + \mathbf{v_i(n)} \tag{2.1}$$

Where $\mathbf{y_i(n)}$ is the obtained EMG signal at time-step n on channel i, $\mathbf{c_{ij}(l)}$ is the MUAP

of motor unit j detected by channel i at time l, $\mathbf{t_j(n)}$ is a unit pulse train representing the MUAP times of motor unit j, where there is a pulse at time n if motor unit j is activated and $\mathbf{v_i(n)}$ is a zero mean additive white Gaussian noise that represents the system noise, which is independent of the motor units. As can be seen from the equation, the convolution operation is over a time period between 0 to L-1, due to the assumption that an MUAP lasts L time steps, and therefore only MUAPs happening in the last L time steps contribute to the signal at time 0 [21, 22].

sEMG electrodes can further be examined in two categories: bipolar and unipolar. Bipolar sEMG electrodes are made up of two metal surfaces placed close to each other on the skin over the muscle of interest. The goal is to detect the same MUAPs twice but spatially shifted along the muscle. The amplification of the difference of these two signals results in a reduction of noise while retaining the signal of interest [23]. On the other hand, unipolar sEMG electrodes consist of one metal plate that is placed on the skin over the examined muscle and a reference electrode placed somewhere else on the body. The unipolar electrode then measures the electrical activity of the muscle relative to the reference electrode and it is capable of representing the entire information about motor unit activity near the measurement point. One major drawback of bipolar electrodes when compared to their unipolar counterparts is that they require alignment of the electrodes with the muscle fiber direction. Achieving this alignment is difficult for most settings as the orientation of the muscle may change as a function of joint position and muscle force. In contrast, unipolar electrodes do not have an alignment constraint. However, their low spatial selectivity makes them susceptible to noise from stray-potentials, movement artifacts, and possible cross talk [23].

2.2 EEGNet

EEGNet is a deep learning architecture developed by Vernon J. Lawhern et al. in 2018 that aims to analyze and classify electroencephalogram (EEG) signals, which are measured by

Table 2.1: **Architecture of EEGNet [2].** C is the number of channels, T is the number of time points, F_1 is the number of temporal filters, D is the depth multiplier (number of spatial filters), F_2 is the number of pointwise filters and N is the number of classes.

Block	Layer	# filters	Size	# params	Output	Activation	Options
1	Input				(C, T)		
	Reshape				(1, C, T)		
	Conv2D	F_1	(1, 64)	$64 * F_1$	(F_1, C, T)	Linear	Mode = same
	BatchNorm			$2 * F_1$	(F_1, C, T)		
	DepthwiseConv2D	D $* F_1$	(C, 1)	$C * D * F_1$	(D $* F_1$, 1, T)	Linear	Mode = valid, depth = D, max norm = 1
	BatchNorm			$2 * D * F_1$	(D $* F_1$, 1, T)		
	Activation				(D $* F_1$, 1, T)	ELU	
	AveragePool2D		(1, 4)		(D $* F_1$, 1, T // 4)		
	Dropout*				(D $* F_1$, 1, T // 4)		$p = 0.25$ or $p = 0.5$
2	SeparableConv2D	F_2	(1, 16)	$16 * D * F_1 + F_2 * (D * F_1)$	(F_2, 1, T // 4)	Linear	Mode = same
	BatchNorm			$2 * F_2$	(F_2, 1, T // 4)		
	Activation				(F_2, 1, T // 4)	ELU	
	AveragePool2D		(1, 8)		(F_2, 1, T // 32)		
	Dropout*				(F_2, 1, T // 32)		$p = 0.25$ or $p = 0.5$
	Flatten				(F_2 * (T // 32))		
Classifier	Dense	N * (F_2 * T // 32)			N	Softmax	Max norm = 0.25

surface electrodes placed on the scalp of the subject and reflect the electrical activity of the brain [2]. The main motivation behind EEGNet is to build a single architecture that is able to achieve high performance in a multitude of EEG classification tasks, while also having the model be as compact as possible to allow real-time usage. The details of the model can be found in the paper "EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces" [2]. In TypEMG, we utilized a modified version of EEGNet to perform our initial decoding experiments.

The full architecture of EEGNet can be summarized in Table 2.1. An important point to note in this architecture is that EEGNet makes use of both spatial and temporal features of the EEG signal. To be more specific, the first convolutional layer examines the data in the direction of the time axis. This allows the generation of feature maps containing the EEG

signal at different band-pass frequencies. Then, the authors use a depthwise convolution to learn spatial filters for each temporal filter, thus enabling the efficient extraction of frequency-specific spatial filters. In block 2 of the architecture, the authors make use of a separable convolution layer, which aims to reduce the number of parameters of the model while also explicitly decoupling the relationship within and across feature maps.

EEGNet was tested on multiple EEG classification datasets. The first dataset, which is referred to as the P300 dataset, is related to the neural response to different visual stimuli, the second, nicknamed the ERN dataset is about the perturbations in EEG following an erroneous or unusual event in the subject's environment or task. The third dataset and the one most relevant to our project is about classification of finger digits from EEG signals. This dataset is referred to as MRCP in the paper. Finally, the fourth dataset, coded SMR, is related to classifying four imaginary movements namely, left and right hands, feet and tongue from EEG data.

For all tasks and for both within-subject and cross-subject classification scenarios, it was observed that EEGNet either achieves state-of-the-art performance or offers a performance very close to models that are designed specifically for a single task [2]. This is especially impressive considering that EEGNet has significantly less parameters than the specialized models, as shown in Table 2.2.

Although the results demonstrated in the paper are surely noteworthy, a natural question one may ask is why a network designed for EEG classification was utilized in TypEMG, which deals with the analysis and classification of EMG signals. There are a couple of key points that validate the usage of this network in our implementation. First of all, EEGNet was tested on a dataset that aims to detect finger movement from EEG signals [24] and was able to achieve close to state-of-the-art results [2]. As our task is also related to detecting and classifying finger digit movements based on physiological signals, it stands to reason to assume that EEGNet will perform well in our task as well. Moreover, in our project we needed an architecture that is capable of achieving real-time performance. Work done by

Table 2.2: **Number of trainable parameters of EEGNet and other architectures for EEG classification [2].** EEGNet-4,2 refers to an EEGNet configuration with 4 temporal filters and 2 spatial filters and EEGNet-8,2 refers to an EEGNet configuration with 8 temporal filters and 2 spatial filters. The models with the least amount of parameters are highlighted in bold.

	Trial length (s)	DeepConvNet	ShallowConvNet	EEGNet-4,2	EEGNet-8,2
P300	1	174127	104002	**1066**	2258
ERN	1.25	169927	91602	**1082**	2290
MRCP	1.5	175727	104722	**1098**	2322
SMR*	2	152219	40644	**796**	1716

Wang et al. demonstrates that EEGNet can be used to build real-time interfaces [25], which shows that the model is applicable in our case.

In addition to the relevancy of the capabilities of the EEGNet architecture for our application, there are also significant similarities between EEG and EMG signals. Much like EEG signals, EMG signals are spatiotemporal signals. This is because in addition to the time-dependent features, they contains spatial attributes due to the characteristics of the motor unit such as depth, number of muscle fibers, motor endplate zone and fiber distribution [26]. As EEGNet has convolutional layers that aim to extract both temporal and spatial characteristics, it is reasonable to assume that such an architecture will also fare well with EMG data. Furthermore, it has been shown that in voluntary muscle movement, such as the movement we are trying to detect, there is significant correlation between EEG and EMG signals [27]. Finally, previous work has shown that some deep learning architectures designed for classifying EEG signals can be utilized in the classification of EMG signals with high performance [28].

Based on the relevance of the model and similarities between EEG and EMG signals, it was deemed appropriate to use EEGNet for the initial EMG decoding experiments of our project. However, to accommodate for our experiment and changes in the structure of

the acquired data, some modifications were made to the original EEGNet architecture, the details of which can be found in the Methods section of this thesis.

CHAPTER 3

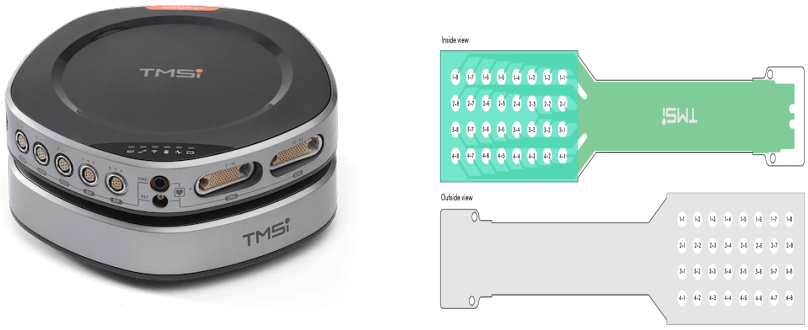
Methods

This chapter will provide more detail into the steps completed to acquire and analyze EMG signals in the TypEMG project. Namely, we will discuss the experiment and data collection setup, give information regarding the collected dataset and explain the signal processing and decoding methodology employed so far.

3.1 Data acquisition hardware

In order to record the EMG signals that are generated during typing, we utilized the Twente Medical Systems International (TMSi) SAGA 32+/64+ High Density Amplifier. In the unipolar mode, this amplifier supports 32 or 64 channel recordings with a sampling rate of 4 kHz. Each unipolar channel has an RMS noise of less than 1 microVolt for signals between 0.1 to 100 Hz with a resolution of less than 20 nanoVolts. The amplifier also has a built-in 1.6 kHz anti-aliasing filter. The experiments for the project were conducted in the unipolar mode with common referencing, where the difference between each unipolar input channel and a reference electrode placed on the body was taken as the collected signal. Further details regarding the technical specifications of the amplifier can be found in its datasheet [29].

To record the MUAPs occurring over the muscles of interest, we made use of the 4-8-L HD-EMG electrodes provided by TMSi. These are unipolar surface EMG electrodes that contain 32 channels, with an inter-electrode distance of 8.75 millimeters (mm) and each individual channel having a contact area of 4 mm diameter with the skin. Surface EMG



(a) (b)

Figure 3.1: **Hardware components of the experiments.** a) TMSi SAGA 32+/64+ High Density Amplifier. b) TMSi 4-8-L HD-EMG Electrode layout. The EMG activity that occurs during typing is detected via the TMSi 4-8-L HD-EMG Electrodes, which have 32 channels equally spaced out. The detected signals are then collected by the TMSi SAGA 32+/64+ High Density Amplifier.

electrodes were chosen instead of needle electrodes due to their non-invasive nature and easier application during experiments. Furthermore, unipolar electrodes were preferred over bipolar options in order to avoid alignment concerns during data collection. More details about the electrodes can be obtained from their datasheet [30].

