HYBRID WEARABLE SIGNAL PROCESSING/LEARNING VIA DEEP NEURAL NETWORKS

Soheil Zabihi

A Thesis
in
The Department
of
Electrical and Computer Engineering (ECE)

Presented in Partial Fulfillment of the Requirements
For the Degree of Doctor of Philosophy
Concordia University
Montréal, Québec, Canada

June 2023
© Soheil Zabihi, 2023

Concordia University
School of Graduate Studies

This is to certify that the thesis prepared

By: **Soheil Zabihi**

Entitled: **Hybrid Wearable Signal Processing/Learning via Deep Neural Networks**

and submitted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. Sivakumar Narayanswamy	Chair
Dr. Soosan Beheshti	External Examiner
Dr. Nizar Bouguila	External to Program
Dr. Hassan Rivaz	Examiner
Dr. Wei-Ping Zhu	Examiner
Dr. Amir Asif	Supervisor
Dr. Arash Mohammadi	Supervisor

Approved by

Dr. Jun Cai, Graduate Program Director

05/26/2023

Dr. Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Hybrid Wearable Signal Processing/Learning via Deep Neural Networks

Soheil Zabihi, Ph.D.
Concordia University, 2023

Wearable technologies are gaining considerable attention in recent years as a potential post-smartphone platform with several applications of significant engineering importance. Wearable technologies are expected to become more prevalent in a variety of areas, including modern healthcare practices, robotic prosthesis control, Artificial Reality (AR) and Virtual Reality (VR) applications, Human Machine Interface/Interaction (HMI), and remote support for patients and chronically ill patients at home. The emergence of wearable technologies can be attributed to the advancement of flexible electronic materials; the availability of advanced cloud and wireless communication systems, and; the Internet of Things (IoT) coupled with high demand from the tech-savvy population and the elderly population for healthcare management. Wearable devices in the healthcare realm gather various biological signals from the human body, among which Electrocardiogram (ECG), Photoplethysmogram (PPG), and surface Electromyogram (sEMG), are the most widely non-intrusive monitored signals. Utilizing these widely used non-intrusive signals, the primary emphasis of the proposed dissertation is on the development of advanced Machine Learning (ML), in particular Deep Learning (DL), algorithms to increase the accuracy of wearable devices in specific tasks. In this context and in the first part, using ECG and PPG biosignals, we focus on development of accurate subject-specific solutions for continuous and cuff-less Blood Pressure (BP) monitoring. More precisely, a deep learning-based framework known as BP-Net is proposed for predicting continuous upper and lower bounds of blood pressure, respectively, known as Systolic BP (SBP) and Diastolic BP (DBP). Furthermore, by capitalizing on the fact that datasets used in recent literature are not unified and properly defined, a unified dataset is constructed from the

MIMIC-I and MIMIC-III databases obtained from PhysioNet. In the second part, we focus on hand gesture recognition utilizing sEMG signals, which have the potential to be used in the myoelectric prostheses control systems or decoding Myo Armbands data to interpret human intent in AR/VR environments. Capitalizing on the recent advances in hybrid architectures and Transformers in different applications, we aim to enhance the accuracy of sEMG-based hand gesture recognition by introducing a hybrid architecture based on Transformers, referred to as the Transformer for Hand Gesture Recognition (TraHGR). In particular, the TraHGR architecture consists of two parallel paths followed by a linear layer that acts as a fusion center to integrate the advantage of each module. The ultimate goal of this work is to increase the accuracy of gesture classifications, which could be a major step towards the development of more advanced HMI systems that can improve the quality of life for people with disabilities or enhance the user experience in AR/VR applications. Besides improving accuracy, decreasing the number of parameters in the Deep Neural Network (DNN) architectures plays an important role in wearable devices. In other words, to achieve the highest possible accuracy, complicated and heavy-weighted Deep Neural Networks (DNNs) are typically developed, which restricts their practical application in low-power and resource-constrained wearable systems. Therefore, in our next attempt, we propose a lightweight hybrid architecture based on the Convolutional Neural Network (CNN) and attention mechanism, referred to as Hierarchical Depth-wise Convolution along with the Attention Mechanism (HDCAM), to effectively extract local and global representations of the input. The key objective behind the design of HDCAM was to ensure its resource efficiency while maintaining comparable or better performance than the current state-of-the-art methods.

Acknowledgments

First and foremost, I would like to express my sincere gratitude to my Supervisors, Dr. Arash Mohammadi and Dr. Amir Asif for the continuous support of my Ph.D. study and research, and for their patience, motivation, enthusiasm, and immense knowledge. I deeply express my gratitude to the committee members, Dr. Soosan Beheshti, Dr. Nizar Bouguila, Dr. Hassan Rivaz, and Dr. Wei-Ping Zhu, for evaluating this dissertation and for their thoughtful feedback. I would like to express my gratitude to my parents for their incredible support, inspiration, and understanding, and last but not least, my wife whose unconditional support and patience helped me believe in myself.

Contents

List of Figures

List of Tables

Chapter 1

Overview of the Thesis

1.1 Introduction

With the advancement of flexible electronic materials, the cloud and wireless system, the Internet of Things (IoT), multimedia devices and smartphones, along with the high demand of the elderly population for health care management, the emergence of wearable medical devices has a significant and widespread impact on people's lives to monitor their personal health information in real-time [1–3]. Collecting various parameters of the human body makes these technologies strong support tools for physicians to provide continuous assessment of critical physiological parameters or to identify precursors of major adverse effects [4]. In addition, wearable technologies have the potential to be used in out-of-hospital settings, resulting in continuous monitoring solutions and real-time feedback on people's health status [4–6]. Moreover, due to the cost-effective habitual data collection in a discrete manner for longitudinal periods in any environment, the power of wearable devices as a practical and clinically useful technology to assist in the diagnosis, treatment, and care of the patient is becoming evident. As a result, they are becoming increasingly popular in several areas of modern healthcare practices, particularly in the provision of care services, ambulatory monitoring in the healthcare setting, and remote support for the rehabilitation of patients and chronically ill patients at home. To date, many wearable healthcare devices collect biometric data from the human body, such as blood glucose

levels, body temperature, electroencephalogram (EEG), electrocardiograms (ECGs), photoplethysmogram (PPG), and electromyograms (EMG), to provide valuable information in the field of healthcare and sports [3, 7–9].

1.2 Development of Wearable Technologies

The increasing usage of fitness trackers and health-related wearables, as well as the world's growing population of tech-savvy individuals, have resulted in a booming market for wearable technology in recent years. As a piece of evidence, according to the recent report from the International Data Corporation (IDC) [10] in March 2022, the worldwide wearables market set/hit a new record high in the fourth quarter of 2021, with sales reaching 171 million units, 10.8% percent higher than the same quarter in 2020. New innovations and ongoing demand for health and fitness tracking devices, as well as hearables helped the market maintain its momentum. Shipments for the full year 2021 totaled 533.6 million units, representing an increase of 20% over 2020. According to the recent report of IDC in March 2023, however, shipments for the full year 2022 were down 7.7% compared to 2021, marking the first year of decline for the category due to challenging macroeconomic conditions and difficult comparisons to the strong results of 2021. Despite the downturn, overall shipments of 492.1 million units in 2022 were well above 2020 and 2019 levels. This tempting and growing market has led the prominent industry including Apple Inc., Fitbit Inc., Samsung, Xiaomi Global Community, and Huawei Device Co., to play an important role in the development of wearable technology.

In recent years, the usage of wearable technology has experienced significant growth, driven by a variety of factors. One of the key factors that have contributed to the growth of wearable technology usage is the increasing consumer awareness of health and wellness. People are becoming more proactive about managing their health and fitness, and wearable devices are seen as an effective tool to help them achieve their goals. With the rise of chronic diseases such as diabetes and heart disease, wearable devices are also being used to monitor and manage these conditions, providing patients with greater control over their health. Furthermore, the

increasing popularity of wearable devices has led to the development of a wide range of specialized applications and services. These include fitness and wellness apps, remote patient monitoring systems, and mobile health platforms, which enable users to track their health data, communicate with healthcare professionals, and receive personalized health coaching.

Another factor that has opened up new avenues for the growth of this industry is the emergence of smart homes and the Internet of Things (IoT). With the increasing number of connected devices in the home, wearable devices can now communicate with other devices, such as smart scales and blood pressure monitors, providing a more holistic view of a user's health. Moreover, the IoT has enabled wearable devices to connect to the cloud and access advanced analytic and machine learning algorithms. This means that wearable devices can now process and analyze large amounts of health data and provide more accurate and actionable insights to users.

Another major driver of the wearable technology market is the growing demand for remote monitoring solutions, particularly in healthcare. Wearable devices have the potential to be used to monitor patients remotely, providing healthcare professionals with real-time data on a patient's health status. This is particularly useful for patients with chronic conditions, who require regular monitoring and management of their health. With the rise of telemedicine and virtual care, wearable devices are expected to play an even greater role in the future of healthcare.

In addition to the factors mentioned earlier, advancements in technology have played a significant role in the growth of wearables. Miniaturization and increased processing power have allowed for the creation of smaller, more powerful wearable devices that can collect and analyze a greater amount of data. This has led to the development of wearable devices with advanced sensors, which enable the collection of more detailed and accurate data.

As the market for wearable technology continues to grow, devices are becoming increasingly sophisticated, offering new features and functionalities that enhance their usefulness and ease of use. For example, wearable devices can now track not just basic metrics like steps taken and calories burned, but also more complex data such as heart rate variability, blood oxygen levels, and even sleep quality. Wearable devices are also

becoming more user-friendly, with improved design and ease of use, making them more appealing to a broader range of users.

The cloud and wireless systems have played a crucial role in the evolution of wearable medical devices, as they allow for real-time monitoring of patient health data by healthcare providers. With the integration of Artificial Intelligence (AI) and Machine Learning (ML) algorithms, healthcare providers can analyze the collected data more effectively, identify patterns, and make timely interventions to prevent or manage chronic conditions. AI and ML algorithms can also predict the likelihood of future health events, such as hospital readmissions or emergency room visits based on patient data, enabling healthcare providers to take proactive measures to prevent such events from occurring. Additionally, the use of cloud and wireless systems has made it easier for healthcare providers to access patient data from anywhere, at any time, which can be particularly useful in emergency situations. Overall, the combination of cloud and wireless systems, along with AI and ML, has the potential to revolutionize healthcare delivery, making it more patient-centered, efficient, and effective, ultimately leading to better patient outcomes and reduced healthcare costs.

The widespread adoption of multimedia devices and smartphones has created a massive market for wearable medical devices that can communicate with these devices, offering patients a convenient way to monitor their health data and share it with healthcare providers. The increasing computation power of these devices has allowed for the development of more sophisticated wearable medical devices, such as smartwatches or fitness trackers, which can collect and analyze various health metrics, such as heart rate, blood pressure, and sleep patterns, among others. This has enabled patients to monitor their health more accurately and to make more informed decisions regarding their lifestyle and healthcare needs.

Overall, the concept of wearable technology continues to be hot due to the growing technology and science, increasing interest in the use of wearable technology, the great demand for monitoring systems for assisted living and eldercare, and the participation of leading companies in the development of wearable technology. The increasing adoption of wearable technology is poised to have a significant impact on the way people monitor and manage their health, and on the broader healthcare industry as a whole. As wearable devices continue to evolve and become more integrated with

other devices and systems, they have the potential to transform the way healthcare is delivered, leading to better outcomes, greater efficiency, and more personalized care.

1.3 Challenges of Wearable Technologies

Along with the advancements, several challenges need to be addressed to ensure the continued growth and adoption of wearable technology. One of the most significant challenges is battery life, as the devices require more energy to operate, leading to shorter battery life, which can be especially problematic for devices that are worn constantly. Design and comfort are also critical issues to ensure that wearable technologies are both functional and aesthetically pleasing, particularly when it comes to integrating sensors and other components into a compact form factor.

Data privacy and security are significant concerns in the field of wearable technology, particularly in the healthcare industry, as these devices collect and transmit sensitive health data. Patients who use wearable devices to track their health data need to trust that their information is secure and private. However, wearable devices can be vulnerable to hacking or data breaches, potentially putting patients' personal health information at risk. Therefore, it is crucial to implement appropriate security measures to protect patients' data from unauthorized access or breaches.

Interoperability and integration are also major challenges in the adoption of wearable technologies in healthcare. Many different types of wearable devices are available in the market, with varying capabilities and data formats. This can make it difficult to integrate wearable devices into existing healthcare workflows and systems, limiting their usefulness and potential impact. Standardization can help address this issue by ensuring that devices are interoperable and can communicate with each other seamlessly, making it easier to integrate wearable technologies into existing healthcare workflows and systems.

Ensuring the accuracy and reliability of wearable devices is crucial to their adoption and usefulness. Inaccurate readings can lead to frustration and distrust among users, making it less likely for them to continue using the device or to trust the data

it collects. Additionally, inaccurate readings can lead to incorrect diagnoses or recommendations, potentially harming the patient's health outcomes. There are several challenges in ensuring the accuracy and reliability of wearable devices. Environmental factors, such as temperature and humidity, can affect the performance of sensors and other components, leading to inaccurate readings. Individual factors, such as differences in body types and movement patterns, can also affect the performance of wearable devices, making it challenging to develop one-size-fits-all solutions. To overcome these challenges, wearable technology manufacturers need to invest in research and development to improve the accuracy and reliability of their devices. This can include developing more advanced sensors and algorithms that can better account for individual and environmental factors. Additionally, manufacturers can work with healthcare providers and other stakeholders to validate the accuracy and reliability of their devices through clinical studies and other validation processes. Ensuring the accuracy and reliability of wearable devices is essential to their adoption and success, particularly in healthcare applications where the data collected by these devices can have a significant impact on patient outcomes. By overcoming the challenges associated with accuracy and reliability, wearable technologies can continue to play a critical role in improving patient health and well-being.

The challenge of a lack of understanding is also a significant obstacle to user adoption and engagement of wearable technology, particularly for the senior population. Wearable devices often come with complex interfaces and a range of features and functions that may be confusing or overwhelming for users who are not familiar with the technology. This lack of understanding can create a barrier to adoption and limit engagement, as users may not be aware of the full range of benefits that wearable devices can offer. Additionally, wearable technology can be perceived as intimidating or even invasive, particularly for users who are not accustomed to using digital devices. This perception can further contribute to the challenge of user adoption and engagement, as users may feel hesitant to use the devices and unsure about how to integrate them into their daily routines. Addressing this challenge will require innovative solutions that simplify interfaces and make wearable devices more intuitive and user-friendly. Additionally, education and training programs can help users better understand how to use their devices and integrate them into their daily routines, which can help to increase engagement and drive adoption.

Addressing these challenges will require ongoing innovation, investment, and collaboration from manufacturers, and policymakers. While there are challenges, the potential of wearable technology to transform sports science, entertainment, HMI systems, and healthcare is significant. As wearable devices continue to evolve and become more integrated with other devices and systems, it is likely that we will see even greater innovation and new applications of wearable technology in the future.

1.4 Research Objectives

Wearable technologies are network devices that collect data and track activities to prevent diseases and emergency health hazards by reminding the wearer or caregiver to take appropriate action [11, 12]. In short, the performance of wearable devices can be divided into the following tasks: measurement, analysis, storage, transmission, and operation. In practical clinical applications, wearable devices, in addition to ensuring the accuracy of signal measurement, must also have accurate analysis and processing of the data provided. The analysis might take place on the device itself or at a remote location such as the cloud or a smartphone. Integrating with the cloud, wearable devices have enormous potential in supplying big data, which encourages and facilitates the utilization of Machine Learning algorithms for novel outcomes. As a result of this, and in light of the significant advances in deep learning, there has been a surge of interest in the development of intelligent algorithms capable of inferring valuable information from collected physiological biosignals using Machine Learning techniques such as statistical classification and Deep Neural Networks (DNNs) [13].

Designing wearable devices is a multidisciplinary task and requires the efforts of scientists to study all aspects of the field to facilitate the consumer experience by improving these devices in various aspects, such as battery stability, useful service life, accurate sensor design, low power consumption, analysis accuracy, and more. The primary emphasis of this dissertation is on the development of modern ML algorithms based on DNNs to increase the accuracy of wearable devices in specific tasks. The most widely monitored signals in a medical setting, i.e., electrocardiogram (ECG), photoplethysmogram (PPG), and electromyogram (EMG) biosignals [9] are used in

this thesis. Specifically, the main research objectives of this thesis are aimed at utilizing sEMG biosignals for hand gesture recognition and ECG and PPG biosignals for blood pressure monitoring. In this context, the thesis targets achieving the following main research objectives:

- ***Improving the Overall Accuracy***: The first objective of this dissertation is to improve the overall accuracy of wearable devices for the mentioned tasks. Achieving this goal can lead to more precise data analysis, allowing for earlier detection and treatment of health problems. For instance, ML algorithms can help to improve the accuracy of blood pressure monitoring to identify changes in blood pressure patterns that may indicate the presence of health problems. This can lead to earlier detection and treatment of health issues, potentially saving lives and reducing healthcare costs. Similarly, accurate hand gesture recognition using sEMG bio-signals can help individuals control devices with greater precision and ease, improving their overall experience.

- ***Reducing Complexity of DNN Architectures***: Overall, computation reduction in wearable devices can have a significant impact on their battery life and real-time processing. By minimizing the computational load, wearable devices can process data in real-time, providing instant feedback to the user, which can improve the user's experience and motivate them to maintain healthy habits. In addition, by reducing the amount of computation required by a wearable device, it can reduce the amount of power consumed and extend the battery life. This reduction can be achieved in several ways, such as optimizing algorithms, reducing the complexity of machine learning models, and minimizing the number of sensors and data collection.

- ***More Representative Feature Extraction in DNNs***: Feature extraction is the process of identifying the relevant features or characteristics of the biosignal that are essential for the tasks at hand. More precisely, by identifying and extracting the most relevant features, DNNs can better capture the underlying physiological processes and discriminate between different signals. This, in turn, can lead to improved performance in wearable devices. Therefore, it is essential to continuously work on developing more efficient feature extractions that can reduce computation in wearable devices without compromising their accuracy.