3.2 Establishing communication between TMSi SAGA amplifier and local workstation

In order to establish synchronized communication between the TMSi SAGA amplifier and the local workstation the data would be stored at, a custom software interface was created. This interface was based on Stanford Brain Interface Laboratory's Linux Comodular Realtime Interactive Computation Engine (LiCoRICE) platform [31], which was previously successfully used in our lab to record EEG recordings. For our project, we modified the

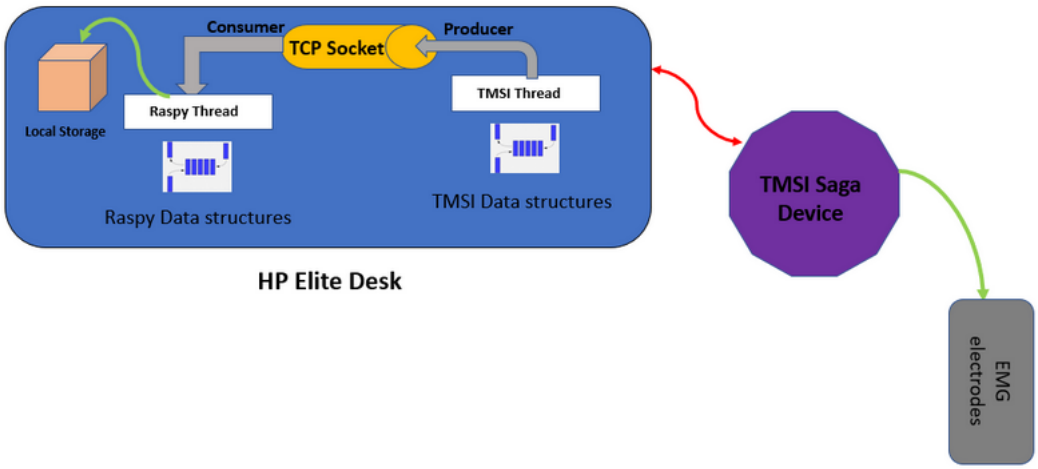


Figure 3.2: **An illustration of the communication setup between the TMSi SAGA amplifier and local workstation.** In our setup, two parallel threads are run to collect data. The TMSI thread initiates data collection from the amplifier and stores the collected data in an internal buffer. Buffer contents are periodically transferred trough a Transmission Control Protocol (TCP) connection to the Real-Time Asynchronous Python (RASPy) thread. This thread synchronizes input from the keyboard with the acquired data and moves the data to the local workstation, which was an HP Elite Desk for our experiments.

existing codebase to be suitable to read EMG data from our TMSi amplifier.

A simplified illustration of the created communication setup can be seen in Figure 3.2. The communication system works by running two parallel threads. The first one, dubbed the TMSi thread, is responsible for initiating data collection from the amplifier. As data is collected, it is stored in an internal buffer. The contents of the buffer are periodically transferred to a bipartite buffer in the other parallel thread through a Transmission Control Protocol (TCP) connection. This thread is called the Real-Time Asynchronous Python (RASPy) thread and it has two main purposes: storing the EMG data in the local storage by periodically emptying the contents of the bipartite buffer and running the experiment

setup used to collect data from a subject. As the user types on a computer, this thread runs a key logger that records each key press and release. When the system detects a key press or release, the index of the associated signal that is moved from the buffer to the local storage is noted. This allows us to determine between which EMG signal indices a specific key event occurred.

3.3 PyGame for key collection

To aid in the data collection process and guide the subjects through the experiment, a simple PyGame was created. In the PyGame, the subject is presented with a sequence of characters to press, where each character corresponds to a specific finger. During the initial calibration phase, only five characters are displayed and the key that the subject will press for that character is recorded. After the calibration phase, the subject is displayed random permutations of the five characters used during calibration, with a random sequence length between a minimum of 20 characters and a maximum of 30 characters. During the experiment, each key press is recorded, even if the pressed key does not match the character displayed on the screen. However, it is important to note that the subject is instructed to always use the same finger to press a specific key. This ensures that even if the recorded key press is different than what is prompted on screen, it can still be treated as a data sample for a specific finger digit. At the end of the experiment, we obtain a dataset with five distinct labels, each corresponding to a different key that is pressed with a specific finger digit. The amount of data for each class is kept approximately the same throughout the experiment.

3.4 EMG data collection

In order to create the dataset to perform decoding on, data was collected from three healthy male subjects, with ages 24, 27 and 31 as the PyGame explained in Section 3.3 was running.

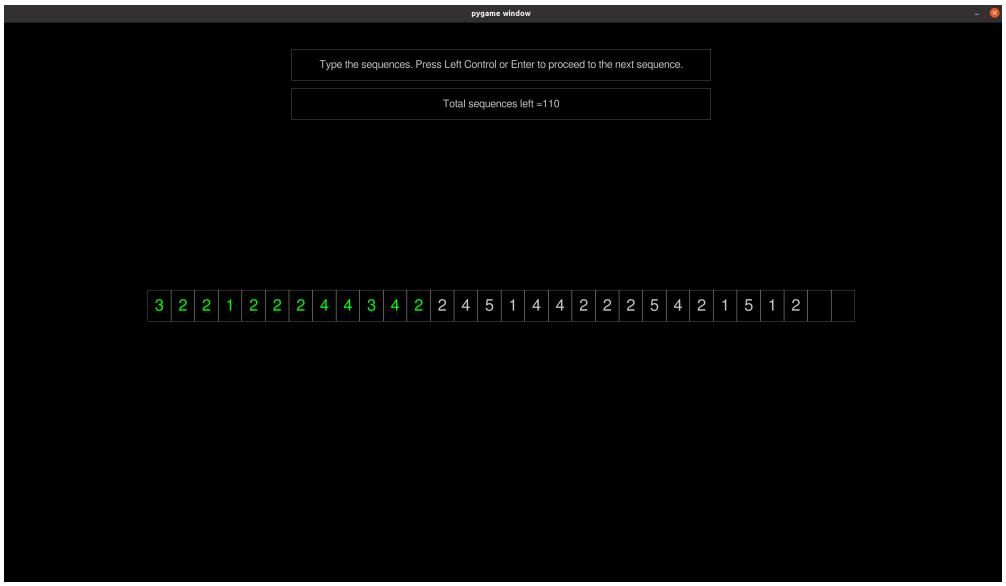


Figure 3.3: **PyGame screen presented to the subject during the experiment.** The display of the PyGame is quite simplistic. The subject is presented with a random sequence of characters that are defined during calibration. As the subject types, the correctly pressed characters on the screen change color from white to green. Once the final character on screen is typed, the next sequence is randomly generated and the screen is updated.

Here, each key press the subjects performed was recorded along with the associated EMG signals. Although the subjects were instructed to press the keys approximately once per second, it was observed that the subjects showed variations in typing speeds among themselves and across different data recording sessions.

To get the EMG signals, the 32 channel TMSi electrodes were placed on the right forearms of the subjects covering the proximal part of the anterior superficial muscles: flexor carpi radialis, palmaris longus, pronator teres and flexor carpi ulnaris in a vertical orientation parallel to the muscle fibers. The position of the electrodes was chosen to cover the muscle groups most associated with movements that occur during typing [32]. During data collection, the TMSi SAGA amplifier was set to be in common reference mode, where for each channel the unipolar inputs were referenced against another electrode. For our experiments,

Figure 3.4: **Positioning of the electrodes.** For data collection, the 32 channel electrodes were placed vertically along the proximal part of the anterior superficial muscles of a subject's right forearm. This ensured maximum coverage of the muscle groups most associated with typing. Before the experiment began, the electrodes were fixed in place with tape to minimize movement artifacts.

this reference electrode was placed on the right external oblique muscles. The location of the reference was chosen to minimize any movement artifacts that may affect signal quality during the experiment.

Each data collection session lasted approximately one hour, resulting in an average of 3527 recorded key presses per session. Details of the data collection from each subject can be summarized in Table 3.1. For all data collection sessions, the amount of data for each of the five classes was evenly distributed.

Table 3.1: **Amount of data collected from each subject per session.** Session refers to an hour of running the experiment and collecting data. Overall, a total of 21161 samples of key presses were recorded for feature extraction and classification.

Subject Number	Age	Number of Data Collection Sessions	Number of Key Samples Collected
1	24	3	Session 1: 3893, Session 2: 2407, Session 3: 4563
2	27	1	Session 1: 3999
3	31	2	Session 1: 2880, Session 2: 3419

3.5 Signal processing

In order to remove the noise and maximize the information that can be obtained, the acquired EMG signals were band-pass filtered between 20 Hz and 500 Hz. The reason 20 Hz was chosen as the lower cutoff frequency was to account for the noise originating from the inherent instability of the signal as done in various other literature [7, 6]. Although our experiments show that performing filtering with lower cutoff frequency at 10 Hz as was done in the work of Crouch et al. [17] can lead to better classification results in some cases, we wanted to keep the lower cutoff at 20 Hz as was done in the work of Ting et al. [6] in order to remove artifacts that may arise from movement, since our end goal is to develop a system that will be used by disabled or tetraplegic individuals.

After filtering, the acquired signal was average referenced by taking the average of all the channels at a time t and subtracting this result from the data of each channel at that time. This was done to eliminate the common noise generated by our data collection system in the acquired signal. After that, the signal was downsampled 4 times to be at a frequency of 1 kHz. The motivation behind downsampling the signal was to reduce the computational load on our model during training. The higher cutoff frequency of 500 Hz set during band-pass filtering ensured that no aliasing effects would occur after downsampling.

In order to shape the data in a way that will be suitable for our deep learning architecture,

a window of 250 milliseconds (ms) was fit around each key press, with 75 ms of samples recorded before the key press and 175 ms of samples recorded after. This window size and setup allowed us to have a fixed signal duration of 250 samples for each key press while also ensuring that the taken window for each key will not coincide with the data from a subsequent key press as much as possible. After this operation, we were able to obtain an EMG signal with 32 channels and 250 samples for each key press in our dataset.

3.6 Training and classification

In order to train our model, the data obtained from a single session was first split to training, validation and test sets, with 80% of the data reserved for training, 10% reserved for validation and the remaining 10% reserved for testing. After this step, the data was fed to our implementation of the EEGNet architecture. The model was trained using the Adam optimizer with the cross-entropy loss function. Training was performed for 1000 epochs with a batch size of 64. After training, the model was tested on both the test set from the same data session training was performed on and the data obtained from other data collection sessions to see if generalization across sessions is possible.

Due to the different structure of our data compared to the datasets used by Lawhern et al. in the EEGNet paper [2], we made some modifications to the original EEGNet architecture in our implementation. The main differences between our version and the original EEGNet can be summarized as follows:

- After getting the 250 (length of data window) by 32 (number of channels) data sample, a padding of 25 units to the top and 24 units to the bottom was performed. This padding ensured that the output of the first convolution matched the dimensions of the original input data.
- We removed all the batch normalization layers in our version, as we got better perfor-

mance in our experiments this way.

- Before the separable convolution layer, we applied 6 units of padding to the top and 6 units of padding to the bottom, so that the output dimension of this convolution matches the input dimension to this layer.
- We set the final dense layer to have 5 outputs, one for each finger digit.
- We changed the sizes and the amount of filters in the first temporal convolution layer and the separable convolution layer based on our hyperparameter search results.
- We changed the size of the final average pooling layer from (1,8) to (5,1).

A more detailed view of our architecture can be seen in Table 3.2. As can be seen, our implementation is quite compact, with only **2805** trainable parameters.

Table 3.2: **The full architecture of our EEGNet implementation.** The input data is first passed through a padding layer. Then, the first temporal convolution is applied. This is followed by the spatial convolution. After ELU activation, average pooling and a dropout layer with probability 0.5, another padding is applied. The result is passed through the final separable convolution layer. After another ELU activation, average pooling and dropout with probability 0.5, the resulting tensor is flattened and passed through a dense layer, which generates the probabilities for each finger digit.

Layer	# filters	Size	# params	Output Size	Activation
Input				(250,32)	
Padding		(0, 0, 25, 24)		(299,32)	
Conv2D	16	(50,1)	50*16	(16,250,32)	Linear
DepthwiseConv2D	1*16	(1,32)	32*1*16	(1*16,250,1)	Linear
Activation				(1*16,250,1)	ELU
AveragePool2D		(4,1)		(1*16,250//4,1)	
Dropout (p=0.5)				(1*16,250//4,1)	
Padding		(0, 0, 6, 6)		(1*16,250//4+12,1)	
SeparableConv2D	16	(13,1)	13*1*16+16*(1*16)	(16,250//4,1)	
Activation				(16,250//4,1)	ELU
AveragePool2D		(5,1)		(16,250//20,1)	
Dropout (p=0.5)				(16,250//20,1)	
Flatten				(16*(250//20))	
Dense	5*(16*250//20)			5	Softmax

CHAPTER 4

Results

In this section, we will cover the different experimental setups that our framework was utilized in and quantitate the performance of our model in decoding individual finger movements from one hand that occur during typing.

4.1 Within-session decoding results

For this section, the model was trained and tested with data of one subject from one recording session. The dataset from the session was divided using a 80/10/10 train/validation/test split and the test accuracy results were noted. This process was repeated for each subject and recording session. On the test dataset, an overall average accuracy of **77.64%** was observed across all subjects and data collection sessions. A more detailed view of the performance of our framework for each subject can be seen in Figure 4.1.

As seen in Figure 4.1, we are able to get high accuracy values for each subject in a within-session scenario, indicating the validity of our approach. The reason that results for subject 3 are not as accurate as the ones for the other subjects was determined to be related to the typing speed, as during the initial data collection session subject 3 typed significantly faster than the expected rate of one key press per second. During the second data collection session, the subject paid more attention to matching the requested typing speed, which resulted in an accuracy increase of 21.2%. The reason that fast typing speeds degrade performance is that in those cases, the 250 ms window used for each key press contains artifacts from the

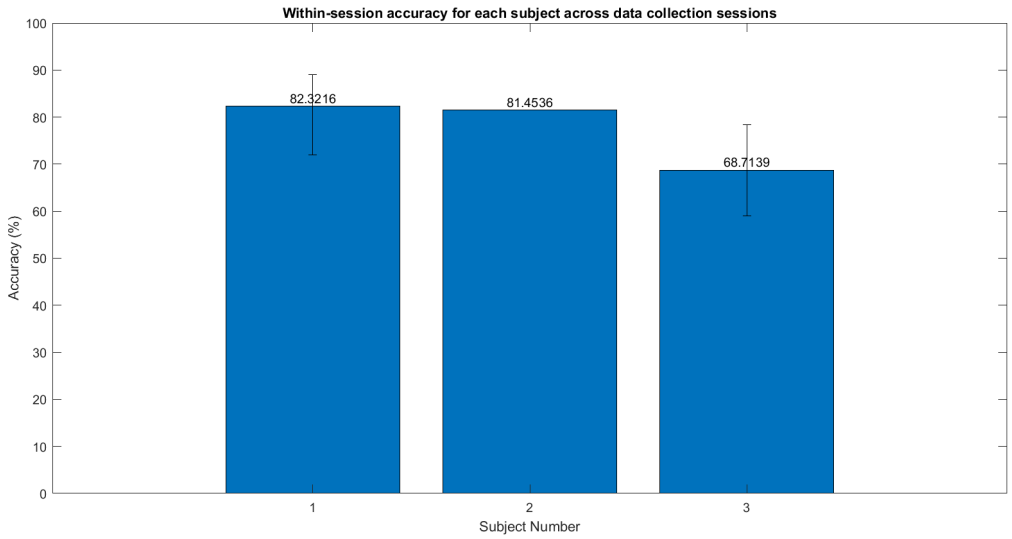


Figure 4.1: **Within-session accuracy for each subject across data collection sessions.** Bars represent the average of the test accuracies across each data collection session the subjects participated in. The error bars indicate the highest and the lowest within-session accuracies obtained on a single data collection session of a subject.

previous and following key presses. This makes it more difficult for the model to differentiate between different classes during training.

Let us now examine the nature of the errors our model makes based on the confusion matrix in Figure 4.2. As can be seen, the majority of the errors are made between adjacent pairs of finger digits. This is in line with the findings of Crouch et al. [17]. Furthermore, the model seems to have a predilection for predicting the pinky finger (also known as the little finger) even when the ground truth is not a finger digit adjacent to the pinky.

4.2 Across-session decoding results

In order to determine the performance of the created system on one subject across different data recording sessions, the model was trained on all data of a subject from one session and

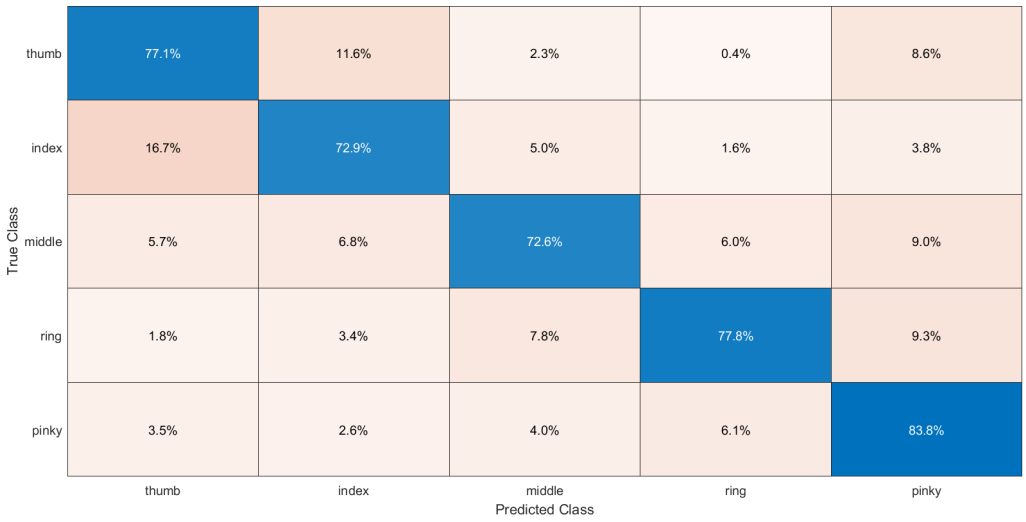


Figure 4.2: **Confusion matrix for within-session performance.** All of the ground truth labels and the model predictions for each subject and data collection session were bundled together to generate this confusion matrix. The matrix is row normalized, so that the sum of each row is 100%.

then tested on the data from all the other sessions that subject participated in. This process was repeated for all sessions of subject 1 and subject 3. Subject 2 was not included as he only participated in one session. The goal of the experiment was to see if a trained model for a single subject can successfully be used without any parameter tuning on sessions from different days.

From Figure 4.3, it appears that the performance of the model is slightly above chance when tested on a dataset from a different session than it was trained on, even if the subject stays the same. Inter-session variability has been known to be a significant factor in reduced performance for EMG applications due to electrode displacement and the distance of the data across different sessions in the input space [33]. The results show that our TypEMG framework is also susceptible to such perturbations. Therefore, in a practical setting, a training or calibration phase before classification must be conducted for each session of a

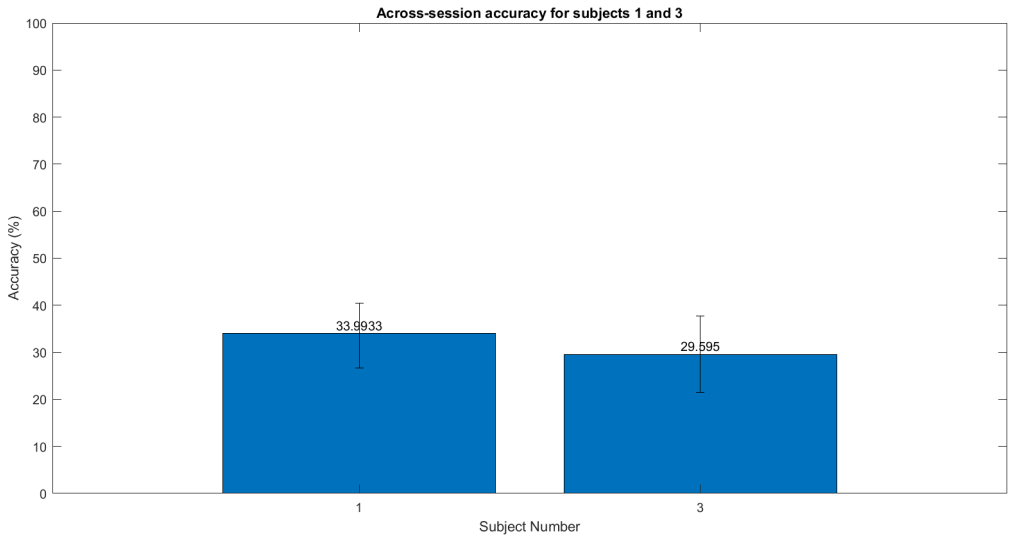


Figure 4.3: **Across-session accuracy for subjects 1 and 3.** Bars represent the average of the test accuracies on data from the sessions the model was not trained on and the error bars indicate the highest and the lowest across-session accuracies obtained for each subject.

subject in order to get satisfactory results.