1.5 Thesis Contributions

As previously stated, the primary objective of this thesis is to leverage widely used bio-signals, i.e., ECG, PPG, and sEMG, to carry out non-invasive blood pressure estimation and hand gesture recognition tasks. By leveraging the capabilities of PPG and ECG bio-signals, we aim to develop a novel and efficient technique for accurately estimating blood pressure without the need for traditional cuff-based methods. Additionally, we aim to design a hand gesture recognition system based on sEMG signals that can be used in various fields such as healthcare, sports, and entertainment, to name a few. In the following, we delve into the specific contributions of the dissertation in each of the sub-domains.

First, this dissertation aims to develop an efficient framework for continuous, cuff-less, and alignment-free Blood Pressure (BP) estimation by capitalizing on recent advancements in deep neural networks. We focus on designing deep learning architectures fed by raw ECG and PPG, unlike the common methods in BP estimation literature which use engineered and pre-defined physiological features as the first building block of their recommended approaches. By utilizing raw signals (ECG and PPG) as inputs, without hand-crafted extraction of features, we explore the real potential of deep learning in the utilization of intrinsic features (deep features) of the input signals. Moreover, the basic building block of the proposed architecture is Dilated Causal Convolutions, therefore, it can be categorized as a Temporal Convolutional Network (TCN) instead of a Recurrent Neural Network. TCN architecture provides several advantages over RNNs such as faster training with lower memory requirements and more stable training because of avoiding the problem of gradient vanishing/explosion. Furthermore, by capitalizing on the significant importance of continuous blood pressure monitoring and the fact that datasets used in recent literature are not unified and properly defined, a benchmark data set is constructed from the MIMIC-I and MIMIC-III databases from the PhysioNet database to provide a unified base for evaluation and comparison of deep learning-based blood pressure estimation algorithms.

Second, we conducted our research focusing on Hand Gesture Recognition (HGR) utilizing Surface-Electromyogram (sEMG) signals. This is due to its unique potential

for decoding wearable data to interpret human intent for immersion in Mixed Reality (MR) environments. Recent studies on sEMG-based hand gesture recognition have primarily focused on deep neural network models that employ a single path for gesture recognition. Unfortunately, these models overlook the spatio-temporal characteristics of sEMG signals, which can lead to a lack of satisfactory generalization feature extraction and poor performance. To improve the accuracy of sEMG-based hand gesture recognition, it is necessary to design models that take into account the spatio-temporal nature of the signals. This can be achieved through the incorporation of multi-path architectures resulting in more advanced feature extraction. In particular, capitalizing on the recent success of Transformers in various fields of Machine Learning [14–17], we aim to examine its applicability and potential for sEMG-based hand gesture recognition. The proposed method, referred to as the Transformer for Hand Gesture Recognition (TraHGR), increases the accuracy of sEMG decoding for the classification of hand movements. The framework being proposed utilizes two parallel paths for both special and spatio-temporal feature extraction, which is then followed by linear layers integrating the output of these two paths resulting in more representative features for different numbers of hand movements. Considering both the spatial and temporal characteristics of the signals, the proposed framework resulted in better accuracy in sEMG-based hand gesture recognition systems, which has the potential to greatly benefit fields such as human-computer interaction and rehabilitation engineering.

Finally, despite extensive research in this area and the fact that academic researchers achieve high classification accuracy in laboratory conditions, there is still a gap between academic research in sEMG pattern recognition and commercialized solutions [18]. In this context, one of the objectives for reducing the gap is to focus on the development of DNN-based models that not only have high recognition accuracy but also have minimal processing complexity, allowing them to be embedded in low-power devices such as wearable controllers [19, 20]. The existing solutions so far rely on complicated and heavy-weighted Deep Neural Networks (DNNs), which have restricted practical application in low-power and resource-constrained wearable systems. Therefore, in another attempt, we propose a light-weighted hybrid architecture (named HDCAM) based on Convolutional Neural Network (CNN) and attention mechanism to effectively extract local and global representations of the input.

Furthermore, the model is trained based on a hybrid loss function consisting of two-fold: (i) Cross Entropy (CE) loss which focuses on identifying the helpful features to perform the classification objective, and (ii) Supervised Contrastive (SC) loss which assists to learn more robust and generic features by minimizing the ratio of intra-class to inter-class similarity.

1.6 Organization of the Thesis

In **Chapter 1** (this chapter), we provided an overview and a summary of important contributions made in the thesis. The rest of the thesis is organized as follows:

- **Chapter 2** provides a literature review on Blood Pressure estimation, as well as the sEMG-based hand gesture recognition approaches.
- In **Chapter 3**, the details and experiments of the proposed deep learning solutions for continuous, cuff-less, and alignment-free BP estimation are provided. Furthermore, a benchmark dataset is introduced, with the potential to serve as a unified base for the evaluation and comparison of deep learning-based BP estimation algorithms.
- In **Chapter 4**, we concentrate on our proposed deep learning-based solution based on Transformer architecture to improve the HGR accuracy based on sEMG signals.
- In **Chapter 5**, the details of the HDCAM architecture is provided which gains the advantages of both Transformers- and CNN-based models to create a lightweight and low-latency network for HGR tasks using sEMG signals.
- In **Chapter 6**, we conclude the dissertation and the potential future works are discussed.

1.7 Publications

Journal Publications

J3. **S. Zabihi**, E. Rahimian, A. Asif, S. Yanushkevich, and A. Mohammadi, "Light-weight CNN-Attention based Architecture Trained with a Hybrid Objective Function for EMG-based Human Machine Interfaces", *Accepted in Transactions on Computational Science*, 2023.

J2. **S. Zabihi**,E. Rahimian, F. Marefat, A. Asif, P. Mohseni, and A. Mohammadi, "BP-Net: Cuff-less and Non-invasive Blood Pressure Estimation Via a Generic Deep Convolutional Architecture", *Biomedical Signal Processing and Control*, vol. 78, p.103850, 2022.

J1. **S. Zabihi**, E. Rahimian, A. Asif, and A. Mohammadi, "Tra-HGR: Transformer for Hand Gesture Recognition via ElectroMyography", *Submitted to IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE)*, 2023.

Conference Publications

C1. **S. Zabihi**,E. Rahimian, A. Asif, and A. Mohammadi, "Light-weighted CNN-Attention based architecture for Hand Gesture Recognition via ElectroMyography", *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023.

Chapter 2

Literature Review

Wearable technologies are gaining traction as a possible post-smartphone platform. Wearable devices have various benefits over smartphones, including being smaller, lighter, and, most crucially, being able to be worn by the consumer. As a result, these technologies are increasingly being used in areas such as healthcare, robotic prosthesis control, VR/AR, and mining personal data (i.e., life-logging, quantified-self) [21, 22]. As stated previously in Chapter 1, the objective of this project is to leverage bio-signals including ECG, PPG, and sEMG, which are among the most commonly used bio-signals in wearable devices, for cuff-less Blood Pressure (BP) estimation and hand gesture recognition tasks. As a result, the emphasis in this chapter is on providing an overview of cuff-less BP estimation methods. Furthermore, we present an overview of advanced deep neural network (DNN)-based methodologies for sEMG-based hand gesture detection algorithms.

2.1 Cuff-less Blood Pressure estimation

An alarming population aging is widely expected in the near future partially due to recent advancements in biomedical health technologies. According to a recent publication by the United Nations [23], the number of seniors over the age of 60 is expected to double by 2050, even it is projected that the population of seniors will

be more than the population of minors/youth at ages 10-24 by 2050. Consequent to this inevitable worldwide population aging trend is a significant increase in age-related health issues, in particular cardiovascular conditions. According to the World Health Organization report and on a global scale, cardiovascular diseases account for approximately 17 million loss of lives annually, which accounts for one-third of the total deaths around the world. Of these, complications of hypertension account for 9.4 million loss of lives annually. These facts call for an urgent quest to develop advanced continuous monitoring, efficient diagnosis, and timely treatment of cardiovascular conditions.

Generally speaking, BP can be described as the pressure applied by blood to the arteries (wall of the blood vessels) ranging between two limits, i.e., a maximum value named Systolic Blood Pressure (SBP) to a minimum value referred to as the Diastolic Blood Pressure (DBP). When an individual's SBP exceeds 140 mmHg and their diastolic blood pressure DBP rises above 90 mmHg, they are diagnosed with hypertension. This condition is characterized by abnormally high blood pressure levels, which can lead to a variety of health problems such as heart disease, vision loss, stroke, and kidney failure, to name a few. Therefore, it is crucial to monitor blood pressure regularly and seek medical attention if levels are consistently above the normal range. Early detection and diagnosis of hypertension can help patients receive accurate treatment and manage the condition, leading to a reduction in the overall mortality rate for those affected by hypertension. While regular BP checkups are recommended by physicians for seniors, this typically can not be achieved due to complications of human activities and the fast pace of modern lifestyle, furthermore, historical data shows that on average 20% of seniors have higher measured BP at clinics in comparison to the relaxed home environment. Consequently, continuous and in-home monitoring of BP [24]- [27] via utilization of advanced Biological Signal Processing (BSP), Artificial Intelligence (AI) and Machine Learning (ML) techniques [28]- [33] becomes of paramount importance, especially with the growing popularity of wearable devices. In Section 2.1.1, we provide an overview of recent literature in this domain.

2.1.1 Review of the Literatures on Blood Pressure Estimation

In general, when it comes to measuring BP, commonly, either a cuff-based approach is utilized, which provides upper arm BP measurements, or one resorts to cuff-less (possibly invasive) solutions. Upper arm BP monitoring can provide users with an indirect and non-continuous BP measurement technique by using an inflatable cuff and stethoscope. Such cuff-based methods suffer from several drawbacks including: (i) Being inconvenient and unhealthy, especially in public places; (ii) Requiring proper training prior to utilization; (iii) Not being ideal for self-use and long-term monitoring of BP, and; (iv) Being incapable of providing continuous BP measurements [34]. Cuff-less BP monitoring [35], on the other hand, eliminates the common uncomfortable factors associated with the former category and has the potential to continuously provide BP estimates without using any inflatable cuff.

Recently, there has been a surge of interest towards the goal of performing continuous BP monitoring via *Physiologically Inspired Models* [36]- [39] in particular pulse transit time (PTT) and pulse arrival time (PAT). The PAT, sum of PTT and pre-ejection period, is considered as the main marker of BP for development of cuff-less BP estimation algorithms due to its simple measurement procedure. The PAT [40,41] is defined as the time required for a heartbeat to transfer to a body peripheral and has a tight relationship (correlation) to the BP. The existing correlation between PAT and BP, although well established, is highly non-linear depending on several uncertain factors varying across different individuals and over time [42,43]. PTT involves simultaneous measurement of ECG and PPG along with other variables such as the patient's size, weight, and age. Consequently, possible temporal misalignment of the PPG and ECG signals results in incorrect values for these temporal features, therefore, incorrect results of the downstream investigations, i.e., BP estimation. Therefore, there have been different attempts [38, 39, 44] to construct alignment/calibration methods between multimode signals to account for such variations. We categorize these approaches as "*Alignment-based*" models, where the focus is on extracting meaningful features to be fed to processing and learning models. Such models, however, are applicable only for use in short intervals such as exercise tests.

Existing methods for cuff-less and continuous BP estimation can be classified into

the following two main categories:

(i) Hand-crafted Regression-based Models: Models belonging to this category are developed by extracting hand-crafted features and exploiting various conventional BSP and ML algorithms such as decision trees (DT), support vector regression (SVR), shallow neural networks, and Bayesian linear regression (BLR) to name but a few. Typically, PPG and ECG signals are used jointly [34, 45] to extract PAT features to construct a regression model for estimating the BP. In Reference [46], for instance, a linear regression model (i.e., the SVR) is coupled with a radial basis function (RBF) kernel, and a single hidden layer neural network with a linear output to estimate the BP. A key drawback of the aforementioned conventional methods is that such models rely heavily on the extraction of hand-crafted features, such as PAT, and directly map the given input into the target value while ignoring the critical temporal dependencies in the BP dynamics. This could be considered as the root of the long-term inaccuracy of such models, which, in turn, results in a lack of robustness due to strong dependencies on the alignment parameters and the choice of hand-crafted features to describe the signal for subsequent regression [47]. Lack of robustness due to frequent alignment requirements in such models translates into accuracy decay over time.

(ii) Deep Learning-based Models: While research works on hand-crafted and regression-based models are extensive, deep-learning-based BP estimation [48, 49] is still in its infancy. In deep-learning models, commonly, hand-crafted features (e.g., extracted PAT features) are fed to neural network models such as long short-term memory (LSTM) models, recurrent neural networks (RNN), convolutional neural networks (CNN), or bidirectional RNN (BRNN). For instance, Reference [48] proposed to formulate BP prediction as a sequence learning problem, and proposed a deep RNN model, which is targeted for multi-day continuous BP prediction. The RNN model works with a set of seven representative hand-crafted features extracted from ECG and PPG signals. Another recent example is Reference [50], where the authors proposed a waveform-based Artificial neural network (ANN)-LSTM model. The model consists of a hierarchical structure where the lower hierarchy level uses ANNs to extract the required features from the ECG/PPG waveforms, while the upper hierarchy level uses stacked LSTM layers to learn the time-domain variations of the features

extracted in the lower hierarchy level.

In most of the studies presented so far, before extracting the deep features, representative hand-crafted features of input signals are first selected/extracted, which are then used to train a deep neural network. In other words, such methods ignore the real potential of deep learning in utilizing the intrinsic features (deep features) of the input signals. In this context, recently, different studies [51]- [56] have proposed end-to-end deep networks for BP estimation from PPG and ECG signals. Such existing end-to-end models are, typically, developed based on small datasets and/or used a single ECG lead, which only allows development of simple architectures. In other words, due to the availability of small training sets, complex end-to-end networks can not be developed. Furthermore, datasets used in recent literature are not unified and properly defined, which makes evaluations and comparisons difficult. Commonly a subset of MIMIC-I or MIMIC-III databases from PhysioNet is used without providing details on the training, validation, and test sets rendering reproducibility and fair comparisons impossible. For instance, in Reference [51], the authors used different deep learning techniques including Fully Connected Neural Network, LSTM, Wavenet, Wavenet + LSTM, and Resnet + LSTM for BP estimation. The experiments were, however, carried out based on only 40 patients from the MIMIC database, which is not enough for constructing an end-to-end model that can be used more generally. Similarly, in Reference [52], the authors estimated continuous blood pressure by using an adaptive weight learning-based multitask deep learning framework. The blood pressure estimation, however, is only performed based on a single lead (lead II) of EGG signals [52]. Generally speaking, using single-lead ECG signals [52] makes it difficult to evaluate the generality of the obtained results. Along a similar path, in Reference [53], the authors developed a convolution-based deep autoencoder (DAE) model for predicting Arterial Blood Pressure (ABP) from the raw PPG signals. A total of $1,227$ records are derived from MIMIC-II database [54], however, the number of patients is not mentioned. The records are divided into three categories based on the range of the BP values, after which, 60% of each category is considered for the training set, 20% for the validation, and the remaining is used as the test set. Although all samples of each record only appear in the train, validation, or the test set, there is no evidence that the records are from different patients, noting that in the MIMIC II database [54], for each patient multiple records are available. After

training the model, 80 seconds (i.e., $10,000$ samples) from the test set is used for penalization, resulting in a Mean Absolute Error of 7.945 and 4.114 for Systolic BP and Diastolic BP, respectively.

In Reference [55], the authors developed and validated a continuous non-invasive BP prediction using PPG signals. More specifically, in Reference [55], pulse wave morphology is analyzed, and then an ML method is used to describe the relationship between BP-indicators and BP. A probabilistic generative model is constructed based on Deep Belief Network Restricted Boltzmann Machine feed-forward neural network [55] to estimate Systolic BP and Diastolic BP. Proprietary data is used to train and validate their proposed model resulting in the Mean Error (ME) of -2.98 and -3.65 for Systolic BP and Diastolic BP, respectively. In Reference [56], an end-to-end deep learning architecture is proposed using only raw signals without the process of extracting features to improve BP estimation. The proposed model consisted of a convolutional neural network, a bidirectional gated recurrent unit, and an attention mechanism. A total of 15 subjects were recruited for the study and 70 percent of the data for each subject was used for training, 10% for validation, and the remaining 20% for testing. In this study [56], the data of each subject is divided into windows of 5 seconds and only one forecast is made for SBP and DBP for the whole segment. In other words, the target labels for SBP and DBP correspond to the end of each segment [56], which is the common approach in BP estimation task.

2.2 sEMG-based Hand Gesture Recognition

Generally speaking, Surface Electromyogram (sEMG) datasets can be collected based on "sparse multichannel sEMG" or "High-Density sEMG (HD-sEMG)". The latter records the electrical activity of muscles by two-dimensional arrays of closely-spaced electrodes, extracting both temporal and spatial changes of muscle action potentials. The advantages of this technique include the ability to obtain a large amount of data and more robustness to electrode changes [57–59]. Despite advantages of HD-sEMG, its utilization leads to structural complexity [60, 61], while adoption of sparse multichannel sEMG signals requires fewer electrodes making it the common modality

of choice for incorporation into wearable devices [57, 62]. Therefore, development of DNN models based on sparse sEMG signals has gained significant recent importance. However, more efforts are needed to bridge the gap between academic research and clinical solutions in this area [18].

sEMG-based Hand Gesture Recognition (HGR) is regarded as a promising approach for a wide range of applications, including myoelectric control prosthesis [19, 63–65], virtual reality technologies [66,67], Human-Computer Interactions (HCI) [68], and rehabilitative gaming systems [69]. sEMG signals contain electrical activities of the muscle fibers that can be employed to decode hand gestures and thereby enhance immersive HMI wearable systems for immersion in Mixed Reality (MR) environments [18, 70]. In healthcare, sEMG is used to diagnose and treat neuromuscular disorders such as muscular dystrophy, cerebral palsy, and spinal cord injuries. It can also be used to control prosthetic limbs, allowing amputees to perform daily tasks and regain independence. sEMG can provide valuable information about muscle activation patterns, allowing clinicians to develop targeted treatment plans and monitor progress over time. In sports science, sEMG is used to measure muscle activation during physical activity, providing insights into muscle function and performance. It is also used in rehabilitation and injury prevention programs to help athletes recover from injuries and maintain optimal physical health. In robotics, sEMG is used for the development of robotic exoskeletons and other assistive technologies, allowing users to control devices using their muscle activity. This technology has promising applications in rehabilitation and mobility assistance. In HMI/HCI, sEMG is used for gesture recognition and control of computer interfaces, allowing users to control devices and applications using natural hand movements. This technology has the potential to improve accessibility and ease of use for individuals with disabilities. Overall, sEMG technology has a wide range of applications, and ongoing advancements in machine learning and signal processing techniques are making it even more powerful and versatile. Consequently, there has been a surge of interest in the development of Deep Neural Networks (DNNs) and Machine Learning (ML) models to identify hand gestures using sEMG signals. In Section 2.2.1, we provide an overview of recent literature in this domain.