4.3 Time requirement for classification

As our end goal is to build a real-time system for individuals with amputations or tetraplegia that will allow them to interface with computers through typing, we need to ensure that the framework we created is able to perform classification without any delay. To test this capability, we trained a model with only 5% of the available dataset, making up a training set of 1161 samples. After training, we tested the model on the remaining 20000 samples and measured the amount of time it took to make a prediction for all of the test data.

In total, the time it took to perform filtering on all 21161 samples was 86.82 seconds, indicating an average time requirement of **4.06 ms** per sample. For average referencing and downsampling, the time needed to process all of the samples was 2.006291 seconds,

meaning an average time of **94.81 microsecond** (µs) per sample. For classification, it took our model 0.1128695011 seconds to make a prediction for 20000 samples with our hardware setup, indicating an average of **5.64** µs per sample. Overall, the amount of time needed to preprocess, downsample and classify a sample was found to be **4.16 ms**. Since an operation that takes less than 100 ms of processing time can be thought as capable of real-time usage [17], we can state that our approach is suitable for real-time operations.

4.4 Amount of data needed for calibration

As can be inferred from Sections 4.1 and 4.2, it is clear that in a practical application of our TypEMG framework a certain amount of data samples must be used to train and calibrate the model before the subject can engage in typing. In order to determine the minimum amount of data needed for within-session training to reach satisfactory performance, the within-session experiment was repeated with differing portions of the entire set reserved for training. For this experiment, 20% of the dataset collected from a session was randomly selected and reserved for testing. The remaining data was randomly sampled to create the training set. The amount of training data ranged from 10% of the overall session data to the maximum of 80% in 10% increments. Reserving the test set at the beginning and performing tests on this set alone ensured that all training data amount configurations could be accurately compared. This entire process was repeated for all of the available data collection sessions and the overall results of the experiment can be seen in Figure 4.4.

As seen in Figure 4.4, there seems to be a positive correlation between the amount of used training data and performance on the test set, with the worst performance observed when 20% of the session data is used for training and the best performance observed when all of the available data is used to train the model. Overall, it can be inferred that the biggest improvement in performance happens when 40% of the data is used for training and the performance improvements start to plateau after the training data makes up 60% of the

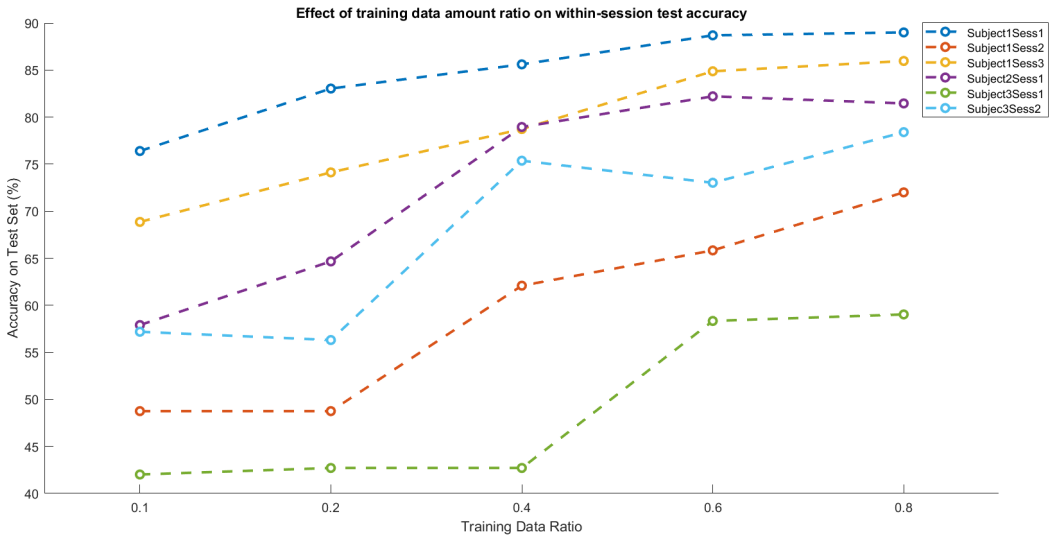


Figure 4.4: **Effect of training data amount ratio on within-session test accuracy.** The training data ratio reflects how much of the overall data for each within-session scenario was used for model training. Maximum is 0.8 as 0.2 of the data in the set was reserved for testing in the beginning. SubjectASessB in the legend refers to the performance of the TypEMG framework on data obtained from the Bth data collection session of subject A.

entire set.

After seeing the trend between training data amount percentage and performance, a new model was trained on a subset of data generated by randomly sampling 40% of the entire dataset from the first session of subject 1, as this was the data recording session that led to the highest within-session classification accuracy. After testing the model on the remaining samples of this session, an accuracy of 87.42% was observed. Instead, when 60% of the entire data from the session was used during training, the accuracy on the remaining test set jumped to 88.81%, indicating that there is not much practical difference between these training data percentages. Since 40% of a data collection session which lasts an hour is enough to get satisfactory performance, it can be inferred that in an actual practical setting, a mere **24 minutes** of data collection is enough to calibrate the model and perform

Figure 4.5: **Alternative placement of the electrodes.** To determine the effect of electrode orientation, the alignment of the electrodes was changed to be perpendicular to the anterior superficial muscles of the forearm in a horizontal position. To minimize movement artifacts, the electrodes were secured in place with adhesive tape.

accurate finger digit decoding for that session. This conclusion not only shows the validity of our EMG classification framework but also demonstrates the practical applicability of our system. The confusion matrices obtained during this section of the experiment can be found in the Appendix section of the thesis.

4.5 Electrode orientation

So far, the electrodes used during data collection were placed vertically, parallel to the anterior superficial muscles of the forearm. As an alternative, an additional data collection session was performed with subject 1 where the electrodes were placed perpendicularly to

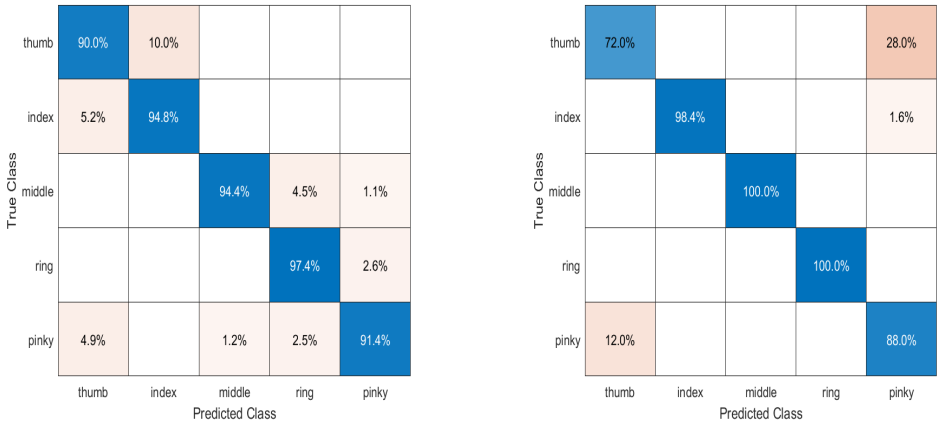


Figure 4.6: **Effect of electrode orientation on within-session accuracy for subject 1.** a) Within-session confusion matrix of data from subject 1 session 1. b) Within-session confusion matrix of the session with horizontal electrode placement. Session 1 of subject 1 was chosen for comparison as this set had the highest within-session accuracy for subject 1. Matrices are row normalized, so that the sum of each row is 100%.

the muscles, in a horizontal orientation on the forearm. The main motivation behind this experiment was the fact that action potentials travel longitudinally inside the muscle fibers [34]. This implies that if we record many points along the longitudinal axis, the signals we get will be highly correlated as the electrodes will be reading the same action potential propagating along the muscle fibers. Instead, by changing the electrode orientation we aimed to decrease the correlation of the channels and capture information about more muscle groups.

The new placement of the electrodes can be seen in Figure 4.5. With this configuration, a new data collection session was run for approximately 1 hour, resulting in a total of 3571 collected key presses. Afterwards, the same train/validation/test split, filtering and model training steps were performed as described in the Methods section of the thesis to determine the performance of the model.

On the data obtained with the alternative electrode orientation, an accuracy of **92.1%**

was observed across all finger digits, indicating an approximate **3%** increase in accuracy when compared to the performance on the data collection session of subject 1 with the highest within-session accuracy. The confusion matrices of the two compared sessions can be seen in Figure 4.6. From the figure, it can be inferred that when the electrodes are placed horizontally, the errors of the model are concentrated between the pinky and thumb digits, whereas when the electrodes are placed vertically, in addition to the errors between pinky and thumb the model also seems to confuse adjacent finger digits.

Based on the results of this experiment, the reader is encouraged to increase the coverage across different relevant muscles instead of focusing on a single muscle group when building a system to decode finger digit movement from EMG signals.

4.6 Electrode channel amount

Until this point, the 4-8-L HD-EMG electrodes provided by TMSi were used for recording the sEMG signals that occur during typing. In order to determine the effect having more channels per electrode would have on model performance, an additional data collection session was conducted with subject 3. This time, the TMSi 8-8-L HD-EMG electrodes were utilized to record the EMG activity. These electrodes have the same specifications as the 4-8-L HD-EMG electrodes, with the main difference being that the 8-8-L HD-EMG has 64 channels instead of 32 [30]. The 64 channel electrodes were placed parallel to the to the anterior superficial muscles of the forearm as was done in the Methods section of the thesis.

The data collection session lasted approximately one hour, with a total of 4603 key presses collected. The preprocessing, train/validation/test split and model training steps described in the Methods section were performed to decode finger digits. To compare the effect of increasing the channel amount, another dataset with less number of channels was created by concatenating the readings from the 32 channels in the middle of the 8-8-L HD-EMG electrode grid. Afterwards, the same training process done in the Methods section was

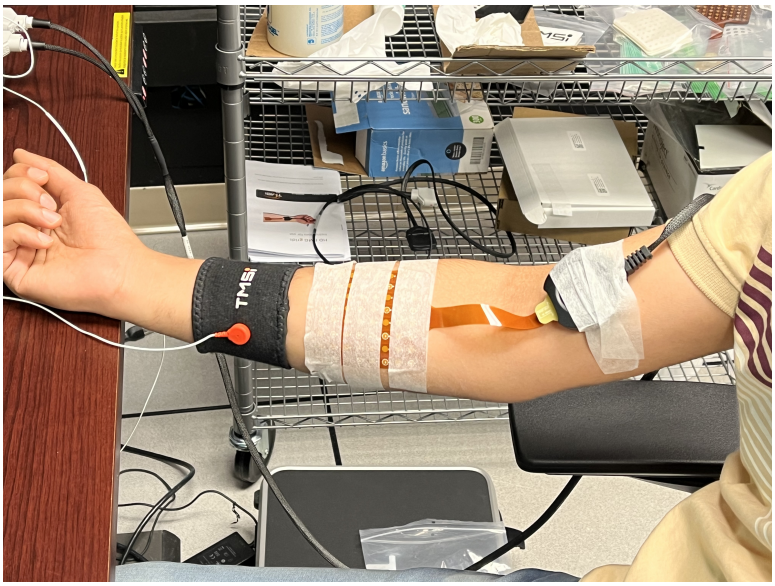
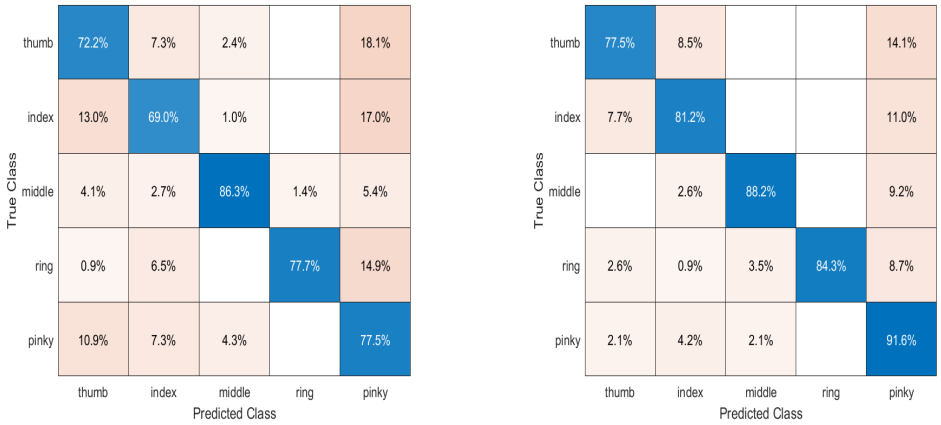


Figure 4.7: **Positioning of the 64 channel electrodes.** The 64 channel electrodes were placed on subject 3 in parallel with the anterior superficial muscles of the forearm in the same manner as was done in the Methods section. Adhesive tape was used to secure the electrodes in place.

performed with this reduced feature set to establish a baseline for comparison.

The comparison in within-session decoding accuracy between using only the 32 electrode channels in the middle of the electrode grid and using all 64 channels of the grid can be found in Figure 4.8. Overall, an accuracy of **84.5%** was obtained when 64 channels were utilized, an increase of **8.4%** when compared to using only the 32 channels in the middle of the grid. Based on these results, it can also be understood that using 64 channels leads to a performance improvement of **6.1%** when compared to the best within-session decoding accuracy of subject 3 on data recorded with the 4-8-L HD-EMG 32 channel electrodes.

The results of this experiment indicate that increasing the amount of channels, thus increasing the area covered during typing results in better decoding performance. However, when considering the increased cost per session due to having more channels on the electrode grid and the fact that placing the electrodes horizontally leads to similar performance



(a) (b)

Figure 4.8: **Effect of electrode channel amount on within-session accuracy for subject 3.** a) Within-session confusion matrix of data from subject 3 when the 32 electrode channels in the middle of the grid are used. b) Within-session confusion matrix of data from subject 3 when all 64 electrode channels are used. Matrices are row normalized, so that the sum of each row is 100%.

improvements, it may be more suitable for future experiments to first simply try using electrodes with less channels in a horizontal position as done in Section 4.5 and seeing if the framework is able to achieve satisfactory performance.

CHAPTER 5

Conclusions

In this thesis, we showed our TypEMG framework to acquire, process and classify sEMG signals obtained from the forearm to decode finger movements that occur during typing, with the end goal to build a system that will allow individuals with amputations or tetraplegia to interface with computers seamlessly via typing in real-time. For this purpose, we established a synchronized communication between the local workstation and the TMSi SAGA amplifier via a TCP connection to acquire sEMG recordings from the TMSi 4-8-L HD-EMG surface electrode sensors, devised a PyGame that guided the test subjects during data collection and acquired data from three different subjects over six individual sessions, each session lasting approximately one hour. We performed filtering and average referencing on the acquired signal to reduce noise elements, performed downsampling to reduce computational load and windowed the signal around the key presses to prepare a dataset with uniform sample length, where each sample corresponds to a specific key press. Afterwards, we implemented a modified version of the EEGNet architecture [2] and trained the model to perform feature extraction and classification of finger digits. Using this approach, we were able to build a system that can accurately decode the subtle finger movements that occur on one hand during typing from sEMG signals, with applicability in a real-time operation setting.

The results show that our system is able to get an average accuracy of **77.64%** across all test subjects in a within-session classification scenario. In addition, our system requires only **24 minutes** of data collection to perform calibration for a new decoding session and get satisfactory performance. Furthermore, with a mere **4.16 ms** time requirement to preprocess

and classify a data sample, our system can be said to be eligible for real-time usage.

Experiments with different electrode configurations show that focusing on a larger variety of muscles involved during typing as opposed to attempting to isolate a smaller muscle group leads to improved classification results. Furthermore, it has been shown that increasing the amount of channels of the electrodes, thus increasing the surface area the electrodes cover, also leads to considerable improvements. Therefore, it is suggested that continuations of our work utilize electrodes with an increased number of channels and more coverage of the forearm muscles.

In addition to the experiment findings, it is hypothesized that utilizing language information in the model can be an important research direction to follow. This can be achieved by incorporating recurrent networks in the model architecture, as was done by Crouch et al. [17] or by making use of pretrained language models. Finally, improving the signal preprocessing scheme to extract MUAPs directly from the sEMG signals is expected to improve classification performance. For this purpose, the work of Holobar and Zazula [21] and Negro et al. [35] can be examined as starting points. After an algorithm to decompose sEMG signals to the MUAPs they are made of is developed, the extracted MUAPs can be provided as input for finger digit classification. Finally, once the desired performance levels are met on healthy subjects, the designed framework must be tested on data from amputated or disabled individuals to ensure proper operation.

This concludes the work done for the TypEMG project throughout the duration of my graduate studies. Although the work is not complete and we are still trying out different approaches to improve our results, our work shows that performing classification of the subtle finger movements that occur naturally during typing by using non-invasive sEMG signals is possible. We hope that our framework will serve as a potential starting point for sEMG applications with a focus on typing while bringing us one step closer to having a fully functioning real-time system that will allow individuals with amputations or disabilities to effectively interface with computers in their day-to-day lives.

APPENDIX A

Additional Confusion Matrices

For the sake of completeness, this appendix includes all of the confusion matrices generated during the within-session, across-session and limited training data experiments.

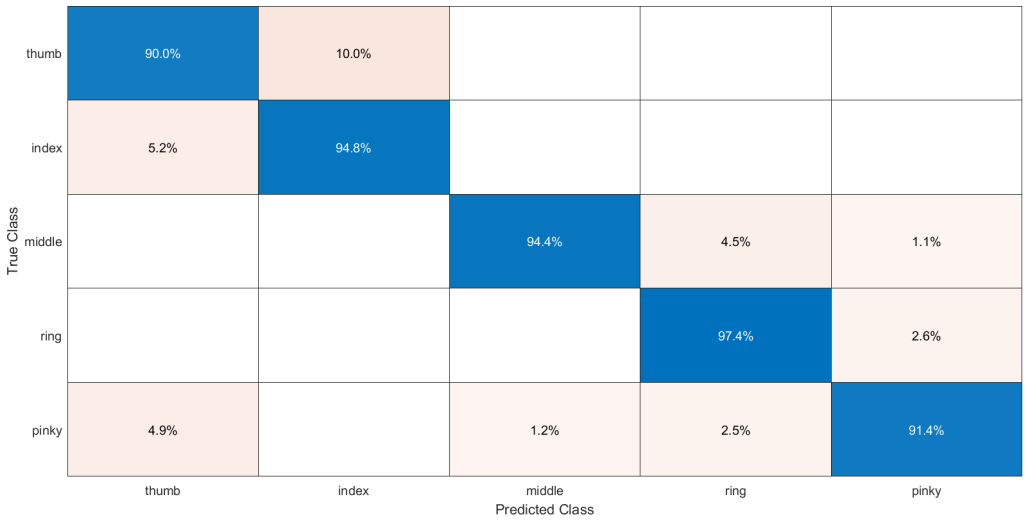


Figure A.1: **Confusion matrix for within-session performance on subject 1 session 1 data.** Matrix is row normalized, so that the sum of each row is 100%.