2.2.1 Review of the Literatures on Myoelectric Control System

Typically, sparse sEMG signals are used in systems with only 2-8 channels to obtain descriptive spectrotemporal features to determine the intended motion for the control of the prosthesis. Early systems used electrodes carefully placed over agonist/antagonist muscle groups to control a single Degree of Freedom (DoF) on a power prosthesis. Issues with noise and spatial resolution prevent this technique from extending well into multi-DoF systems [71]. To enhance the performance, multiple sensors can be used to detect the intended DoF and the intensity of the activation. This modification is named as proportional control which has been used for multi-degree of freedom control of prostheses [18]. This type of direct/proportional control is incredibly robust and remains the basis for most if not all control paradigms in clinical use today. However, studies have shown that sEMG amplitude is non-linearly related to muscle output force, especially when there is variation in joint angle [72]. Thus, amplitude-based direct control systems tend to result in disproportionate and non-intuitive limb motions, which is believed to contribute to the high incidence of upper limb-powered prosthesis abandonment [72]. Pattern Recognition control techniques have emerged as potential solutions to utilize sEMG signals for high precision, multi-DoF control, although further work is necessary for these to become practical for clinical use.

More recently, sEMG signals have been collected with arrays of closely-located high density (HD) electrodes for use in classification systems and Motor Unit Action Potentials (MUAPs) Decomposition. When used with an ML-based pattern recognition algorithm, data from HD-sEMG arrays can accurately differentiate between a greater number of classes than a traditional 8 electrode setup [73]. HD-sEMG classification systems have also been shown to have increased robustness to electrode shift, temporal variation, and electrode failure [71, 73]. This approach can offer greater spatial resolution and more comprehensive information about muscle activity, making it ideal for research and clinical applications that require detailed muscle analysis. However, HD-sEMG systems can be expensive, bulky, and require significant processing power to manage the large amounts of data generated. Moreover, some studies with

HD-sEMG systems have found that the relationship between accuracy and the number of electrodes is not necessarily a monotonic function, and increasing the number of electrodes beyond an optimal point can cause the system to lose accuracy due to increased noise and over-fitting. Thus one use of HD-sEMG arrays is to select optimal electrode placement [71]. Other studies record data with the full array but only utilize a subset of the electrode channels for the control algorithm.

Although high-density sEMG (HD-sEMG) has been shown to offer higher spatial resolution and more detailed information about muscle activity than sparse sEMG data collection, the use of sparse sEMG data collection has its own set of advantages. One of the primary benefits of sparse sEMG is that it can significantly reduce the complexity of wearable devices, making them more accessible, smaller, and practical for a wider range of applications. Additionally, sparse sEMG data collection can reduce the amount of processing power and storage required to analyze the data, making it easier to implement real-time signal processing and control algorithms. Finally, sparse sEMG can be more comfortable for users as it requires fewer electrodes, and they can be placed in more convenient locations, reducing the risk of skin irritation or discomfort. Overall, both approaches have their own set of advantages and limitations, and the choice of which method to use depends on the specific requirements of the application. and the decision on which technique to use ultimately relies on the specific needs of the application

The existing researches on prosthetic myoelectric control focus primarily on traditional ML approaches as a common strategy for HGR [74]. In general, the developed methods for classifying hand movements can be classified into the following two main categories: (i) Traditional approaches based on Machine Learning (ML) architectures [75–79]; and (ii) DNN-based techniques [57, 58, 64, 79, 80, 83, 84, 92, 96]. The common approach to perform Hand Gesture Recognition (HGR) in traditional methods (Case (i)) is to extract hand-crafted (engineered) features to train classical ML models such as Linear Discriminant Analysis (LDA), Support Vector Machine (SVM), and Random Forests (RF). More specifically, in such methods, handcrafted features, in the time domain, frequency domain, or time-frequency domain [58], are first extracted by human experts, which are then fed to a classifier. Extraction and

feature selection, however, can affect the overall performance [85], as such some researchers [57] have explored and integrated several classical feature sets that provide multi-view of sEMG signals to achieve higher gesture recognition accuracy. On the other hand, different classifiers such as SVM, LDA, RF, and Principal Components Analysis (PCA) are utilized in the literatures [78, 79, 85–87] to increase the discriminating power of the model and improve gesture recognition performance.

Although the traditional ML-based approaches have shown strong potential for HGR task, more recently, there has been a great deal of interest in using deep-learning architectures (Case (ii)) to process multi-channel sEMG signals and increase the discrimination power of the model. In particular, it has been shown [79] that the automatic feature extraction used in deep learning architecture can lead to higher classification accuracy compared to their classical counterparts. This achievement was the starting point for considering CNN as a promising approach in the context of sEMG data classification [64, 84, 88]. In [59], authors proposed a CNN architecture to extract spatial information from sEMG signals and perform HGR classification. CNN-based architectures [57, 59, 88, 89] are common approaches for hand movement classification, where sEMG signals are first converted into images and then used as input for CNN-based architectures. However, the nature of sEMG signals is sequential, and CNN architectures only take into account the spatial features of the sEMG signals. Therefore, in recent literature [20, 61, 90, 91], authors proposed using recurrent-based architectures such as Long Short Term Memory (LSTM) networks to exploit the temporal features of sEMG signals. On the other hand, it is suggested in [58,92–94] to use hybrid models (CNN-LSTM architecture) instead of using a single model to capture the temporal and spatial characteristics of sEMG signals. Although recent academic researchers are improving the performance by using RNNs or hybrid architectures, the sequence modeling with recurrent-based architectures has several drawbacks such as consuming high memory, lack of parallelism, and lack of stable gradient during the training [65, 96]. It is demonstrated in [95] that sequence modeling using RNN-based models does not always outperform CNN-based designs. Specifically, CNN architectures have several advantages over RNNs such as lower memory requirements and faster training if designed properly [95]. Therefore, in the recent literature [65, 96–98], the authors took advantage of 1-D Convolutions developed based

on the dilated causal convolutions, where the sequence of sEMG signals can be processed as a whole with lower memory requirement during the training compared to RNNs. Convolution operation in CNNs, however, has two main limitations, i.e., (i) it has a local receptive field, which makes it incapable of modeling global context, and; (ii) their learned weights remain stationary at inference time, therefore, they cannot adapt to changes in input. Attention mechanism [14] can mitigate both of these problems. Consequently, the authors in the recent research papers [19, 63, 99–101] used the attention mechanism combined with CNNs and/or RNNs to improve the performance of sEMG-based HGR. The attention mechanism's major disadvantage is that it is often computationally intensive. Therefore, a carefully engineered design is required to make attention-based models computationally viable, especially for low-power devices.

Worth noting that one inherent problem in the sEMG-based hand gesture recognition task is the time- and user-dependent nature of the sEMG signal [102]. In other words, due to physiological differences in muscle activities from one user to another, a pre-trained model on existing users (source) cannot be expected to perform well on a new user (target) [103, 104]. In addition, various electrophysiological and user-related variables can affect sEMG signals. These include muscle fatigue [105], changes in electrode-skin impedance due to perspiration/humidity [106], electrode displacement [107], and user-related issues such as variations in contraction intensity, hand orientations, and arm postures [108]. As a result of these changes, the accuracy of a pre-trained model on source data may degrade when testing on the target user due to the domain shift. To this end, domain adaptation methods are highly recommended in this field of study, where learning algorithms involve some techniques to transfer information from the source to the target domain despite the existence of a distribution mismatch among them. As a result, Transfer Learning (TL) techniques in upper-limb hand motion estimates have received a lot of attention in recent years [109–114]. Furthermore, in [99], as a domain adaptation methodology, a novel Few-Shot learning method is introduced with the objective of inferring the desired output based on just one or a few observations from the target domain, resulting in a promising performance for *unseen user* hand gesture recognition.

Table 2.1: Maximum path lengths, per-layer complexity, and the minimum number of sequential operations for different layer types. n is the sequence length, d is the representation dimension, k is the kernel size of convolutions and r the size of the neighborhood in restricted self-attention.

Layer Type	Complexity per Layer	Sequential Operations	Maximum Path Length
Self-Attention	$O(n^2 \cdot d)$	$O(1)$	$O(1)$
Recurrent	$O(n \cdot d^2)$	$O(n)$	$O(n)$
Convolutional	$O(k \cdot n \cdot d^2)$	$O(1)$	$O(log_k(n))$
Self-Attention (restricted)	$O(r \cdot n \cdot d)$	$O(1)$	$O(n/r)$

2.3 Brief Review on the Complexity of Common Deep Neural Network Layers

Deep Neural Networks (DNNs) consist of various layers that perform specific operations on the input data. Each layer has its own computational complexity, which influences the overall efficiency and performance of the network. Understanding the computational complexity of different layers in deep neural networks is crucial for designing efficient models. In this section, we review the complexities of common layers, including self-attention layers, recurrent layers, and convolutional layers. Table 2.1 provides an overview to summarize the complexities of these layers, the table is borrowed from Ref. [14].

CNNs are widely used for capturing spatial patterns in data. The complexity of a single convolutional layer depends on the kernel width (k) and the sequence length (n). When $k < n$, a single layer fails to connect all input and output positions. To achieve complete connectivity, a stack of $O(n/k)$ convolutional layers is required for contiguous kernels. In the case of dilated convolutions, $O(log_k(n))$ layers are needed. However, this increases the length of the longest paths between positions in the network. Although convolutional layers are generally more computationally expensive than recurrent layers, techniques like separable convolutions can significantly decrease the complexity to $O(k \cdot n \cdot d + n \cdot d^2)$. However, this will reduce the representation power on convolutional layers.

RNNs are particularly suitable for handling sequential data. The computational complexity of recurrent layers is $O(n)$, as they require sequential operations. However, RNNs have certain drawbacks, including high memory consumption, lack of parallelism, and unstable gradients during training. These limitations can impact their performance, especially in tasks with longer sequences.

Transformers have gained prominence due to their exceptional performance in various applications. The complexity of Transformers primarily lies in their self-attention layers. A self-attention layer connects all positions in a sequence with a constant number of sequentially executed operations ($O(1)$). This makes self-attention layers faster than recurrent layers when the sequence length (n) is smaller than the representation dimensionality (d). However, for tasks involving very long sequences, the computational performance of self-attention can be improved by restricting attention to a neighborhood of size r around each output position. This modification increases the maximum path length to $O(n/r)$.

Chapter 3

Cuff-less and Non-invasive Blood Pressure Estimation

This chapter focuses on development of deep learning-based architecture for continuous and cuff-less blood pressure (BP) monitoring. In this regard, a robust deep learning-based framework is proposed for computation of low latency and continuous upper and lower bounds on the systolic and diastolic BP. Referred to as the BP-Net, the proposed framework, shown in Fig. 3.1, is a novel convolutional architecture that provides longer effective memory while achieving superior performance due to the incorporation of casual dilated convolutions and residual connections. It is worth mentioning that in the realm of blood pressure estimation, the prevailing approach has been the utilization of RNN-based models. However, our research endeavors to introduce a novel paradigm by adopting a different architectural foundation - Temporal Casual Convolution. This choice stems from the notable advantages offered by this architecture, which justifies the departure from the commonly employed RNN-based models. By leveraging the strengths of Temporal Causal Convolutions, our model brings forth faster and fewer memory requirements during training times. Another crucial advantage of Temporal Causal Convolutions is their ability to address the problem of gradient vanishing or explosion, which can hinder training and limit the performance of RNN-based models. By avoiding this issue, our model benefits from enhanced training stability, ensuring more consistent and reliable convergence

during the learning process. By harnessing the strengths of Temporal Causal Convolutions, our model offers a promising approach to blood pressure estimation. We anticipate that these advantages will translate into improved accuracy and robustness compared to the prevalent RNN-based models. Through our research, we aim to push the boundaries of performance and contribute to advancing the field of blood pressure estimation through the adoption of Temporal Causal Convolutions as a viable alternative to RNN architectures. To utilize the real potential of deep learning in extraction of intrinsic features (deep features) and enhance the long-term robustness, the BP-Net uses raw Electrocardiograph (ECG) and Photoplethysmograph (PPG) signals without extraction of any form of hand-crafted features as it is common in existing solutions. By capitalizing on the fact that datasets used in recent literature are not unified and properly defined, a benchmark dataset is constructed from the MIMIC-I and MIMIC-III databases [115, 116] obtained from PhysioNet. The proposed BP-Net is evaluated based on this benchmark dataset demonstrating promising performance and showing superior generalizable capacity. The proposed BP-Net architecture is more accurate than canonical recurrent networks and enhances the long-term robustness of the BP estimation task. The proposed BP-Net architecture addresses key drawbacks of existing BP estimation solutions, i.e., relying heavily on extraction of hand-crafted features, such as pulse arrival time (PAT). Finally, the constructed BP-Net dataset provides a unified base for evaluation and comparison of deep learning-based BP estimation algorithms.

The BP-Net architecture proposes a novel convolutional architecture for estimating BP. After denoising the photo-plethysmograph (PPG) and electrocardiogram (ECG) signals, the pre-processed signals are provided as inputs to the designed convolutional architecture, i.e., excluding the need for feeding the models with hand-crafted features, therefore, intrinsic deep features of the PPG and ECG signals are being used. In brief, our main contributions for BP estimation can be summarized as follows:

- Most of the existing data-driven methodologies proposed for cuff-less BP estimation depend on extraction of specific hand-crafted features such as pulse arrival time (PAT) [36–44]. Capitalizing on recent evidence that neural networks can extract the necessary features automatically without the need for complex feature engineering, we propose a convolutional architecture model that extracts

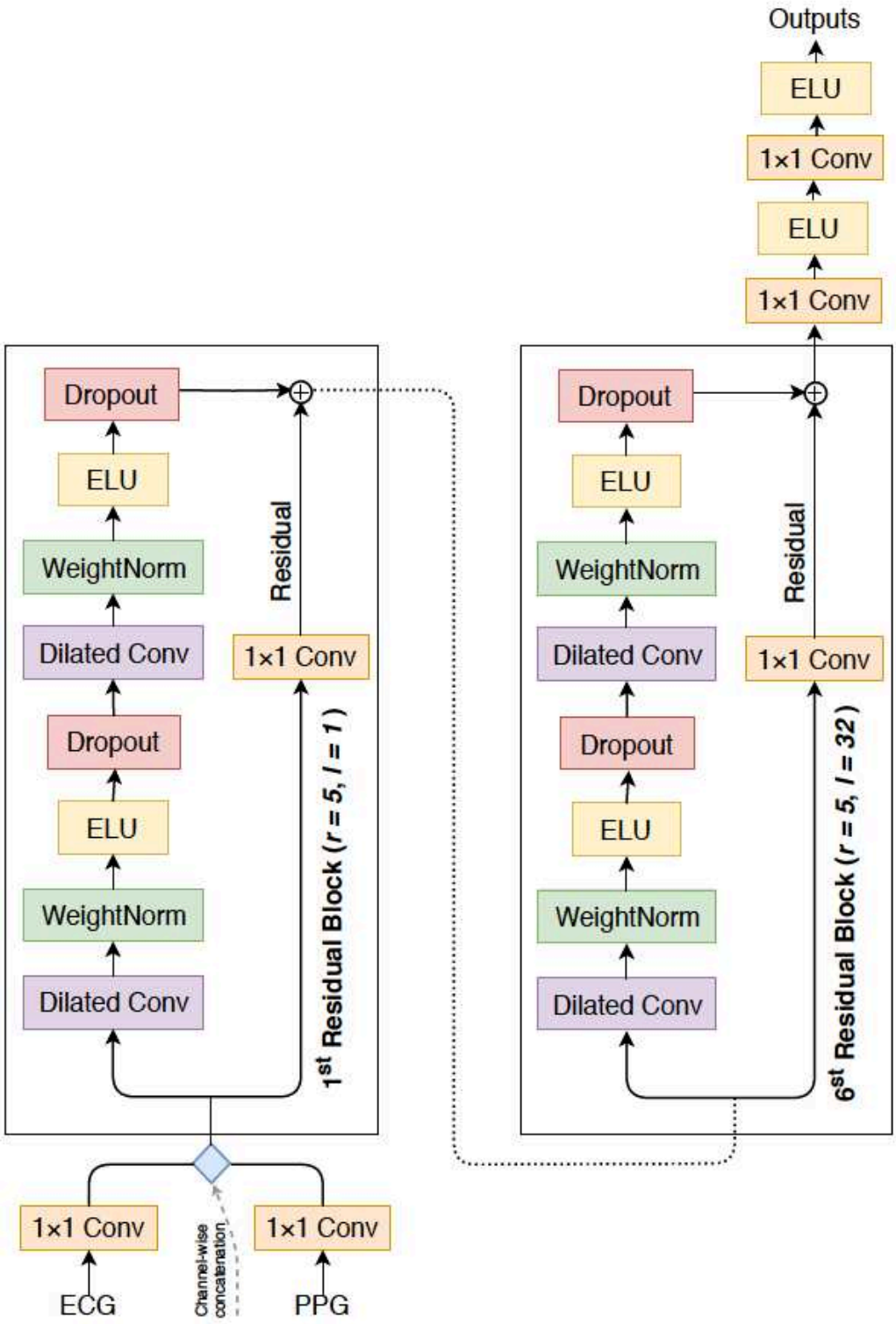

Figure 3.1: The architecture of proposed BP-Net.

necessary features automatically using raw ECG and PPG waveforms. The proposed model is able to estimate BP with high accuracy in an end-to-end manner.