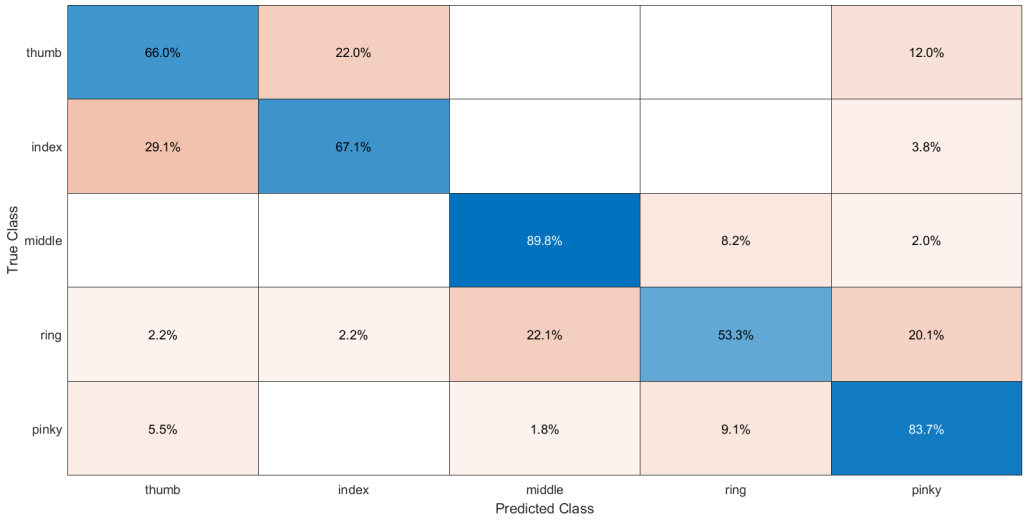


Figure A.2: **Confusion matrix for within-session performance on subject 1 session 2 data.** Matrix is row normalized, so that the sum of each row is 100%.

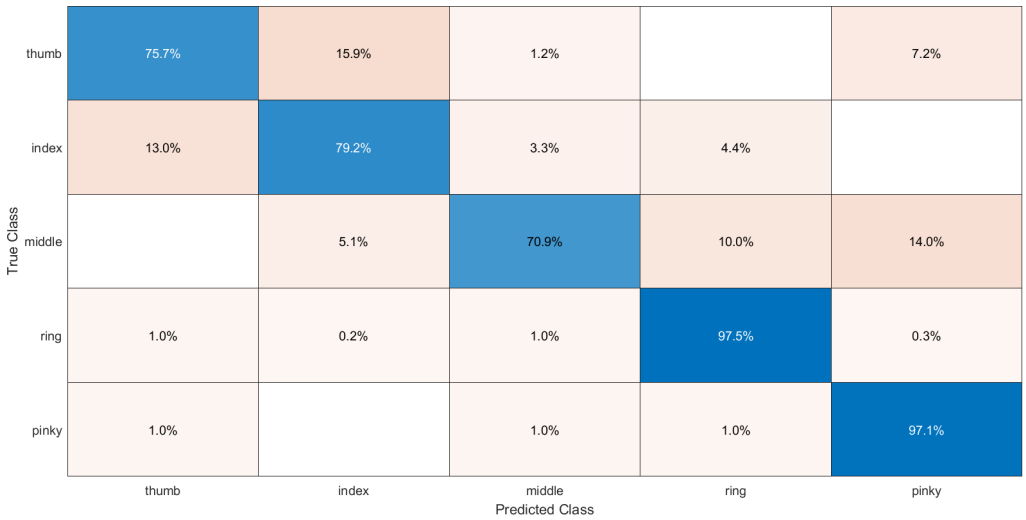


Figure A.3: **Confusion matrix for within-session performance on subject 1 session 3 data.** Matrix is row normalized, so that the sum of each row is 100%.

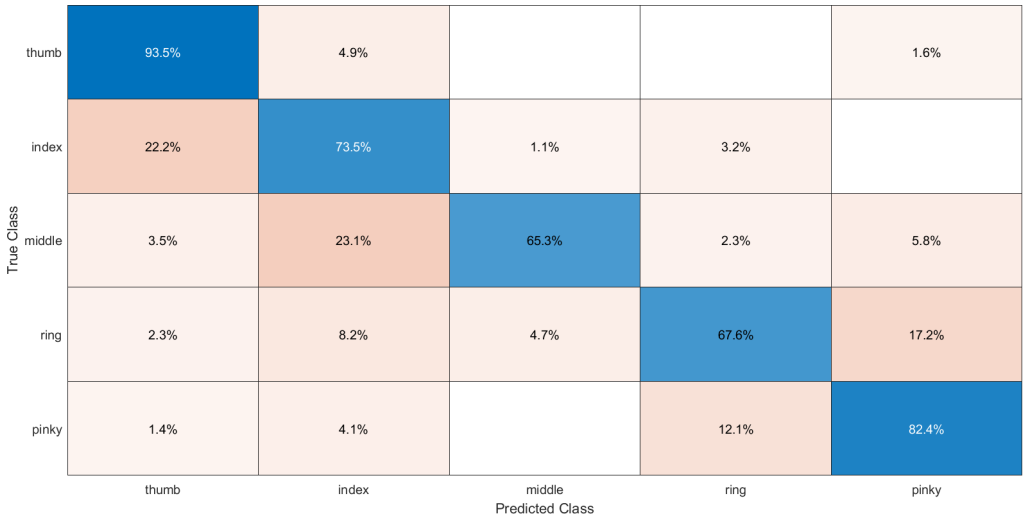


Figure A.4: **Confusion matrix for within-session performance on subject 2 session 1 data.** Matrix is row normalized, so that the sum of each row is 100%.

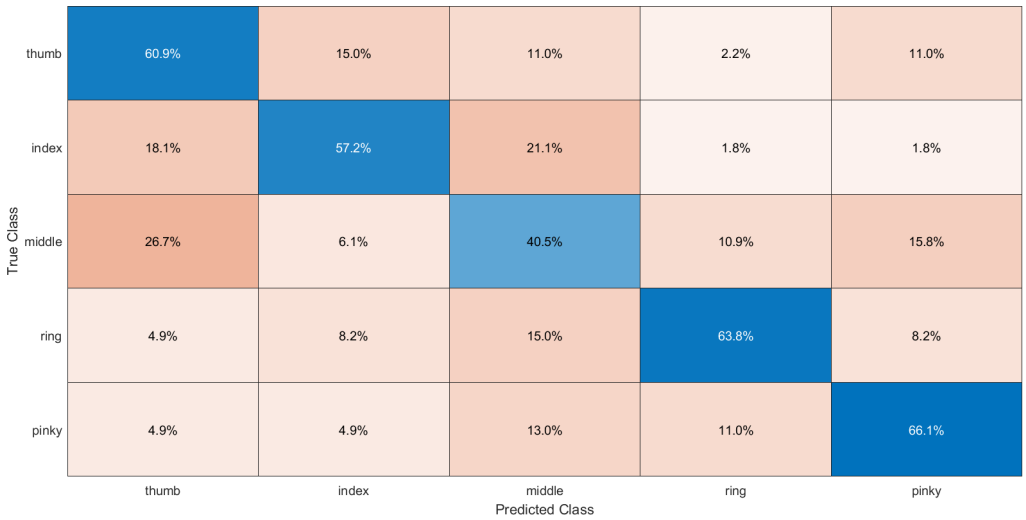


Figure A.5: **Confusion matrix for within-session performance on subject 3 session 1 data.** Matrix is row normalized, so that the sum of each row is 100%.

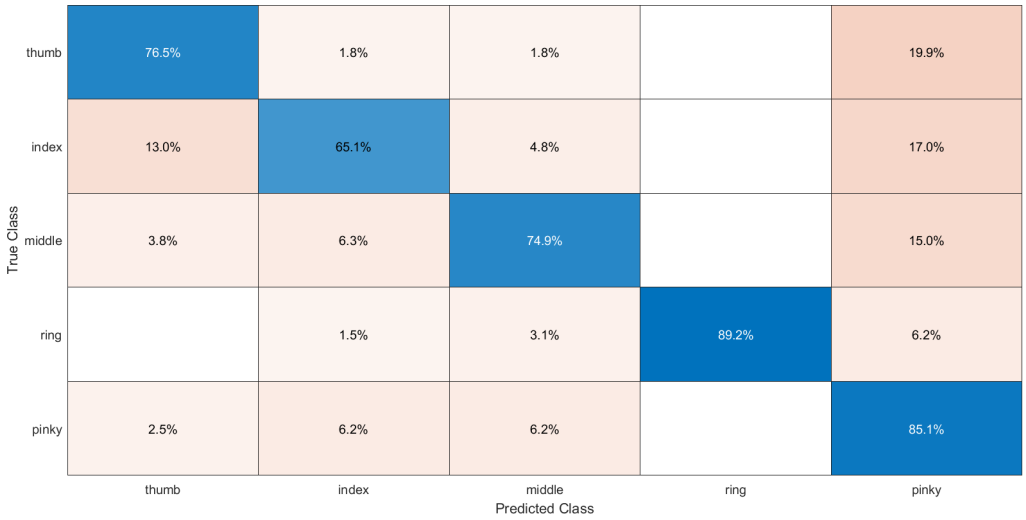


Figure A.6: **Confusion matrix for within-session performance on subject 3 session 2 data.** Matrix is row normalized, so that the sum of each row is 100%.

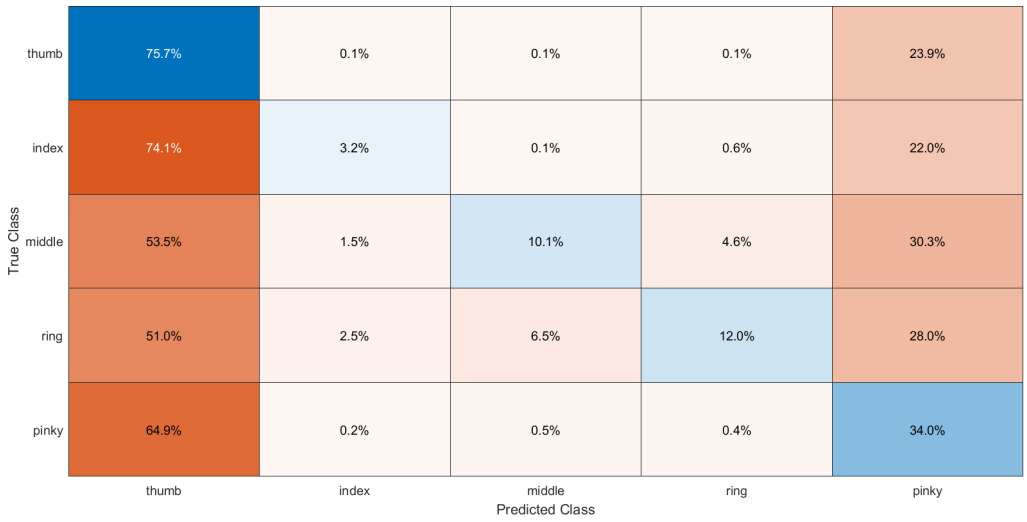


Figure A.7: **Confusion matrix for across-session performance when subject 1 session 1 data used for training.** The model was trained on the session 1 data of subject 1 and tested on the data from the remaining sessions of subject 1. Matrix is row normalized, so that the sum of each row is 100%.

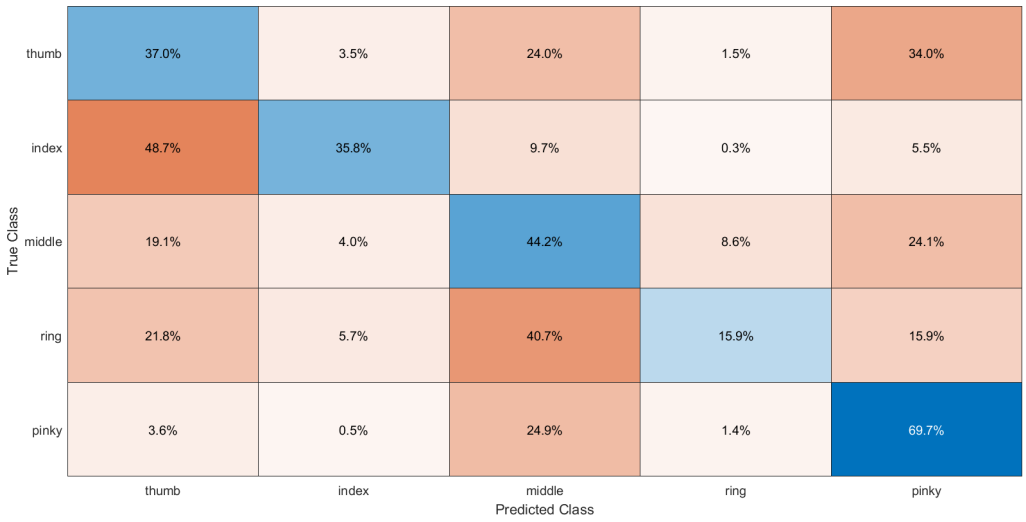


Figure A.8: **Confusion matrix for across-session performance when subject 1 session 2 data used for training.** The model was trained on the session 2 data of subject 1 and tested on the data from the remaining sessions of subject 1. Matrix is row normalized, so that the sum of each row is 100%.

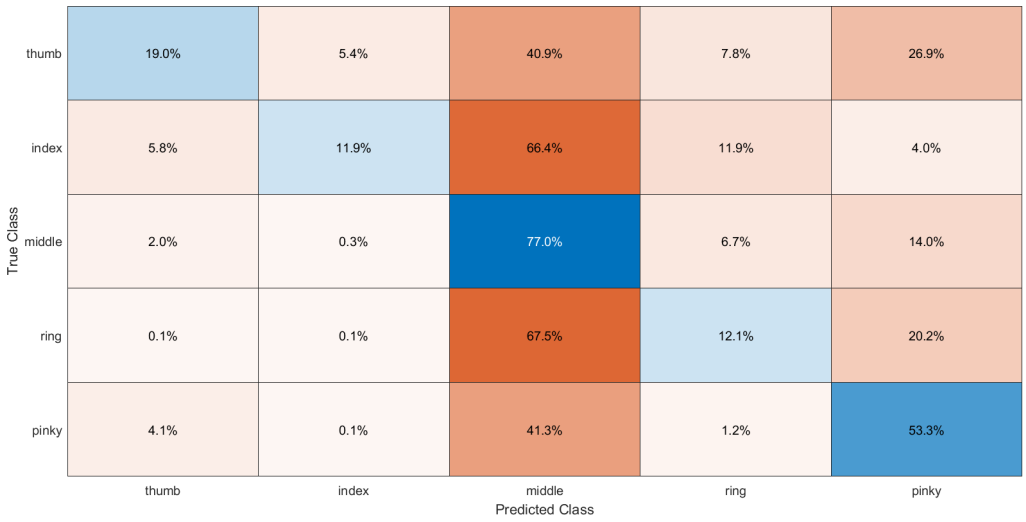


Figure A.9: **Confusion matrix for across-session performance when subject 1 session 3 data used for training.** The model was trained on the session 3 data of subject 1 and tested on the data from the remaining sessions of subject 1. Matrix is row normalized, so that the sum of each row is 100%.

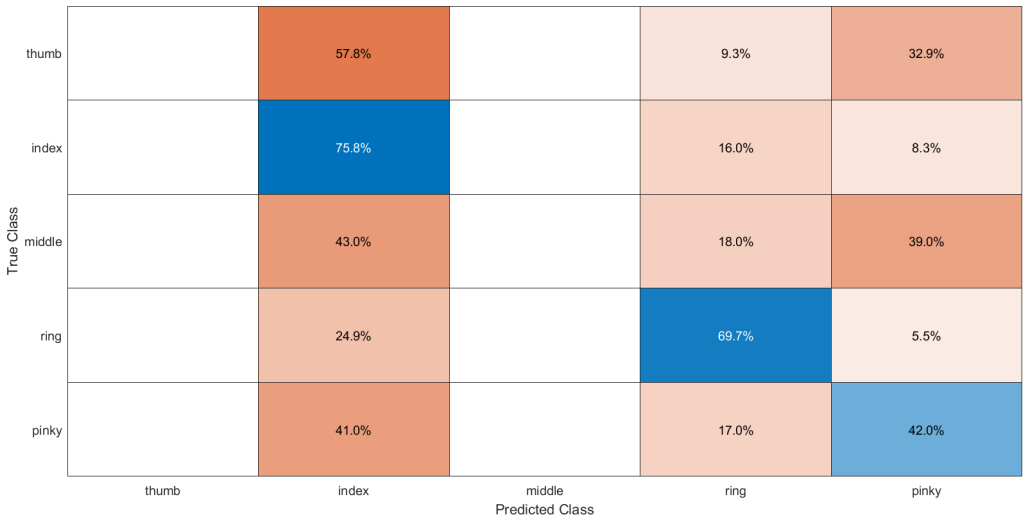


Figure A.10: **Confusion matrix for across-session performance when subject 3 session 1 data used for training.** The model was trained on the session 1 data of subject 3 and tested on the data from the remaining session of subject 3. Matrix is row normalized, so that the sum of each row is 100%.

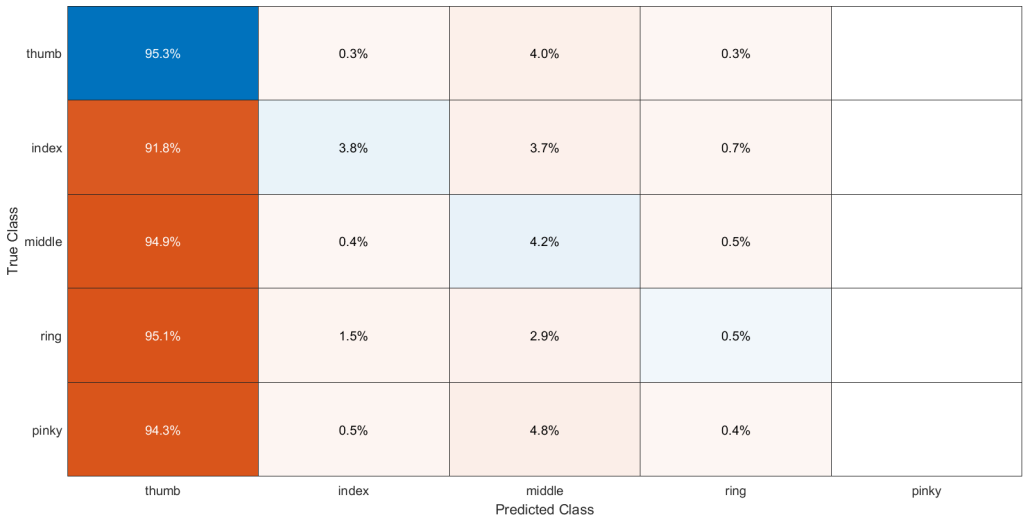


Figure A.11: **Confusion matrix for across-session performance when subject 3 session 2 data used for training.** The model was trained on the session 2 data of subject 3 and tested on the data from the remaining session of subject 3. Matrix is row normalized, so that the sum of each row is 100%.

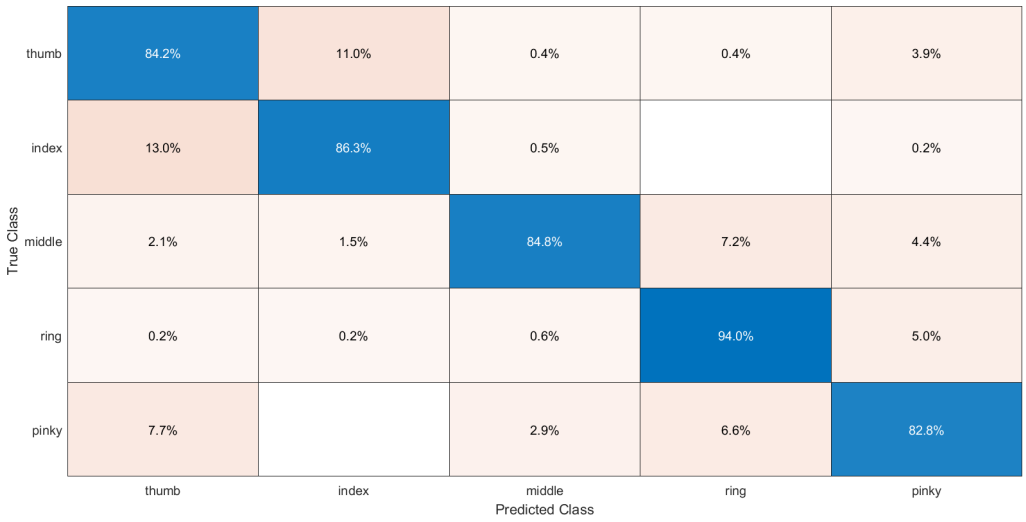


Figure A.12: **Confusion matrix for within-session performance when 40% of subject 1 session 1 data used for training.** The model was trained with 40% of the data from session 1 of subject 1 and tested on the remainder. Matrix is row normalized, so that the sum of each row is 100%.