- The proposed BP-Net provides longer effective memory while achieving superior performance in comparison to recurrent neural networks due to the incorporation of casual dilated convolutions and residual connections.

- In the studies presented so far, the lead of ECG signal is not considered (i.e., the BP is estimated based on one type of ECG lead). The availability of different leads of ECG signals (which are not the same for every subject) makes it difficult to evaluate the generality of the obtained results. To address this problem, and also to show the generality of the proposed model, we use different leads of ECG signals such as I, II, III, V, AVR, and MCL.

- By capitalizing on the significant importance of continuous BP monitoring and the fact that datasets used in recent literature are not unified and properly defined, a benchmark data set is constructed from the MIMIC-I and MIMIC-III databases from PhysioNet to provide a unified base for evaluation and comparison of deep learning-based BP estimation algorithms.

The rest of the Chapter is organized as follows: The proposed BP-Net architecture is presented in Section 3.1. Section 3.2 presents the experimental results for the evaluation of the proposed framework based on real constructed datasets. Finally, Section 3.3 concludes this chapter.

3.1 The BP-Net Framework

In the proposed method, estimation of the SBP and the DBP is performed automatically via extraction of deep features from raw ECG and PPG signals without incorporation of any form of hand-crafted PAT features. To achieve this goal, we approach the BP estimation problem as a sequence modeling task. Before describing the architecture of the proposed BP-Net, in what follows, first we provide a brief overview of the constructed dataset utilized to develop the proposed BP-Net.

3.1.1 BP-Net Dataset

As stated previously, in this Chapter, we introduce a unified and properly defined benchmark dataset given the significant importance of continuous BP monitoring and the fact that recent research works to train and test their algorithms on datasets of their choice, impeding a fair judgment between their solutions. Capitalizing on this issue, we aim to provide a platform with a reference dataset, where different algorithms can be evaluated and compared by utilizing the same training set to optimize new processing algorithms, and the same test dataset to be used to measure the associated performance. Despite the differences, all existing studies share a common validation procedure in which experiments are conducted on a proprietary database containing a fewer number of subjects with reference to our work, therefore, it is difficult to evaluate the generality of the obtained result. For this purpose, we increased the number of subjects (293 individuals).

The BP-Net dataset is collected from the Multi-parameter Intelligent Monitoring for Intensive Care (MIMIC) [115] provided by the PhysioNet server. MIMIC-I database contains PPG, multi-lead ECGs, and arterial blood pressure (ABP) signals at 125 samples per second with 8 bit precision. Data is derived from 90 patients monitored in the medical, surgical, and cardiac intensive care units (ICU) of different hospitals [115]. The requirement set forward to construct the BP-Net dataset is to have concurrent PPG, ECG, and ABP signals, therefore, out of the 90 available subjects, we were able to collect data from 56 patients. To further increase the number of subjects, MIMIC-III [116] database, which is an update to the common MIMIC-II [54], is also used as the second source for data preparation. The MIMIC-III contains data associated with a large number of different hospitals for distinct patients in ICU between 2001 and 2012. Similar to MIMIC-I, signals were sampled at the frequency rate of 125 Hz with 8 bit accuracy [115]. We have collected data from 237 patients who had simultaneous PPG, ECG, and ABP from this dataset resulting in a total of 293 subjects in the BP-Net dataset. In terms of duration, the size of the data is 140.43 hours, where we have a total of 653 records for 293 subjects. The ground truth SBP and DBP values are extracted from an identical ABP signal. Then, these beat-to-beat SBP and DBP data points are interpolated using Piecewise Cubic Hermite

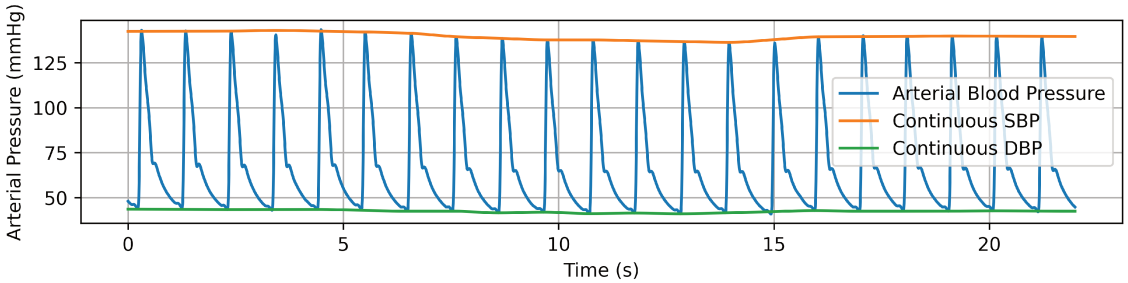

Figure 3.2: Interpolation of the intervals between max- and min-points in of the ABP signal to form continuous SBP and DBP signals.

Interpolating Polynomial (PCHIP) [117] in order to have equal input and output sampling time, result is shown in Fig. 3.2. PCHIP was chosen because it can interpolate while maintaining data structure (i.e., shape-preserving), avoiding unnecessary oscillations at each interbeat-interval, and maintaining monotonicity in the interpolation data [118, 119]. As a result of these attributes, as well as the fact that two consecutive beat-to-beat SBP (DBP) does not change significantly [120], the utilized PCHIP interpolation behaves very similarly to linear interpolation (weighted "average"). Despite the fact that the number of interpolated samples at each interbeat-interval may vary depending on the number of missing values between two consecutive beat minima (for DBP) or beat maxima (for SBP), the final continuous SBP and DBP have the same frequency as the PPG and ECG signals, i.e., 125 Hz. It is worth mentioning that in the pre-processing pipeline of the proposed BP-Net, ECG/PPG signals are first upsampled from 125 Hz to 1 kHz for filtering purposes. Once the filtering process is completed, the filtered ECG/PPG signals are then down-sampled to 125 Hz at the final step of pre-processing.

Capitalizing on that BP does not change substantially under common conditions [120], in previous studies [51, 52, 120–125], some form of approximation for SBP and DBP were utilized. In such prior works, it is common to crop ECG/PPG signals into segments of fixed length (e.g., 2, 5, 10, 15, or 25 seconds) according to the requirement of the underlying learning model. The corresponding BP signals are then also cropped into segments with the same fixed length as that of the ECG/PPG segments for extraction of ground truth BP values. Finally, for each EEG/PPG segment, SBP (DBP) values are approximated by computing the maximum (minimum) value of a segmented BP signal or taking the average of detected local maximum (local

Table 3.1: Information about the type of available ECG leads, their duration, and the number of available records.

ECG Lead	I	II	III	V	AVR	MCL	Not-Available
Number of Records	14	183	24	139	149	9	138
Duration (Hour)	3.11	40.57	5.33	30.77	32.32	1.93	26.39

minimum) values of a segmented BP signal. In other words, in these prior works, an approximation approach is used to form a ground truth label for each segment resulting in a constant SBP (DBP) representing the whole segment. In our approach, instead of using a constant SBP (DBP) for each segment, we used interpolation to have a continuous signal. A potential drawback with prior approaches is that the BP signals are, typically, contaminated by artifact noise, which can lead to incorrect ground truth SBP (DBP) extraction. This, in turn, would reduce the BP estimation accuracy of the learning model. Potential effects of artifact noise are reduced by rejecting outliers through the incorporated interpolation approach.

As stated previously, in the existing studies, to the best of our knowledge, the lead of ECG signal is not considered, i.e., the BP is estimated based on one lead of ECG. The availability of different leads of ECG signals makes it difficult to evaluate the generality of the obtained results. To address this problem, and also to show the generality of the proposed model, different leads of ECG signals are included in the dataset. Table 3.1 presents details about the type of available ECG leads, their duration, and the number of records associated with each lead. Finally, the constructed BP-Net dataset is available through the link provided in Reference [126].

3.1.2 The BP-Net Architecture

We consider the problem of BP estimation from ECG and PPG signals collected from $N_s = 293$ number of subjects, where ECG, PPG, and ABP data associated with subject l, for $(1 \leq l \leq N_s)$, each has a total of $T^{(l)}$ number of samples. We define an ECG vector $\boldsymbol{x}^{(l)}(t) = [X_1^{(l)}, \ldots, X_t^{(l)}]^T$ consisting of samples from ECG time-series collected from the l^{th} subject from the starting time $(t = 1)$ to time $(t \leq$

$T^{(l)}$). Note that $\boldsymbol{x}^{(l)}(t)$ is a vector representing all ECG samples available for the l^{th} subject. Similarly, we define a PPG vector $\boldsymbol{p}^{(l)}(t) = [P_1^{(l)}, \ldots, P_t^{(l)}]^T$ representing PPG measurements collected from the l^{th} individual upto and including time ($t \leq T^{(l)}$). Finally, vector $\boldsymbol{b}^{(l)}(t) = [B_1^{(l)}, \ldots, B_t^{(l)}]^T$ represents the BP values from time instant 1 to $T^{(l)}$.

The goal of the BP-Net architecture is to learn a nonlinear function $\boldsymbol{h}(\boldsymbol{x}^{(l)}(t), \boldsymbol{p}^{(l)}(t))$ that takes as input ECG $\boldsymbol{x}^{(l)}(t)$ and PPG $\boldsymbol{p}^{(l)}(t)$ sequences and provides a predicted value $\hat{B}(t)$ for the BP at time t. The target of the network is to minimize the cost function $\mathcal{L}\left(B(t), \boldsymbol{h}\left(\boldsymbol{x}^{(l)}(t), \boldsymbol{p}^{(l)}(t)\right)\right)$ between all the results of the hypothesized functions $\boldsymbol{h}(\cdot)$ with ECG and PPG input sequences, and the actual output $B(t)$. The Mean Square Error is used for the cost function. In what follows, we describe different aspects of the proposed BP-Net architecture.

Casual Convolutions: A basic aspect of the proposed BP-Net architecture is that we want to make sure that the output $\hat{B}(t)$ estimated at time step t depends only on previous and current input samples (i.e., ECG $\boldsymbol{x}^{(l)}(t)$ and PPG $\boldsymbol{p}^{(l)}(t)$ sequences) and not on any "future" values. In other words, while during the training phase, we have access to future values of the input signals (ECG and PPG sequences), a network trained by using such information can not be practically used to provide real-time predictions due to information leakage. To address this issue, one needs to implement causal filters within a deep learning architecture. The basic approach in this regard is to train the model with no causality restrictions and, during the implementation phase, mask out those regions of the feature maps that are derived from future input values. This masking approach can be achieved by setting the associated parts of the filter kernel to zero at each stochastic gradient descent (SGD) update. This approach is, however, costly in terms of required/wasted computational resources as, more or less, half of the multiplication and addition operations are wasted. In the BP-Net architecture, we utilize an alternative approach, i.e., the *Causal Convolutions* [127] are incorporated within the BP-Net architecture. Fig. 3.3 illustrates one example of casual convolutions. In causal convolutions, by capitalizing on the translation-equivalence property of the convolution, the input signal is first shifted and padded by the kernel size and then the introduced shifting is removed.

For long sequences, architectures with causal convolutions are much faster to train

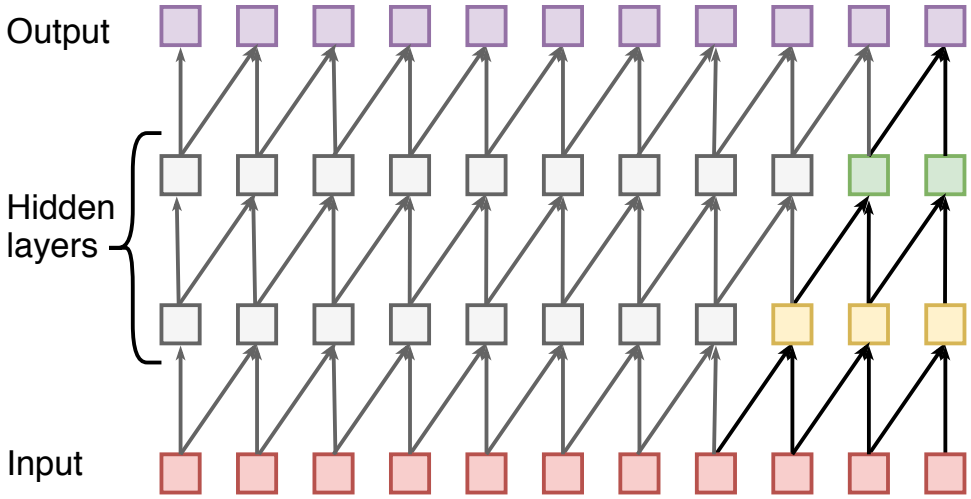

Figure 3.3: Causal Convolution.

than RNNs due to the absence of recurrent connections. However, one of the main drawbacks of the CNN-based models is the limited receptive field. For example, in Fig. 3.3, the receptive field of the neurons at the output layer is only 4, i.e., they see effects from up to four previous input samples (limited history size). For increasing the receptive field in casual convolutions, very deep networks or large filters should be applied, which are not generally feasible approaches. To address this issue, therefore, dilated convolutions [128] are used within the BP-Net architecture as described below.

Dilated Convolutions: Dilated convolutions are used as an effective way to enlarge the receptive field within the BP-Net without losing resolution as shown in Fig. 3.4. Consider a 1-D time series $\boldsymbol{x} \in \mathbb{R}^{N_x}$ and a 1-D Kernel $\mathcal{K} : \{0, 1, \ldots, R-1\} \rightarrow \mathbb{R}$ with size R. Discrete dilated convolution operation $D(p)$ on the p^{th} element of vector $\boldsymbol{x}$ with dilation rate L is defined as follows

$$D(p) \triangleq (\boldsymbol{x} *_L \mathcal{K})(p) = \sum_{i=0}^{R-1} \mathcal{K}(i) \times \boldsymbol{x}(p - L \times i), \tag{1}$$

where dilated convolution is denoted by $*_L$, and $(p - L \times i)$ refers to elements of the input vector $\boldsymbol{x}$ prior to time p^{th} element. Fig. 3.4 provides an illustration of dilated convolution. The dilation rate (L) is a design parameter, and dilated convolution becomes similar to the regular convolution with a dilation rate of $L = 1$. Generally

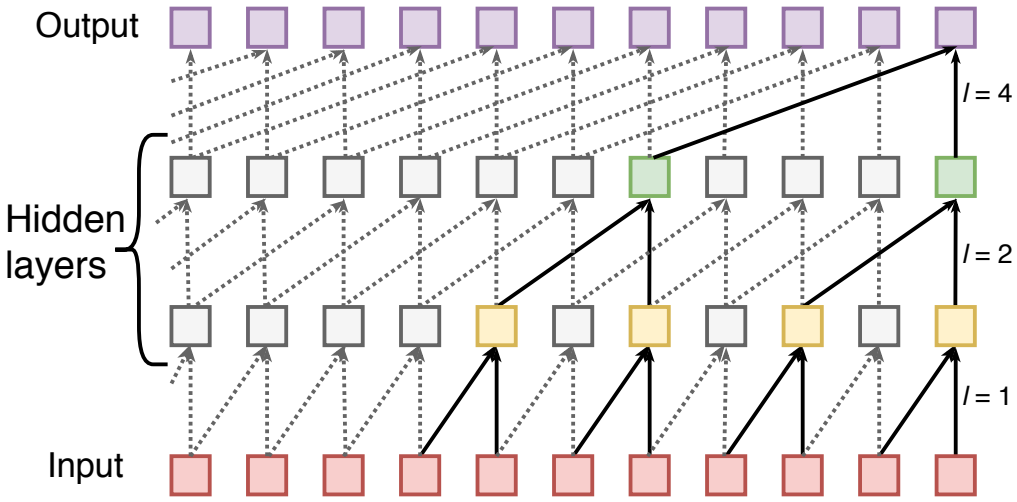

Figure 3.4: Dilated Causal Convolution.

speaking, in dilated convolution, the receptive field of a kernel $\mathcal{K}$ with size R is expanded to $R + (R-1)(L-1)$ with dilated stride of L.

Residual Connections: In theory, by stacking more layers, we expect that the network achieves lower training error and learns better, however, in practice, it is very challenging to train very deep neural networks due to the vanishing and exploding gradient problems. In other words, by adding more layers to a deep neural network, accuracy gets saturated and then degrades promptly. This problem is called "degradation", which was introduced by [129]. In [129], the authors, addressed the degradation problem by adding shortcut connections to the network referred to as "residual connections". In this case, by skipping over some layers, information passes into the network. Fig. 3.5(a) shows a sample of "Identity Block" which is a standard block used in ResNets. An identity Block is used when the input and output dimensions of the block are the same. If the block's input and output dimensions do not match, another type of ResNets block called "Resnet Convolutional Block" is employed to match dimensions by using 1×1 convolutions as illustrated in Fig. 3.5, φ represents the activation function, which applied on the element-wise addition of residual mapping function (F) and the input (x).

BP-Net Structure and Hyperparameters Settings: Inspired by Reference [95],

(a) (b)

Figure 3.5: Residual learning (a) Identity block. (b) Convolutional block.

the residual block as a basic block is used for the network structure. As shown in Fig. 3.1, this block consists of two dilated casual convolutions and two nonlinear activation functions (ELU) [130]. Moreover, dropout [131] for regularization, and weight normalization [132] are used in this block. We use Adam optimizer as the optimization algorithm with the learning rate set to 0.001. The learning rate changes in a cycle with the length of 100 epochs. After 20 epochs, we divide the learning rate by 2, but after 100 epochs instead of dividing by 2, we multiply it by 14.4. Therefore, the learning rate at the beginning of each cycle will be 90% of the learning rate at the beginning of the previous cycle. This novel approach of creating a learning rate helps to avoid being stuck in local minimums while speeding up the training process. These models are trained with a mini-batch size of 64. The exact structure of the proposed BP-Net network is as follows:

- PPG and ECG signal as inputs are separately fed to a 1×1 convolutions layer with 32 kernels, and then the concatenated channel-wise results are fed to the first residual block.
- Six residual blocks are stacked in the proposed architecture with the following characteristics:
  - All the dilated causal convolutions have kernel size of 5.
  - The dilation factor (L) is doubled for every layer, i.e., 1, 2, 4, 8, 16, and 32.
  - For Residual block 1 to 6, the number of kernels are 32, 32, 64, 64, 128, and 256, respectively.
- The output of the sixth block is fed to a 1×1 convolution layer with 256 kernels followed by an ELU activation function, (1×1) convolution layer with 2 kernels, and again an ELU activation function.

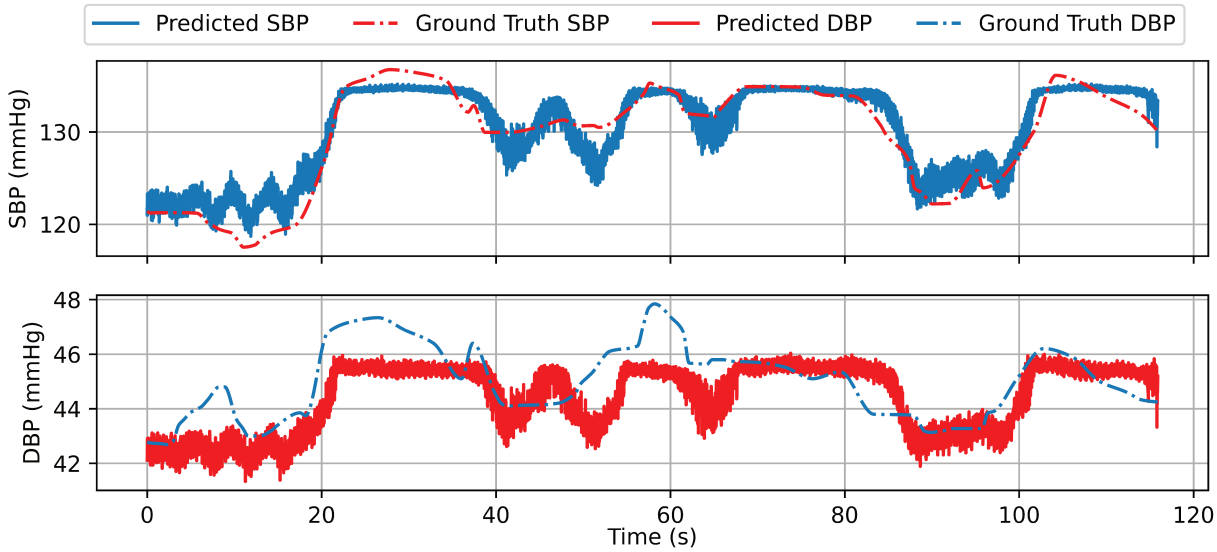

Figure 3.6: Comparison between predicted and reference SBP and DBP. The scale of y-axis is different in SBP and DBP subplots.

This completes the presentation of the proposed BP-Net architecture. Next, we present experimental results to evaluate performance of the proposed architecture.

3.2 Experiments and Results

In this section, we present different experimental results based on real-data sets to evaluate the performance of the proposed continuous BP estimation architecture. Details of the constructed dataset are described in Section 3.1. Fig. 3.6 shows the continuous SBP and DBP tracking of an individual using the proposed BP-Net architecture. From this figure, it is observed that the proposed model is capable of reliably tracking continuous SBP and DBP. It is worth mentioning that in contrary to existing BP estimation solutions, the proposed framework provides continuous SBP and DBP outputs for all samples. Intuitively speaking, the output of the proposed framework can be considered as upper and lower bounds on the BP signal computed at all time samples. To measure the accuracy of the proposed architecture, commonly used evaluation metrics, i.e., the root mean square error (RMSE) and the mean absolute error

(MAE), are used given by

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{j=1}^{n}|B_j - \hat{B}_j|^2}, \quad \text{and} \quad \text{MAE} = \frac{1}{n}\sum_{j=1}^{n}|B_j - \hat{B}_j|, \tag{2}$$

where B_j is the actual observed BP, whereas $\hat{B}_j$ determines its corresponding predicted value. Table 3.2, summarizes the average RMSE and MAE results obtained from averaging individual RMSE and MAE values corresponding to each subject (293 subjects). Table 3.2 also illustrates the combined MAE and RMSE results obtained by stacking together the target signals of all the subjects. The overall results presented in Table 3.2 show exceptional performance of the proposed BP estimation framework especially for estimating the DBP.

Table 3.2: The RMSE/MAE between ground truth BP(SBP, DBP) and estimated BP in the proposed model.

	Data Split Train:Validation:Test	Average RMSE (mmHg)	Average MAE (mmHg)	RMSE (mmHg)	MAE (mmHg)
SBP	7 : 1 : 2	3.03 ± 1.97	2.59 ± 1.78	3.59	2.57
DBP		1.58 ± 1.19	1.33 ± 1.03	1.97	1.32
SBP	4 : 2 : 4	3.59 ± 2.19	3.05 ± 2.02	4.16	3.02
DBP		1.79 ± 1.16	1.48 ± 0.99	2.12	1.47
SBP	2 : 2 : 6	3.90 ± 2.84	3.25 ± 2.51	4.72	3.21
DBP		1.97 ± 1.47	1.61 ± 1.25	2.41	1.59

We would like to point out that one general issue in BP estimation task is the time- and user-dependent nature of the ECG, PPG, and blood pressure signals. Variations in the probability distribution of these biomedical signals across different subjects make the experience gained on an unseen person difficult. Therefore, domain adaptation methods are highly recommended in this field of study, where learning methods focus on transferring information between a source and a target domain despite the existence of a distribution mismatch among them. However, in this study, training of the network is fully supervised, therefore, following previous studies [48, 50, 52, 56], the data for each patient was divided into train, validation, and test sets. We consider this as a limitation for our current study and a future direction of research. It is worth mentioning that, even in subject-independent training approaches [34, 50, 53], it

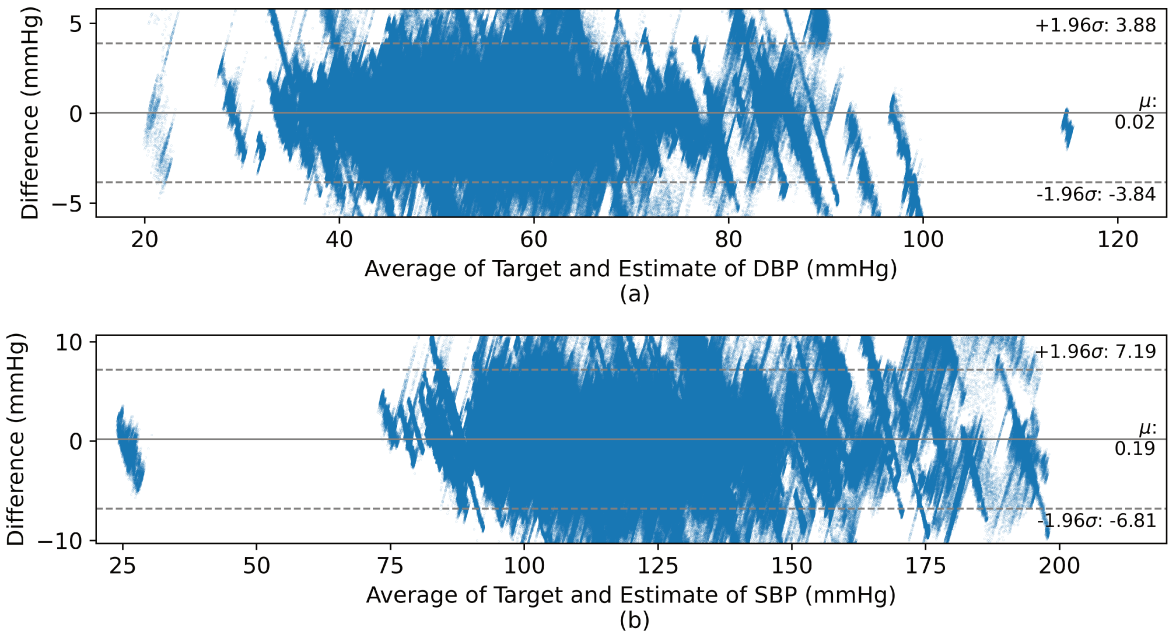

Figure 3.7: Bland-Altman plot of (a) The DBP, and; (b) The SBP. The limits of agreement (LOA) for DBP and SBP are $[-3.84, 3.88]$ and $[-6.81, 7.19]$, respectively.

is common to use some portion of the test set for personalization/fine-tuning models. To imitate this behavior, we varied the size of the training set. In other words, to further investigate this context and also check for potential overfitting issues, we have conducted several experiments by reducing the size of the training set as shown in Table 3.2. It is worth noting that to prevent potential information leakage between train, validation, and test, the overall pipeline for the data division is as follows: (i) train set division; (ii) skip 500 samples; (iii) splitting validation set; (iV) skip another 500 samples, and; (V) keep the remaining data for the test set. The overall results presented in Table 3.2 show the exceptional performance of the proposed BP-Net model even with a significant reduction in the training size.

3.2.1 Statistical Analysis

Fig. 3.7 shows the Bland-Altman plot of the SBP and the DBP estimation. For the proposed BP-Net architecture, the limits of agreement $[\mu - 1.96\sigma, \mu + 1.96\sigma]$ for SBP and DBP have been found to be [-6.81, 7.19] and [-3.84, 3.88] respectively. This

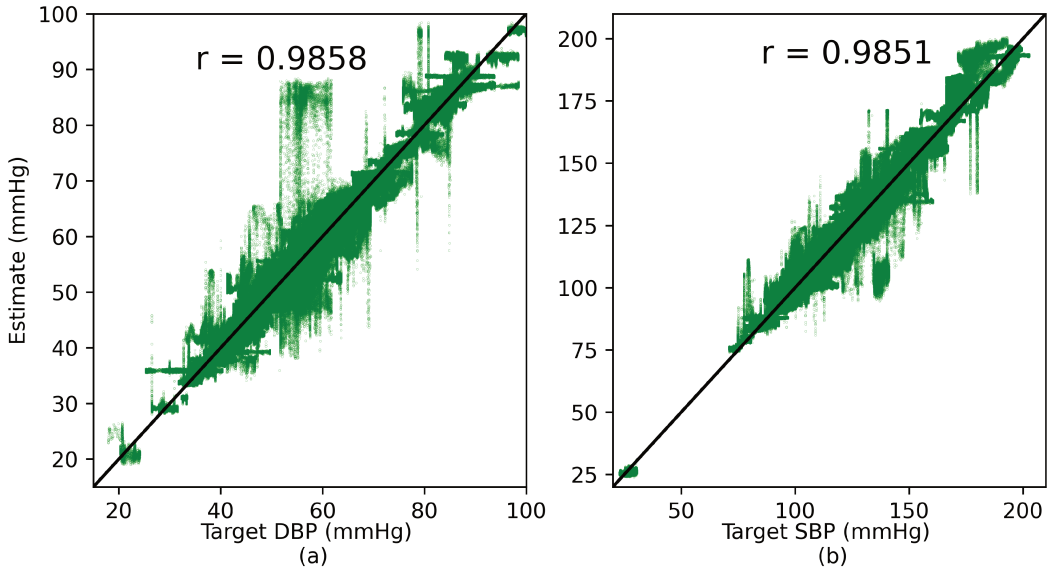

Figure 3.8: The regression plot for (a) The DBP, and; (b) The SBP. Pearson's correlation coefficients are $r = 0.9858$ and $r = 0.9851$ for DBP and SBP, respectively.

means that 95% of the estimated SBPs have error less than 7 mmHg and 95% of the measured DBPs have an error less than 3.86 mmHg, which indicates that the model provides acceptable estimates. Finally, Figs. 3.8(a)-(b) illustrate the regression plot for SBP and DBP estimation. The Pearson correlation coefficients of DBPs and SBPs are $r = 0.9858$ and $r = 0.9851$, respectively. Both of the coefficients are very close to 1.0 indicating high linearity between the target and estimated BP.

3.2.2 Comparisons

As previously mentioned, the lack of standardized and well-defined datasets poses a significant challenge in the field of blood pressure estimation. Recent literature utilizes datasets that are diverse, inconsistent, and sourced from various origins, leading to difficulties in comparing and evaluating the effectiveness of deep learning-based algorithms. In light of this issue, we have compiled a comprehensive summary of state-of-the-art research results [34,48,50–53,55,56], encompassing not only their performance but also the specific data utilized and the methodologies employed. These

Table 3.3: Comparison with state-of-the-art researches.

Author	**Data Used**	**Data Split**	**Model**	**Signal**	**Error**	
		Train:Validation:Test			**SBP**	**DBP**
Kachuee et al. [34]	MIMIC II 1000 subjects 10 min	8 : 1 : 1	classical ML (AdaBoost)	ECG, PPG	MAE: 11.17	MAE: 5.35
Su et al. [48]	Proprietary data 84 subjects 10 min	7 : 1 : 2	Deep Learning (LSTM)	ECG, PPG	RMSE: 3.73	RMSE: 2.43
Tanveer et al. [50]	MIMIC 1 39 subjects	7 : 1 : 2	Deep Learning (LSTM + ANN)	ECG, PPG	MAE: 0.93 RMSE: 1.27	MAE: 0.52 RMSE: 0.73
Eom et al. [56]	Proprietary data 15 subjects 30 min	7 : 1 : 2	Deep Learning (CNN + Bi-GRU + Attention)	ECG, PPG, BCG	MAE: 4.06	MAE: 3.33
Qin et al. [53]	MIMIC II 1227 records	Categorize records into 3 classes (Normal, Prehypertension, Hypertension) For Each Class → 6 : 2 : 2	Deep Autoencoder (DAE) model	PPG	MAE: 7.95	MAE: 4.11
Fan et al. [52]	MIMIC II	8 : 1 : 1	Deep Learning (Res2Net)	ECG (lead II)	MAE: 7.69 RMSE: 12.30	MAE: 4.36 RMSE: 6.88
Paviglianiti et al. [51]	MIMIC 40 subjects	-	Deep model (ResNet+LSTM)	ECG (lead V), PPG	MAE: 4.12 RMSE: 5.68	MAE: 2.23 RMSE: 2.97
This study	MIMIC 293 subjects 140.43 hours	7 : 1 : 2	Deep model (TCN)	PPG ECG (Multi-Lead I, II, III, V, AVR, MCL)	MAE: 2.59 RMSE: 3.03	MAE: 1.33 RMSE: 1.58
		4 : 2 : 4			MAE: 3.05 RMSE: 3.59	MAE: 1.48 RMSE: 1.79
		2 : 2 : 6			MAE: 3.25 RMSE: 3.90	MAE: 1.61 RMSE: 1.97

findings are presented in Table 3.3, allowing for a comprehensive comparison and analysis of different approaches in the field. While it may not be fair to directly compare models based on their error metrics due to the variability in data, Table 3.3 sheds light on the diverse data sources, number of subjects, total duration of data, and methodologies employed in the recent literature, providing valuable insights for researchers and practitioners seeking to enhance blood pressure estimation through machine learning techniques. In what follows we discuss and compare the proposed BP-Net with its state-of-the-art counterparts:

- As shown in Table 3.3, the authors in the previous studies [48,50,51,56] trained and validated their model on a few numbers of subjects (e.g., 15, 39, 40, or 84) which is not enough for establishing a model that can potentially be used broadly for general subjects. While we extend the number of subjects to 293 patients with a total duration of 140.43 hours of data.

- In the studies presented so far such as References [51,52], the lead of ECG signal is not considered, i.e., the BP is estimated based on a single lead of ECG. The availability of different leads of ECG signals makes it difficult to evaluate the generality of the obtained results. As stated previously in subsection 3.1.1, to address this problem the prepared dataset consists of different leads of ECG signal, and a single model is trained using all leads of ECG signal.

- It is worth mentioning that the basic building block of the proposed BP-Net architecture is Dilated Causal Convolutions, therefore, it can be categorized as a Temporal Convolutional Network (TCN) instead of RNN. TCN architecture provides several advantages over RNNs such as faster training with lower memory requirements and more stable training because of avoiding the problem of gradient vanishing/explosion. Performance of temporal convolutional and recurrent architectures are evaluated in Reference [95], where it was shown that TCN architecture outperforms canonical RNNs across a comprehensive suite of tasks and datasets. Consequently, by inheriting these characteristics of the TCN, the proposed BP-Net architecture is expected to be more accurate than its counterparts developed based on canonical recurrent networks. To further support this conclusion, Table 3.3 also compares performance of the proposed

BP-Net architecture with RNN-based approaches [48, 50, 51, 56]. It can be observed that even when the size of the training set is decreased, BP-Net has comparable performance to that of the RNN-based models.

3.3 Conclusion

In this chapter, we proposed a deep learning-based architecture (named as BP-Net) for continuous, cuff-less, and alignment-free BP estimation, which utilizes raw ECG and PPG signals as inputs, unlike the common methods in BP estimation literature, which use engineered and pre-defined physiological features as the first building block of their recommended approaches. By utilizing raw signals (ECG and PPG) as inputs, without hand-crafted extraction of features, we explore the real potential of deep learning in the utilization of intrinsic features (deep features) of the input signals. In addition, a benchmark data set is being prepared that has the potential to provide a unified framework for the evaluation and comparison of deep learning-based BP estimate techniques. The dataset is publicly available via the link provided in Reference [126].