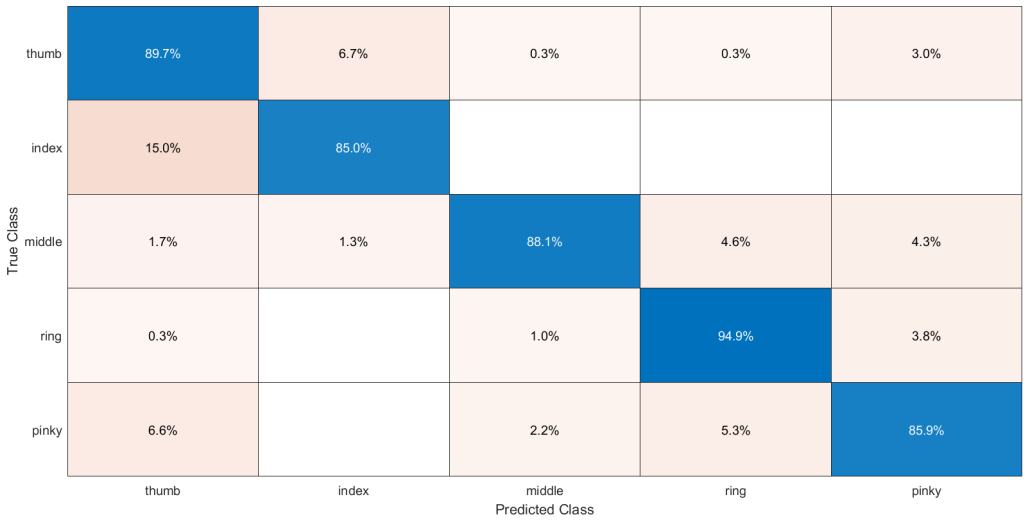


Figure A.13: **Confusion matrix for within-session performance when 60% of subject 1 session 1 data used for training.** The model was trained with 60% of the data from session 1 of subject 1 and tested on the remainder. Matrix is row normalized, so that the sum of each row is 100%.

REFERENCES

[1] C. L. Gooch and R. Henderson, "Overview of electromyography," *UpToDate Inc.*, 2023. [Online]. Available: https://www.uptodate.com/contents/overview-of-electromyography

[2] V. J. Lawhern, A. J. Solon, N. R. Waytowich, S. M. Gordon, C. P. Hung, and B. J. Lance, "Eegnet: a compact convolutional neural network for eeg-based brain–computer interfaces," *Journal of neural engineering*, vol. 15, no. 5, p. 056013, 2018.

[3] B. Jarosiewicz, A. A. Sarma, D. Bacher, N. Y. Masse, J. D. Simeral, B. Sorice, E. M. Oakley, C. Blabe, C. Pandarinath, V. Gilja *et al.*, "Virtual typing by people with tetraplegia using a self-calibrating intracortical brain-computer interface," *Science translational medicine*, vol. 7, no. 313, pp. 313ra179–313ra179, 2015.

[4] C. Pandarinath, P. Nuyujukian, C. H. Blabe, B. L. Sorice, J. Saab, F. R. Willett, L. R. Hochberg, K. V. Shenoy, and J. M. Henderson, "High performance communication by people with paralysis using an intracortical brain-computer interface," *Elife*, vol. 6, p. e18554, 2017.

[5] F. R. Willett, D. T. Avansino, L. R. Hochberg, J. M. Henderson, and K. V. Shenoy, "High-performance brain-to-text communication via handwriting," *Nature*, vol. 593, no. 7858, pp. 249–254, 2021.

[6] J. E. Ting, A. Del Vecchio, D. Sarma, N. Verma, S. C. Colachis 4th, N. V. Annetta, J. L. Collinger, D. Farina, and D. J. Weber, "Sensing and decoding the neural drive to paralyzed muscles during attempted movements of a person with tetraplegia using a sleeve array," *Journal of Neurophysiology*, vol. 126, no. 6, pp. 2104–2118, 2021.

[7] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of emg signal analysis: detection, processing, classification and applications," *Biological procedures online*, vol. 8, pp. 11–35, 2006.

[8] K.-J. You, K.-W. Rhee, and H.-C. Shin, "Finger motion decoding using emg signals corresponding various arm postures," *Experimental neurobiology*, vol. 19, no. 1, p. 54, 2010.

[9] C. Amma, T. Krings, J. Böer, and T. Schultz, "Advancing muscle-computer interfaces with high-density electromyography," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015, pp. 929–938.

[10] T. S. Saponas, D. S. Tan, D. Morris, and R. Balakrishnan, "Demonstrating the feasibility of using forearm electromyography for muscle-computer interfaces," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 515–524.

[11] J. Too, A. Abdullah, N. M. Saad, N. M. Ali, and T. Zawawi, "Featureless emg pattern recognition based on convolutional neural network," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 14, no. 3, pp. 1291–1297, 2019.

[12] W. Li, P. Shi, and H. Yu, "Gesture recognition using surface electromyography and deep learning for prostheses hand: state-of-the-art, challenges, and future," *Frontiers in neuroscience*, vol. 15, p. 621885, 2021.

[13] A. Mora Rubio, J. A. Alzate Grisales, R. Tabares-Soto, S. Orozco-Arias, C. F. Jiménez Varón, and J. I. Padilla Buriticá, "Identification of hand movements from electromyographic signals using machine learning," 2020. [Online]. Available: https://doi.org/10.20944/preprints202002.0443.v1

[14] M. Cognolato, M. Atzori, D. Faccio, C. Tiengo, F. Bassetto, R. Gassert, and H. Muller, "Hand gesture classification in transradial amputees using the myo armband classifier," in *2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob)*. IEEE, 2018, pp. 156–161.

[15] F. V. Tenore, A. Ramos, A. Fahmy, S. Acharya, R. Etienne-Cummings, and N. V. Thakor, "Decoding of individuated finger movements using surface electromyography," *IEEE transactions on biomedical engineering*, vol. 56, no. 5, pp. 1427–1434, 2008.

[16] E. F. Melcer, M. T. Astolfi, M. Remaley, A. Berenzweig, and T. Giurgica-Tiron, "Ctrl-labs: Hand activity estimation and real-time control from neuromuscular signals," in *Extended Abstracts of the 2018 CHI Conference on Human Factors in Computing Systems*, 2018, pp. 1–4.

[17] M. S. Crouch, M. Zheng, and M. S. Eggleston, "Natural typing recognition via surface electromyography," *arXiv preprint arXiv:2109.10743*, 2021.

[18] W. Yu, R. Kato, F. Fabio, H. Yokoi, and Y. Kakazu, "An emg keyboard for forearm amputees," *Applied Bionics and Biomechanics*, vol. 1, no. 1, pp. 33–43, 2003.

[19] M. Bear, B. Connors, and M. A. Paradiso, *Neuroscience: exploring the brain, enhanced edition*. Jones & Bartlett Learning, 2020.

[20] C.-N. H. C.-H. Chen and H.-Y. Chung, "The review of applications and measurements in facial electromyography," *Journal of Medical and Biological Engineering*, vol. 25, no. 1, pp. 15–20, 2004.

[21] A. Holobar and D. Zazula, "Multichannel blind source separation using convolution kernel compensation," *IEEE Transactions on Signal Processing*, vol. 55, no. 9, pp. 4487–4496, 2007.

[22] S. Shahid, J. Walker, G. M. Lyons, C. A. Byrne, and A. V. Nene, "Application of higher order statistics techniques to emg signals to characterize the motor unit action potential," *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 7, pp. 1195–1209, 2005.

[23] M. Mohr, T. Schön, V. Von Tscharner, and B. M. Nigg, "Intermuscular coherence between surface emg signals is higher for monopolar compared to bipolar electrode configurations," *Frontiers in physiology*, vol. 9, p. 566, 2018.

[24] S. Gordon, V. Lawhern, A. Passaro, and K. McDowell, "Informed decomposition of electroencephalographic data," *Journal of neuroscience methods*, vol. 256, pp. 41–55, 2015.

[25] X. Wang, M. Hersche, B. Tömekce, B. Kaya, M. Magno, and L. Benini, "An accurate eegnet-based motor-imagery brain–computer interface for low-power edge computing," in *2020 IEEE international symposium on medical measurements and applications (MeMeA)*. IEEE, 2020, pp. 1–6.

[26] J. van Dijk, D. F. Stegeman, M. J. Zwarts, and J. Blok, "Motor unit number estimation with high-density surface emg: principles and implications," in *Supplements to Clinical Neurophysiology*. Elsevier, 2009, vol. 60, pp. 105–118.

[27] B. Kim, L. Kim, Y.-H. Kim, and S. K. Yoo, "Cross-association analysis of eeg and emg signals according to movement intention state," *Cognitive Systems Research*, vol. 44, pp. 1–9, 2017.

[28] J. J. Bird, J. Kobylarz, D. R. Faria, A. Ekárt, and E. P. Ribeiro, "Cross-domain mlp and cnn transfer learning for biological signal processing: Eeg and emg," *IEEE Access*, vol. 8, pp. 54 789–54 801, 2020.

[29] *TMSi User Manual Saga*, Twente Medical Systems International, 2019. [Online]. Available: https://info.tmsi.com/hubfs/92-1000-0201-0-1-0_User_Manual_SAGA-EN-rev1.pdf

[30] *TMSi Data Sheet Textile HD-EMG Grid*, Twente Medical Systems International, 2022. [Online]. Available: https://info.tmsi.com/hubfs/Knowledge base/datasheet textile grids/92-0525-2032-0003-EN-2-0 Data Sheet Textile HD-EMG Grid_Rev2.pdf

[31] S. B. I. Laboratory, "Licorice - linux comodular realtime interactive computation engine," https://github.com/bil/licorice, 2023.

[32] G. G. Simoneau, R. W. Marklin, and J. E. Berman, "Effect of computer keyboard slope on wrist position and forearm electromyography of typists without musculoskeletal disorders," *Physical Therapy*, vol. 83, no. 9, pp. 816–830, 2003.

[33] C. Castellini and P. Van Der Smagt, "Surface emg in advanced hand prosthetics," *Biological cybernetics*, vol. 100, pp. 35–47, 2009.

[34] G. S. Posterino, G. D. Lamb, and D. G. Stephenson, "Twitch and tetanic force responses and longitudinal propagation of action potentials in skinned skeletal muscle fibres of the rat," *The Journal of Physiology*, vol. 527, no. 1, pp. 131–137, 2000.

[35] F. Negro, S. Muceli, A. M. Castronovo, A. Holobar, and D. Farina, "Multi-channel intramuscular and surface emg decomposition by convolutive blind source separation," *Journal of neural engineering*, vol. 13, no. 2, p. 026027, 2016.