Chapter 4

Improving Accuracy for Hand Gesture Recognition

In recent years, there has been a growth of interest in developing deep learning-based approaches for HGR, which show encouraging classification results [58, 89]. In particular, deep learning techniques provide an effective venue to automatically extract features from sEMG data and improve gesture recognition accuracy compared to their classical counterparts. However, many of the existing deep learning approaches involve only a single model, which may not effectively extract representative features and can cause a reduction in performance. In this chapter, we address this gap by designing a Transformer-based hybrid solution that has great potential for extracting special and spatio-temporal representation to improve HGR accuracy. In particular, capitalizing on the recent success of Transformers in various fields of Machine Learning [15–17, 133], we aim to examine its applicability and potential for sEMG-based hand gesture recognition. More precisely, the Transformer-based models offer several advantages over RNN- and CNN-based deep neural networks. Firstly, the Transformer model has a self-attention mechanism that allows it to capture long-range dependencies in the input data more effectively than RNNs, which are prone to the vanishing gradient problem. This makes it particularly suitable for sEMG-based hand gesture recognition, as it enables the model to capture temporal dependencies and complex patterns in sequential electromyography signals. Secondly, Transformers are

inherently parallelizable, making them more efficient to train and evaluate compared to RNNs, which rely on sequential computation. This parallelism can be crucial for real-time hand gesture recognition applications, where low latency is desired. Lastly, the Transformer architecture does not have any spatial assumptions like CNNs, which makes it more flexible for capturing non-local relationships in the sEMG signals. This is beneficial as sEMG data can exhibit complex spatial patterns that may not be captured effectively by CNNs. Therefore, given the ability of Transformers to capture long-range dependencies, their parallelizable nature, and their flexibility in handling non-local relationships, they are a well-justified choice for sEMG-based hand gesture recognition tasks.

As illustrated in Fig. 4.1, our proposed framework for hand movement recognition, referred to as the Transformer for Hand Gesture Recognition (TraHGR), incorporates two parallel paths, namely the Special Transformer Network (SNet) and the Feature Transformer Network (FNet), followed by a linear layer. This approach is designed to effectively process special-temporal data in the form of surface electromyography (sEMG) signals. Specifically, the SNet network is responsible for extracting features from each sensor, while the FNet network focuses on extracting spatio-temporal features from the input sEMG data. The outputs of both paths are then integrated using a linear layer, resulting in a more comprehensive representation that augments the discriminating power of the model and enhanced the overall performance of the hand movement recognition task.

It is worth noting that sEMG signals are special-temporal data, which means that they contain information about both the spatial location of the sensors and the temporal changes in muscle activation over time. Therefore, an effective hand movement recognition model needs to take into account both types of features to accurately classify different hand movements. The proposed TraHGR framework achieves this by leveraging the strengths of both the SNet and FNet networks, resulting in a more accurate classification model for sEMG-based hand movement recognition.

The performance of the proposed TraHGR framework is evaluated using the second Ninapro database [87,134] referred to as DB2, which is a publicly available dataset that provides sparse multi-channel sEMG signals from various hand movements similar to those obtained in real-life conditions. The sEMG signals in the DB2 dataset are

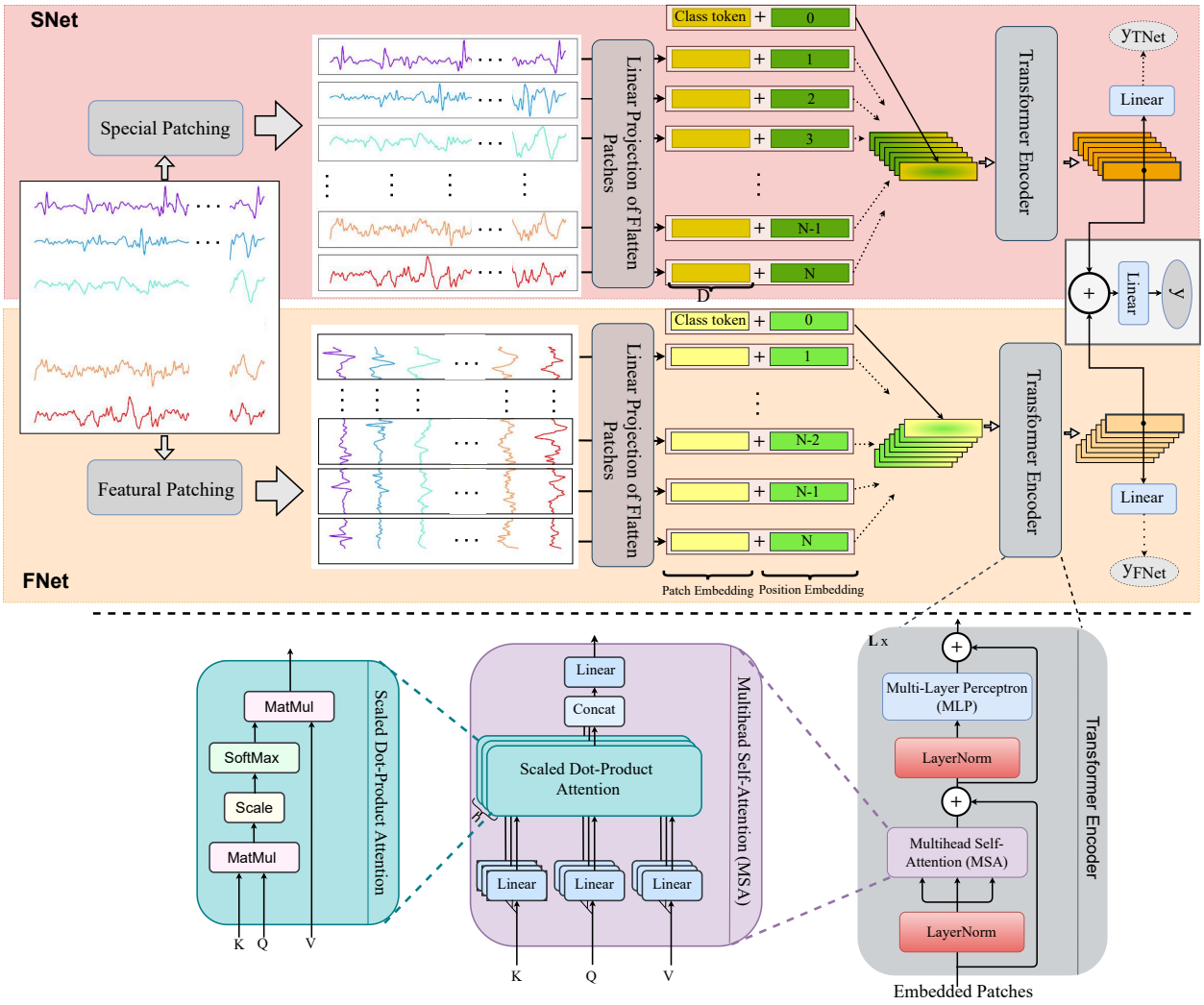

Figure 4.1: **The proposed TraHGR architecture** consists of two parallel paths (SNet and FNet). Each segment of sEMG signals $\boldsymbol{X}$ is divided into N non-overlapping patches. The TraHGR uses the SNet path to get the special patches while simultaneously the FNet is utilized to consider the featural patches including both special and temporal information. In both SNet and FNet, the patches are mapped linearly into the model dimension D. We refer to the output of this step as "Patch Embedding". Then, a "class token" is prepended to the sequence of patch embeddings which is finally used for the classification purpose. The "Positional Embedding" is added to the "Patch Embedding" to retain the positional information. The result is fed to the Transformer encoder consisting of $\mathcal{L}$ layers, each layer consisting of Multi-head Self Attention (MSA) and Multi-Layer Perceptron (MLP) modules. Finally, we add the output of the SNet and FNet class tokens to get the final representation, which then acts as the input to the linear layer.

measured in real-life conditions from 40 healthy users, each performing 49 gestures. Thus, Ninapro dataset enables development of innovative DNN-based recognition

solutions for HGR tasks. We conduct an extensive set of experiments to test and validate the proposed TraHGR architecture and compare its achievable accuracy with several recently proposed HGR classification algorithms based on the same datasets. Results show that the proposed TraHGR framework provides superior performance over all its counterparts on the DB2 dataset and its sub-exercises. More specifically, The DB2 dataset is presented in three sub-exercises; i.e., DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures). TraHGR classifies a high number (49) of gestures with a high accuracy. More specifically, compared with the proposed architectures in the recent state-of-the-art studies, TraHGR improves the recognition accuracies to 86.18% on DB2 (49 gestures), to 88.91% on the DB2-B (17 gestures), to 81.44% on DB2-C (23 gestures), and to 93.84% on the DB2-D (9 gestures).

The rest of the Chapter is organized as follows: In Section 4.1, we describe the details of the proposed TraHGR architecture. The experiments and results are presented in Section 4.2. Finally, the conclusion is presented in Section 4.3.

4.1 The TraHGR Framework

This section provides a detailed explanation of the TraHGR architecture that has been proposed for hand gesture recognition. The architecture is designed based on the Transformers in which the attention mechanism is employed. The attention mechanism has been used in previous studies [58, 83, 99] in conjunction with CNNs and/or recurrent-based architectures for HGR task. However, the proposed Transformer-based architecture relies solely on attention mechanisms and outperforms the previous studies in which CNN, RNN, and hybrid architectures (e.g., attention-based hybrid CNN-RNN) have been adopted. The overall proposed architecture is illustrated in Fig. 4.1, which is inspired by the Vision Transformer (ViT) [17], in which each input is divided into patches, and the network is supposed to perform label prediction based on the sequence of patches. The patching mechanism is a technique used in transformers to reduce computational requirements and enhance the ability to capture features of long sequences. As shown in Fig. 4.1, the proposed TraHGR consists of a SNet path implemented in parallel with a FNet path followed by a linear layer,

which acts as the fusion center combining the extracted features from each of the two parallel paths in order to classify the hand gestures. In the following, we will further elaborate on the details of the proposed TraHGR architecture.

4.1.1 Patching and Embedding

In this sub-section, we focus on the input of the Transformer encoder, which is a sequence of embedded patches. As illustrated in Fig. 4.1, the embedded patches are constructed from patch embeddings and position embeddings, which are described below.

Patching: Even though sEMG data is naturally time-series data, the patching mechanism is still utilized in our approach for feeding the data to transformers. This is because the transformer model has a quadratic growth in computation with the length of the input sequence. By dividing the input sequence into smaller, i.e., fixed-size patches, we can reduce the computational requirements of the transformer model, making it more efficient and effective from both the memory consumption and computation perspective.

Each input from the sEMG signal segmentation phase is denoted by $\boldsymbol{X} \in \mathbb{R}^{S \times \boldsymbol{w} \times \boldsymbol{C}}$, where S shows the number of sensors in the DB2 dataset, $\boldsymbol{w}$ shows the number of samples of electrical activities of muscles obtained at the rate of 2 kHz for a window of 200ms, 150ms, or 100ms, and $\boldsymbol{C}$ denotes the number of channels of the sEMG signals. We split each segment of sEMG signals $\boldsymbol{X}$ into non-overlapping patches $\boldsymbol{X_p} = \{\boldsymbol{x}_p^i\}_{i=1}^N$. More specifically, each segment $\boldsymbol{X} \in \mathbb{R}^{S \times \boldsymbol{w} \times \boldsymbol{C}}$ is divided into N non-overlapping patches in which each patch is flattened. We represented the sequence of these flattened patches with $\boldsymbol{X_p} \in \mathbb{R}^{N \times (P_1.P_2.\boldsymbol{C})}$, where ($P_1$, P_2) shows the size of each patch, and $N = S.\boldsymbol{w}/(P_1.P_2)$ represents the length of this sequence, i.e., the number of patches. As shown in Fig. 4.1, we applied two types of patching:

- *Special Patching:* Here, the size of each patch is (1, $\boldsymbol{w}$); therefore, the number of patches is $N = S$. We refer to this patching technique as Special Patching because each patch contains information from only one of the sensors in the dataset for a sequence with a length of $\boldsymbol{w}$. Therefore, Special Patching as

the first building block of the SNet path provides temporal changes in muscle activation over time at each patch corresponding to a specific sensor.

- *Featural Patching:* We set the size of each patch to (S, S), i.e., $P_1 = P_2 = S$, therefore, the number of patches is $N = \boldsymbol{w}/S$. We refer to this type of patching as Featural because each patch contains the information of all S sensors for a sequence with a length of S. Therefore, both spatial and temporal information are included in a Featural patch. The Featural patches are provided as the input only to the FNet layer as shown in Fig. 4.1.

Patch Embeddings: As shown in Fig 4.1, after patching mechanisms are applied to the input data, a linear mapping is applied to create the embedding vectors form each patch. To achieve this, a shared matrix $\mathbf{E} \in \mathbb{R}^{(P_1.P_2.\boldsymbol{C})\times D}$ is used to linearly project each patch $\boldsymbol{x}_{\boldsymbol{p}}^{i}$ to a D dimensional vector (as shown in Eq. (3)). It is important to note that both Special and Featural Patching have their own corresponding projection matrix. The output of this projection is known as the "patch embeddings".

Class Token: Similar to the BERT framework [16], a trainable embedding is prepended to the sequence of patch embeddings ($\boldsymbol{Z}_0^0 = \boldsymbol{x}_{\text{class}}$) with the goal of capturing the meaning of the entire segmented input as a whole. More specifically, the class token's embedding after the last Transformer encoder layer ($\boldsymbol{Z}_L^0$) is used for classification purposes (Eq. (11)).

Position Embeddings: As HGR based on sEMG signals is a time-series processing task, the order of data is an essential part of sequence modeling. Recurrent-based architectures such as LSTM inherently consider signal order, however, Transformers do not process the input sequentially and combine the information of all the elements through an attention mechanism. Therefore, there is a need to encode the order of each element in the sequence. This is where positional embedding comes in. In fact, position embedding allows the network to determine where a particular patch came from. There are several ways to retain position information at the Transformer input, e.g., Sinusoidal positional embedding, 1-dimensional positional embedding, 2-dimensional positional, and Relative positional embeddings embedding [17, 133]. Following [17], we used the standard trainable 1-dimensional positional embeddings. As shown in Fig. 4.1, position embeddings indicated by $\mathbf{E}_{pos} \in \mathbb{R}^{(N+1)\times D}$ is added to

the patch embeddings. The formulation which governs patch and position embeddings is as follows

$$\boldsymbol{Z}_0 = [\boldsymbol{x}_{\text{class}}; \boldsymbol{x}_{\boldsymbol{p}}^1\mathbf{E}; \boldsymbol{x}_{\boldsymbol{p}}^2\mathbf{E}; \ldots; \boldsymbol{x}_{\boldsymbol{p}}^N\mathbf{E}] + \mathbf{E}_{pos}. \tag{3}$$

where $[;]$ represents concatenation operation. The output of Eq. (3) is fed as an input to the Transformer encoder.

4.1.2 Transformer Encoder

The Transformer encoder takes the $\boldsymbol{Z}_0$ as an input. This block is inspired by the main Transformer encoder introduced in [14], which treats all embedded patches as tokens. As illustrated in Fig. 4.1, the Transformer encoder consists of $\mathcal{L}$ layers. Each layer contains two modules, namely the Multihead Self-Attention (MSA) and a Multi-Layer Perceptron (MLP) module, i.e.,

$$\boldsymbol{Z}_l^{'} = \quad MSA(LayerNorm(\boldsymbol{Z}_{(l)-1})) + \boldsymbol{Z}_{(l)-1}, \quad {}^{(l)} = 1 \ldots \mathcal{L} \tag{4}$$

$$\boldsymbol{Z}_l = MLP(LayerNorm(\boldsymbol{Z}_{(l)}^{'})) + \boldsymbol{Z}_{(l)}^{'}, \qquad {}^{(l)} = 1 \ldots \mathcal{L} \tag{5}$$

It is worth noting that a layer-normalization [135] is used before MSA and MLP modules, and the residual connections are applied to address degradation problem. The MLP module consists of two linear layers in which the first layer is followed by Gaussian Error Linear Unit (GELU) activation function. Moreover, the MSA module is defined based on the Self-Attention (SA) mechanism, which is discussed next.

Self-Attention (SA): The SA mechanism [14] measures the pairwise similarity of each query and all the keys and obtains a weight for each value. Finally, the output is computed based on the weighted sum over all values. In particular, if we define an input $\boldsymbol{Z} \in \mathbb{R}^{N \times D}$ consisting of N vectors, each of length D, the three matrices, i.e., Queries $\mathbf{Q}$, Keys $\mathbf{K}$, and Values $\boldsymbol{V}$, are calculated as follows

$$[\mathbf{Q}, \mathbf{K}, \boldsymbol{V}] = \boldsymbol{Z}\boldsymbol{W}_{QKV}, \tag{6}$$

where $\boldsymbol{W}_{QKV} \in \mathbb{R}^{D \times 3D_h}$ denotes the trainable weight matrix and D_h shows the length of each vector in $\mathbf{Q}$, $\mathbf{K}$, and $\boldsymbol{V}$. To measure the weights for $\boldsymbol{V}$, the dot-product of $\mathbf{Q}$ and $\mathbf{K}$ is calculated, then scaled with $\sqrt{D_h}$. These weights are converted to the probabilities $\mathbf{P} \in \mathbb{R}^{N \times N}$ using the softmax function as follows

$$\mathbf{P} = softmax(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{D_h}}). \tag{7}$$

Finally, the output of the SA mechanism is computed as follows

$$SA(\boldsymbol{Z}) = \mathbf{P}\boldsymbol{V}. \tag{8}$$

By using the attention mechanism, the model pinpoints a piece of specific information in the input sequence.

Multihead Self-Attention (MSA): Here, the SA mechanism is used for $\boldsymbol{h}$ times in parallel, allowing the architecture to pinpoint specific pieces of information in the input sequence for each head differently. In particular, each head has its own trainable weight matrix. The final matrix in the MSA mechanism is a projection of the concatenated outputs of the $\boldsymbol{h}$ heads, which is formulated as follows

$$MSA(\boldsymbol{Z}) = [SA_1(\boldsymbol{Z}); SA_2(\boldsymbol{Z}); \ldots; SA_h(\boldsymbol{Z})]\boldsymbol{W}_{MSA}, \tag{9}$$

where $\boldsymbol{W}_{MSA} \in \mathbb{R}^{\boldsymbol{h}.D_h \times D}$. Here, D_h is set to $D/\boldsymbol{h}$ to keep the number of parameters constant when $\boldsymbol{h}$ changes.

4.1.3 TraHGR's Output

As shown in Fig. 4.1, the TraHGR consists of two paths, i.e., SNet and FNet. For each path, the aforementioned calculations (Eqs. (3)-(9)) are performed in parallel. Then, the predicted class labels of each path is calculated based on its corresponding $\boldsymbol{Z}_L^0$ as follows

$$\mathrm{y}_{path} = Linear(LayerNorm(\boldsymbol{Z}_L^0)_{path}), \tag{10}$$

Table 4.1: The number of parameters in different variants of FNet, SNet, and TraHGR architectures with respect to the number of layers, model dimension (D), and the number of heads ($\boldsymbol{h}$) and MLP size in Transformer Encoder. The number of parameters (#Params) is reported for window sizes 200ms, 150ms, and 100ms.

Model	**#Layers ($\mathcal{L}$)**	**Model dimension (D)**	**MLP size**			**#Heads ($\boldsymbol{h}$)**	**#Params**		
			200**ms**	150**ms**	100**ms**		200**ms**	150**ms**	100**ms**
TraHGR-Base	1	32		128		4	83,731	74,259	63,603
TraHGR-large	2	64		256		4	316,051	297,107	275,795
TraHGR-Huge	1	144		720		8	846,579	803,955	756,003
SNet	1	144		720		8	472,513	431,041	384,385
FNet	1	144		720		8	366,673	365,521	364,225
SNet-Huge	1	200	1084	1120	1162	8	846,733	803,569	756,611
FNet-Huge	1	224	1176	1085	1102	8	846,377	803,726	756,247

where $path \in \{\text{SNet}, \text{FNet}\}$. Finally, the output of the TraHGR is calculated based on the sum of $\boldsymbol{Z}_L^0$ in the SNet and FNet as follows

$$\text{y} = Linear(LayerNorm[(\boldsymbol{Z}_L^0)_{\text{SNet}} + (\boldsymbol{Z}_L^0)_{\text{FNet}}]). \tag{11}$$

It is worth mentioning that y_{SNet}, y_{FNet}, and y are used for TraHGR training. More details are provided in the subsection 4.2.1. This completes description of the proposed TraHGR architecture, next, its performance is evaluated through several experiments.

4.2 Experiments and Results

In this section, we evaluate the performance of the proposed TraHGR architecture through a series of experiments. In all experiments, the Adam optimizer [136] was used with the learning rate of 0.0001 and the weight decay of 0.001. Moreover, the batch size is set to 512. Table 4.1 shows the different configurations of the hyperparameters in the TraHGR architecture resulting in different variants of the model denoted by TraHGR-Base, TraHGR-large, and TraHGR-Huge. These variants are then used for training and evaluation purposes with different window sizes of 100ms, 150ms, and 200ms. Moreover, we evaluated the performance of a single deep model (SNet or FNet) when they are trained independently. In Table 4.1, the number

Table 4.2: Comparing different variants of TraHGR. The average accuracy of hand gesture recognition across all subjects in the DB2 (49 gestures) dataset for different variants of TraHGR architecture on several window sizes (200ms, 150ms, and 100ms).

Model	**Accuracy ± STD**		
	200**ms**	150**ms**	100**ms**
TraHGR-Base	78.60 ± 6.03	77.54 ± 5.99	76.17 ± 6.09
TraHGR-large	83.58 ± 5.48	82.58 ± 5.60	81.30 ± 5.87
TraHGR-Huge	**86.18 ± 4.99**	**85.43 ± 5.24**	**84.13 ± 5.21**

of parameters (Params) is calculated for DB2 (49 gestures) while this number will be less for DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures).

4.2.1 Loss Function

The loss function $\mathcal{L}$ of TraHGR consists of the following three components

$$\mathcal{L} = \mathcal{L}_{\text{SNet}} + \mathcal{L}_{\text{FNet}} + \mathcal{L}_{\text{TraHGR}}, \tag{12}$$

where the first term $\mathcal{L}_{\text{SNet}}$ is loss of the SNet path in the proposed TraHGR architecture. More specifically, the cross-entropy loss is considered for measuring classification performance using the SNet's output y_{SNet} (Eq. (10)) and the target values. Similarly, the second term $\mathcal{L}_{\text{FNet}}$ is the cross-entropy loss computed using the second path (FNet) of the TraHGR architecture where FNet's outputs y_{FNet} (Eq. (10)) are considered. Finally, the last term $\mathcal{L}_{\text{TraHGR}}$ is calculated using the TraHGR's output y (Eq. (11)).

4.2.2 Evaluation of the Proposed TraHGR Architecture

This subsection provides evaluations on the prediction performance of the proposed hybrid transformer-based architecture. In this regard, first, we compare different

Table 4.3: Comparison of architectures with the same structure. The average accuracy of hand gesture recognition across all subjects in the DB2 (49 gestures) dataset for FNet, SNet, and TraHGR-Huge architectures on several window sizes (200ms, 150ms, and 100ms). As shown in Table 4.1, the network structure in SNet and FNet is not changed compared to the TraHGR-Huge structure.

Model	**Accuracy ± STD**		
	200**ms**	150**ms**	100**ms**
TraHGR-Huge	**86.18 ± 4.99**	**85.43 ± 5.24**	**84.13 ± 5.21**
SNet	83.39 ± 5.44	82.81 ± 5.60	81.43 ± 5.88
FNet	80.72 ± 5.82	80.05 ± 6.03	79.38 ± 6.15

Table 4.4: Comparison of architectures with the same scale. The average accuracy of hand gesture recognition across all subjects in the DB2 (49 gestures) dataset for SNet-Huge, FNet-Huge, and TraHGR-Huge architectures on several window sizes (200ms, 150ms, and 100ms). As shown in Table 4.1, the number of parameters in SNet-Huge and FNet-Huge is on the same scale as TraHGR-Huge.

Model	**Accuracy ± STD**		
	200**ms**	150**ms**	100**ms**
TraHGR-Huge	**86.18 ± 4.99**	**85.43 ± 5.24**	**84.13 ± 5.21**
SNet-Huge	83.80 ± 5.78	83.25 ± 5.34	82.21 ± 5.45
FNet-Huge	81.10 ± 5.68	80.44 ± 5.48	79.94 ± 5.83

variants of the TraHGR architecture and show the effect of different hyperparameters (e.g., number of layers, model dimension, MLP size, and number of heads) on the overall accuracy. Then, to demonstrate the performance of the hybrid transformer, we also compare the TraHGR architecture with single deep models, i.e., SNet and FNet, and their Huge versions.

Table 4.2, 4.3, and 4.4 show HGR recognition accuracy, which is averaged over all subjects for the test set. From Table 4.2, it can be observed that the proposed TraHGR-Huge architecture outperformed other TraHGR architecture variants

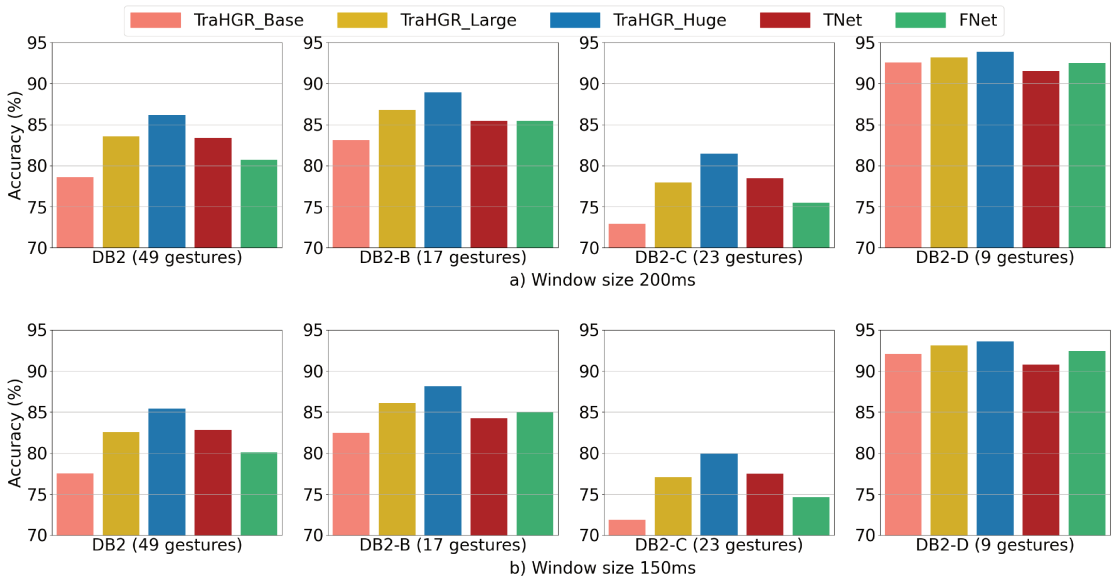

Figure 4.2: Breakdown of DB2 (49 gestures) performance in DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures) exercises.

(TraHGR-Base and TraHGR-large) when evaluated based on the DB2 (49 gestures) for the same window size. However, as shown in Table 4.1, the number of parameters of the TraHGR-Huge is much higher than that of the TraHGR-Base and TraHGR-large models. This fact indicates that increasing the number of layers ($\mathcal{L}$), model dimension (D), MLP size, and number of heads ($\boldsymbol{h}$) have a positive effect on the model's accuracy, however, this comes with the cost of increasing the complexity. In addition, as shown in Table 4.1, each model has a larger number of trainable parameters for window size 200ms than its counterpart in the window size of 150ms or 100ms, resulting in higher complexity. However, as shown in Table 4.2, a larger window size can further improve the results because the model has access to a longer sequence length.

4.2.3 TraHGR Hybrid Architecture Versus SNet and FNet

We independently trained and evaluated the proposed model on DB2 subsets, i.e., DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures). In Fig. 4.2, the

performance of the proposed architectures for DB2 (49 gestures) and its three sub-exercises, i.e., B, C, and D are shown. It can be observed that for both window sizes of 200ms and 150ms achieving a high accuracy for DB2-C is more challenging than DB2-B and DB2-D subsets. More specifically, DB2-C consists of 23 grasping and functional movements for which everyday objects like a bottle and knife are presented to the user for grasping, in order to mimic daily-life actions such as opening a bottle or cutting something [137]. Therefore, the performance reduction in the DB2-C subset is not far from expectation as the muscle groups which are predominantly used during movements of DB2-C are more complicated than basic hand posture and wrist movements in DB2-B and finger force patterns in DB2-D subsets.

As shown in Table 4.1, when comparing the number of trainable parameters in SNet and FNet against the different variants of proposed TraHGR architectures, TraHGR-large, although smaller, has the closest number of trainable parameters to these single networks. More precisely, for window sizes 200ms and 150ms, SNet has approximately 1.5× more parameters than TraHGR-large, and FNet is almost 1.2× larger. For the window size of 100ms, both single networks have almost 1.3× more parameters than TraHGR-large. However, as shown in Table 4.2 and 4.3, TraHGR-large has comparable performance to SNet, and it outperforms FNet, showing that a hybrid model with fewer number of parameters is capable to extract more generic representations resulting in comparable or even better performance compared to larger single networks, SNet and FNet. According to Table 4.1, the network structure in SNet and FNet is completely different than TraHGR-large. Therefore, we conducted a new experiment in which the structure of the single and hybrid networks remained unchanged. To do so, we can compare the performance of TraHGR-Huge against the SNet and FNet (see Table 4.1). As shown in Table 4.3, the TraHGR-Huge outperforms the single deep models (SNet and FNet) when the structure of the networks is preserved. However, since the number of trainable parameters is TraHGR-Huge is considerably larger than SNet and FNet, the performance improvement could be conducted due to the TraHGR-Huge capacity to represent more complex hypothesis space. As a result, we conducted new experiments in which the number of parameters for new variants of SNet and FNet architectures is expanded to be on the same scale as TraHGR-Huge. Specifically, to increase the number of parameters in new variants of SNet and FNet, we began by increasing model dimension D and stopped just before

exceeding the number of parameters in TraHGR-Huge. Then, the size of the MLP layer in the transformer encoder is enlarged to fill the remaining gap in terms of the number of parameters as much as possible, resulting in SNet-Huge and FNet-Huge architectures. Detailed information about the structure of different variants of these single networks and their number of parameters are provided in Table 4.1. As shown in Table 4.4, TraHGR-Huge significantly outperforms SNet-Huge and FNet-Huge architectures while they are all in the same scale.

As shown in Table 4.3 and 4.4, although the number of trainable parameters in SNet-Huge and FNet-Huge are significantly increased compared to SNet and FNet, their average recognition accuracy improvement for different window sizes is not significant. As a result, since the single networks were not capable to achieve high performance even with massive parameters expansion and given the outstanding performance of TraHGR-Huge architecture, it can be concluded that the hybrid approach integrates the advantages of two parallel paths to model better and more generic representation resulting in performance improvement. It is worth mentioning that for hybrid models such as TraHGR-Base, TraHGR-large, and TraHGR-Huge, the classification accuracy is calculated using the output of Eq. (11), while for single deep models such as SNet and FNet this number is computed using the output of Eq. (10).

4.2.4 Statistical Analysis

Following [99,138], we considered each user as a separate dataset and conduct Wilcoxon signed-rank test [139]. To do so, given that we have 40 users, for each model we will have 40 accuracies resulting from each user's test set. Having accuracies for each model, we performed statistical analysis on the effectiveness of the observations for DB2 (49 gestures) For the window size of 200ms.

According to the results shown in Fig. 4.3, the difference in accuracy between TraHGR-Huge and other proposed architectures such as TraHGR-Base, TraHGR-large, SNet, and FNet, for window sizes 200ms were considered statistically significant by the Wilcoxon signed-rank test. Worth to mention that, in Fig. 4.3, the p-value of significance is considered 0.05 and the annotated $*$ mark represents $p \leq 0.05$. Fig. 4.3 illustrates the performance distribution across 40 users for each proposed model.

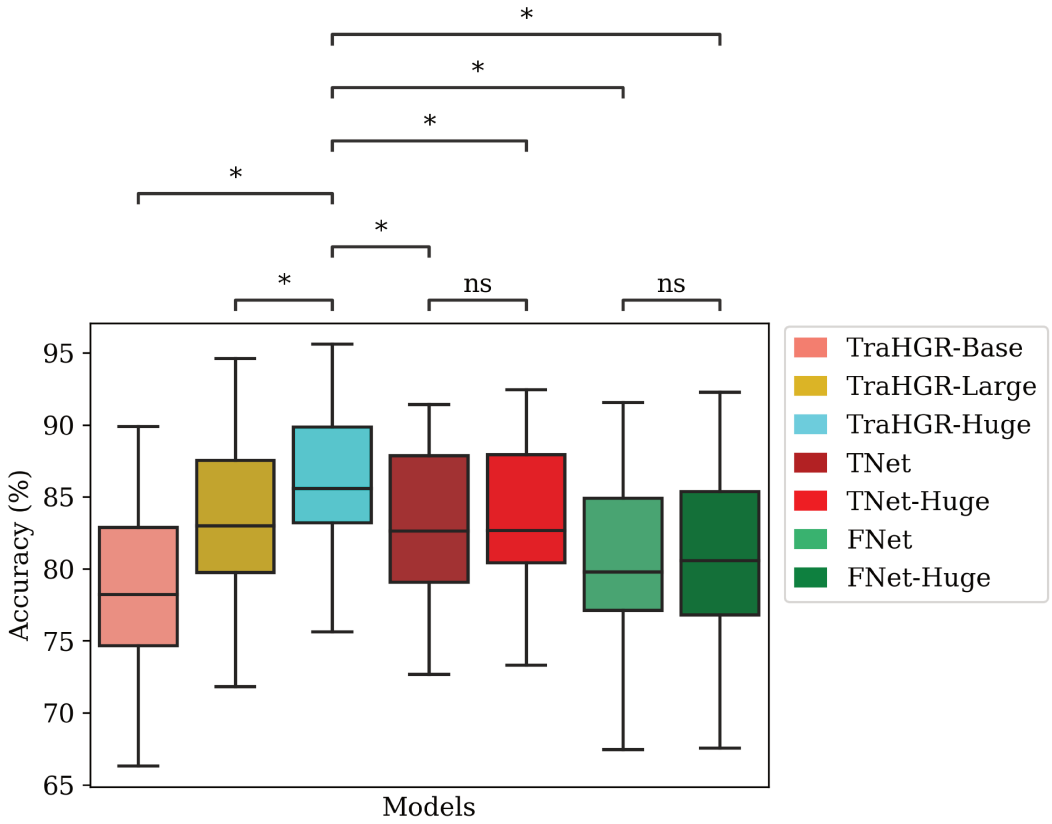

Figure 4.3: The accuracy boxplots for all TraHGR architecture variants, SNet, and FNet for all 49 gestures in Ninapro DB2 dataset. The IQR of each model is shown by a boxplot for all users. The Wilcoxon signed-rank test is used to compare the TraHGR-Huge with other architectures, and different variants of SNet and FNet. p-value is annotated by the following markers: (i) $0.05 < p\text{-value} \leq 1$ is marked as not significant (ns); (ii) $p\text{-value} \leq 0.05$ is depicted with $*$.

Each boxplot shows the Interquartile Range (IQR), which presents the performance of each model for all users into quartiles. More specifically, the upper and lower whiskers show the 75th and 25th percentiles. In a sense that, in each boxplot, the achieved accuracy for 25% of the users, i.e., 10 users, are in the range defined by the lower whisker and the other 25% of the users have accuracy in the range defined by the upper whisker. The horizontal lines at the beginning of the lower whisker and the end of the upper whisker indicate the models' minimum and maximum accuracies, respectively. Finally, the boxplot covers the range of accuracy for 50% of the users. The horizontal line in each boxplot illustrates the median performance. In a sense that the accuracy of 25% of users falls into the bottom portion of the box, while the

other 25% of users fall into the higher part of the box. As shown in Fig. 4.3, The boxplot corresponding to TraHGR-Huge compared to other counterparts is shifted up. In other words, TraHGR-Huge has improved the performance of all users. Furthermore, when comparing different TraHGR variations, it is clear that increasing the number of parameters led to an increase in accuracy due to the models' enhanced capacity to extract more generic representations. However, increasing the number of parameters does not have a significant improvement on the SNet and FNet as shown in the Fig. 4.3, "ns" stands for not significant, i.e., $0.05 < p\text{-value} \leq 1$.

For evaluating the robustness of the proposed approach, in particular, 100 MC runs are performed where at each run sensor measurements are contaminated by additive Gaussian noise based on a specific level of signal-to-noise ratio (SNR). MC simulation results (100 times and SNR $= 25\,\text{dB}$) for the proposed TraHGR-Huge is $85.68\% \pm 5.32\%$, while without MC simulation (Table 4.2) the accuracy for the same model is $86.18\% \pm 4.99\%$. The achieved accuracy shows a remarkably stable performance of the proposed model.

4.2.5 Position-Wise Cosine Similarity

As illustrated in the proposed TraHGR architecture in Fig. 4.1, each patch in the in SNet only consists of the temporal information of one sensor for the length of window size (e.g., 200ms, 150ms). As a result, the positional embeddings represent their associated sensors. Therefore, as shown in Fig. 4.4, the position-wise cosine similarity of the positional embedding vectors in the SNet captures the mutual correlation/entanglement of the sensors in the hand movements. As depicted in Fig. 4.4, the sensory information is highly correlated for the TraHGR-Base as the smallest network, when the network gets larger (left to right) and the sequence length gets longer (down to up), the network's capacity to cherry-pick the sensors to associate is increased.

On the other hand, each patch in FNet consists of both temporal and spatial information. As illustrated in Fig. 4.1, unlike the patching mechanism in SNet, there is temporal information flow from one path to another in the FNet patching mechanism

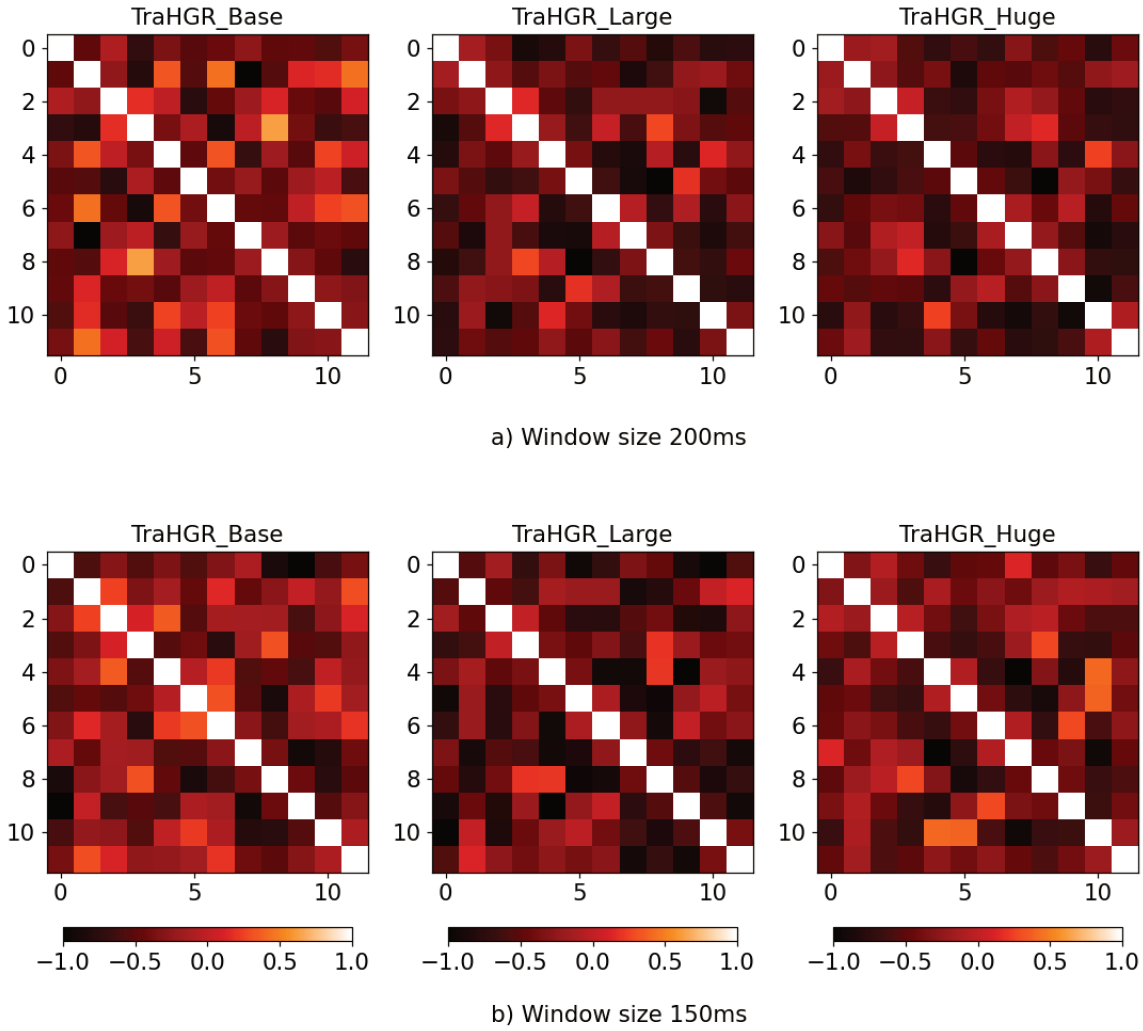

Figure 4.4: Position embedding similarities for SNet path in TraHGR-Base, TraHGR-large, and TraHGR-Huge architectures: (a) window size is 200ms, and (b) window size is 150ms. Each row in each figure represents the cosine similarity between one embedding position and all the other embeddings. The brightness of the pixels in the figures indicates more similarity.

which makes the order of the sequence of patches/information important. These sequential correlations of the patches are expected to be deduced by the FNet. The optimal similarity should result in a matrix with bright colors on the main diagonal and its neighbors. In a sense that consecutive positions are required to be more similar/brighter to reflect the importance of the sequence of patches' order. As shown in Fig. 4.5, TraHGR-Huge captures the position meanings better than

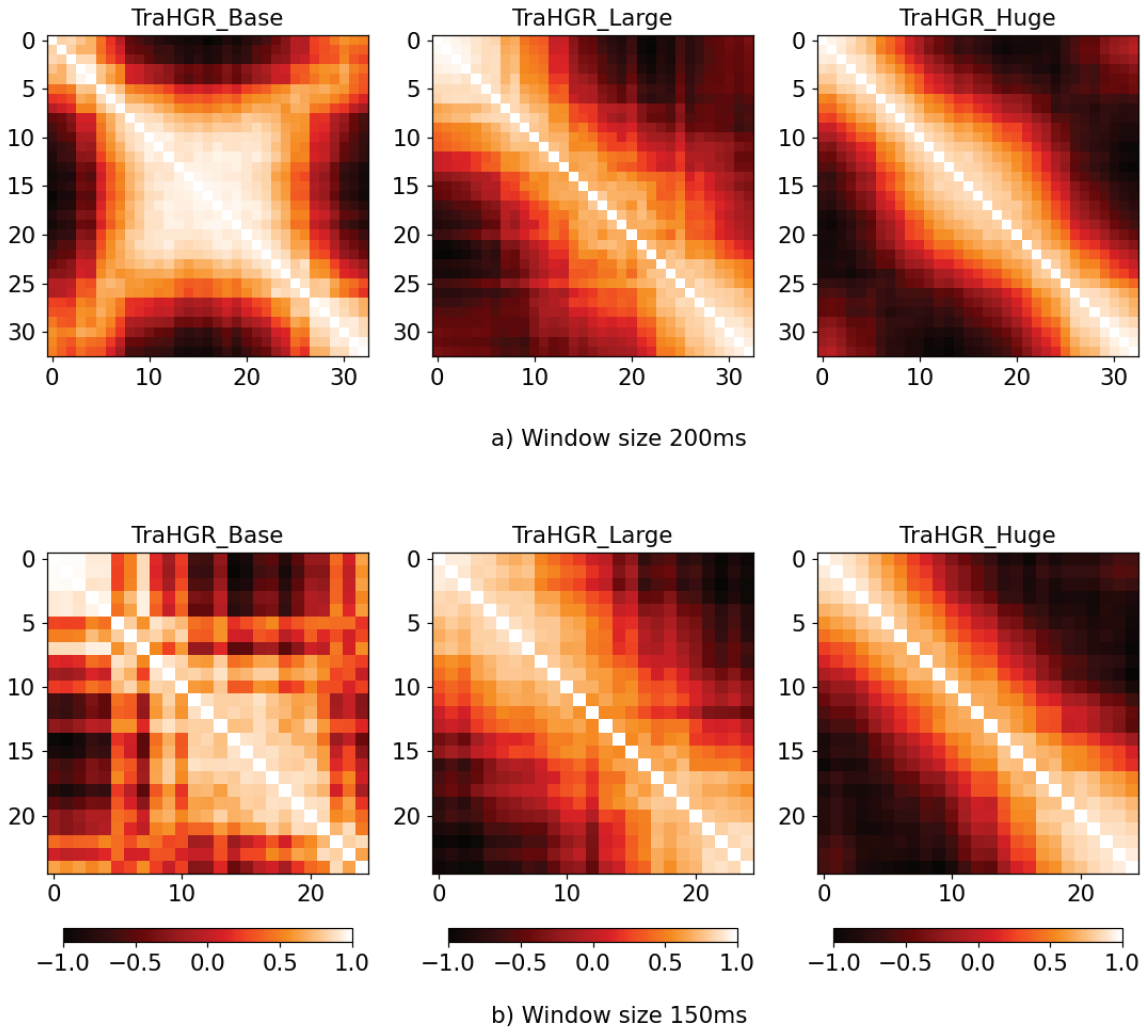

Figure 4.5: Position embedding similarities for FNet path in TraHGR-Base, TraHGR-large, and TraHGR-Huge architectures: (a) window size is 200ms, and (b) window size is 150ms. Each row in each figure represents the cosine similarity between one embedding position and all the other embeddings. The brightness of the pixels in the figures indicates more similarity.

TraHGR-large, and TraHGR-large better than TraHGR-Base for both window sizes 200ms and 150ms. As a consequence, it is possible to conclude that a more complex architecture can improve position embedding inference and includes more location data for transformer encoders. Moreover, as shown in Fig. 4.5, for longer window sizes, the sequential nature of sEMG signals can be better encoded. For instance, as shown in Fig. 4.5(b), the position embeddings for the TraHGR-Base architecture did

not adequately infer the concept of positions. As a result, it is reasonable to deduce that the window size has a direct influence on the transformer encoder's ability to infer position information.

4.2.6 Comparison with existing deep learning approaches

Table 4.5 provides a comparison between our proposed approach TraHGR-Huge and the available methodologies, which shows the superiority of our architecture over the experimental results obtained from the state-of-the-art researches [57, 58, 79, 86–88]. This comparison was evaluated based on the same settings for the DB2 (49 gestures) dataset and its sub-exercises, i.e., DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures). The Ninapro database was collected from 40 users. Each user performs 49 movements in which each movement is repeated 6 times, each time lasting for 5 seconds, followed by 3 seconds of rest. The sEMG signals were gathered using the Delsys Trigno Wireless EMG system with 12 wireless electrodes, sampled at 2 kHz. The DB2 dataset was presented in three exercises B, C, and D, which consist of different types of movements. In particular, Exercises B, C, and D consist of 17, 23, and 9 movements, respectively.

According to the recommendations in [140], the window size should be less than 300ms to meet the acceptable delay time for myoelectric control systems. Therefore, in this study, we segmented sEMG signals with three windows, i.e., 200ms, 150ms, and 100ms, to fulfill the mentioned limitation. As shown in Table 4.5, our proposed approach TraHGR-Huge achieved higher accuracy than the existing methodologies evaluated based on DB2 (49 gestures), DB2-B (17 gestures), DB2-C (23 gestures), DB2-D (9 gestures), with different time window sizes. More specifically, we compared the proposed architecture with both advanced DNNs and classical ML approaches.

For instance, Reference [79] showed the average classification accuracy obtained using all the classical methods such as SVM, RF, KNN, and LDA on the DB2 (49 gestures) dataset is 60.28%. They achieved the highest gesture recognition accuracy for RF which is 75.27%. Moreover, in Reference [86], they achieved the recognition

Table 4.5: Comparison between our methodology (TraHGR-Huge) and previous works [20, 57, 58, 79, 86, 88, 137].

Method	**Database**	**Window size**		
		200ms	**150ms**	**100ms**
CNN [57]	DB2 (49 gestures)	83.70	82.70	81.10
Attention-based Hybrid CNN-RNN [58]	DB2 (49 gestures)	82.20	-	-
CNN [88]	DB2 (49 gestures)	78.86	-	-
CNN [86]	DB2 (49 gestures)	78.71	-	-
CNN [79]	DB2 (49 gestures)	-	60.27	-
SVM [86]	DB2 (49 gestures)	77.44	-	-
RF [137]	DB2 (49 gestures)	75.27	-	-
RF [78]	DB2 (49 gestures)	72.25	-	-
TraHGR-Huge	DB2 (49 gestures)	**86.18**	**85.43**	**84.13**
CNN + Dilated LSTM [20]	DB2-B (17 gestures)	79.00	-	-
CNN [86]	DB2-B (17 gestures)	82.22	-	-
CNN [88]	DB2-B (17 gestures)	83.79	-	-
SVM [86]	DB2-B (17 gestures)	81.07	-	-
TraHGR-Huge	DB2-B (17 gestures)	**88.91**	**88.14**	-
CNN [86]	DB2-C (23 gestures)	72.62	-	-
SVM [86]	DB2-C (23 gestures)	71.08	-	-
TraHGR-Huge	DB2-C (23 gestures)	**81.44**	**79.99**	-
CNN [86]	DB2-D (9 gestures)	89.54	-	-
SVM [86]	DB2-D (9 gestures)	88.56	-	-
TraHGR-Huge	DB2-D (9 gestures)	**93.84**	**93.58**	-

accuracy of 77.44% using SVM over all the movements. In addition, the recognition accuracy of 72.25% is reported in Reference [78] for the RF classifier. For DNN

architectures, on the other hand, the best detection accuracy is reported in Reference [57] using CNN, which is 83.70%. As shown in Table 4.5, for a window size of 200ms, our proposed architecture achieved 86.18% classification accuracy which is 2.48% higher than the state-of-the-art DNN approach and 8.74% higher than state-of-the-art classical ML method. Moreover, it can be observed that for other window sizes, the classification accuracy of our proposed approach achieved better gesture recognition performances than its counterparts. For example, when the window size is set to 100ms, our proposed approach TraHGR-Huge was able to achieve gesture recognition accuracy of 84.13%, but using the proposed approach of [57], the accuracy of 81.1% is achieved. It should be noted that the accuracy of 84.13% obtained by TraHGR-Huge with a window size of 100ms is still higher than the case where the window size in Reference [57] has doubled, i.e., 200ms. We also evaluated and compared our proposed method for DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures) with the previous studies [86, 88], which demonstrates the superiority of our hybrid Transformer-based framework.

Authors in [20] introduced a hybrid CNN-LSTM model achieving 79% average accuracy on the window size of 200ms as shown in Table 4.5. They reduced the number of parameters in the proposed model using dilated LSTM, resulting in $1,102,801$ parameters for 17 gesture classifications. However, as shown in Table 4.5, TraHGR-Huge outperforms [20] for 17 gesture classification with less number of parameters $(832,659)$.

4.2.7 Transfer Learning Impact on TraHGR Performance

In this experiment, The 5^{th} Ninapro database [78], referred to as the DB5, is used for the ease of comparison with Ref. [114]. The DB5 dataset is recorded with two Thalmic Myo-armbands recording muscular activity at a rate of 200Hz. The DB5 dataset, in particular, consists of signals collected from 10 users executing 52 actions/movements. Each movement in the DB5 dataset is repeated 6 times, each lasting for 5 seconds followed by 3 seconds of rest. The DB5 dataset is provided in three sets of exercises [78]. In this work, we only consider data collected by the lower armband in DB5 in the second exercise of the DB5 to follow the same criteria in [114] and also have a fair comparison. Moreover, out of 6 movement repetition for each target user, following

Table 4.6: The average accuracy of hand gesture recognition across all subjects in the second experiment of Ninapro DB5 dataset on the window size of 260ms. The average accuracy is reported on 5 and 6 repetitions for all models in Ninapro DB5 dataset.

Repetitions (Rep.) Used for Training/Fine-tuning	Accuracy ± STD			
	ConvNet [114]	**ConvNet+TL [114]**	**TraHGR-Huge**	**TraHGR-Huge**+TL
Rep. 1, 2, 3, 4	66.30 ± 3.77	68.98 ± 4.09	71.21 ± 1.99	74.63 ± 2.52
Rep. 1, 2, 3	61.91 ± 3.94	65.16 ± 4.46	66.82 ± 2.07	69.01 ± 2.77
Rep. 1, 2	55.65 ± 4.38	60.12 ± 4.79	58.68 ± 2.81	62.07 ± 2.70
Rep. 1	46.06 ± 6.09	49.41 ± 5.82	51.33 ± 2.93	53.42 ± 3.31

[114], the first four repetitions are used to fine-tune the pre-trained network, and the last two repetitions serve as the test set.

Table 4.6 shows the average accuracy on the second experiment of the Ninapro DB5 dataset. As shown in Table 4.6, the TraHGR-Huge outperforms ConvNet [114] whether the training process of the network is involved with the TL stage or solely trained for each user. In Table 4.6, the networks without TL training stage are independently trained for each user. However, to integrate TL techniques into the training process, we conducted a typical TL method to utilize the knowledge learned in the source domain to promote the learning process in a target domain. Specifically, given a user as the target, in the first stage, the training sets of the remaining nine participants/users are employed to pre-train the TraHGR-Huge network. Then, to fine-tune the pre-trained network, the weights of the SNet and FNet in TraHGR-Huge are maintained intact by freezing them, and the non-frozen parts of the network are updated using one, two, three, or four repetitions of the target data (see Table 4.6). As shown in Table 4.6, using transfer learning as a domain adaptation approach is conducted to performance improvement of both the TraHGR-Huge and ConvNet models compared to their corresponding user-specific trained models. When comparing our transformer-based model to ConvNet with convolutional structure, we can infer that TraHGR-Huge achieves higher accuracies, demonstrating the proposed model's ability to extract more useful representations from raw sEMG data.

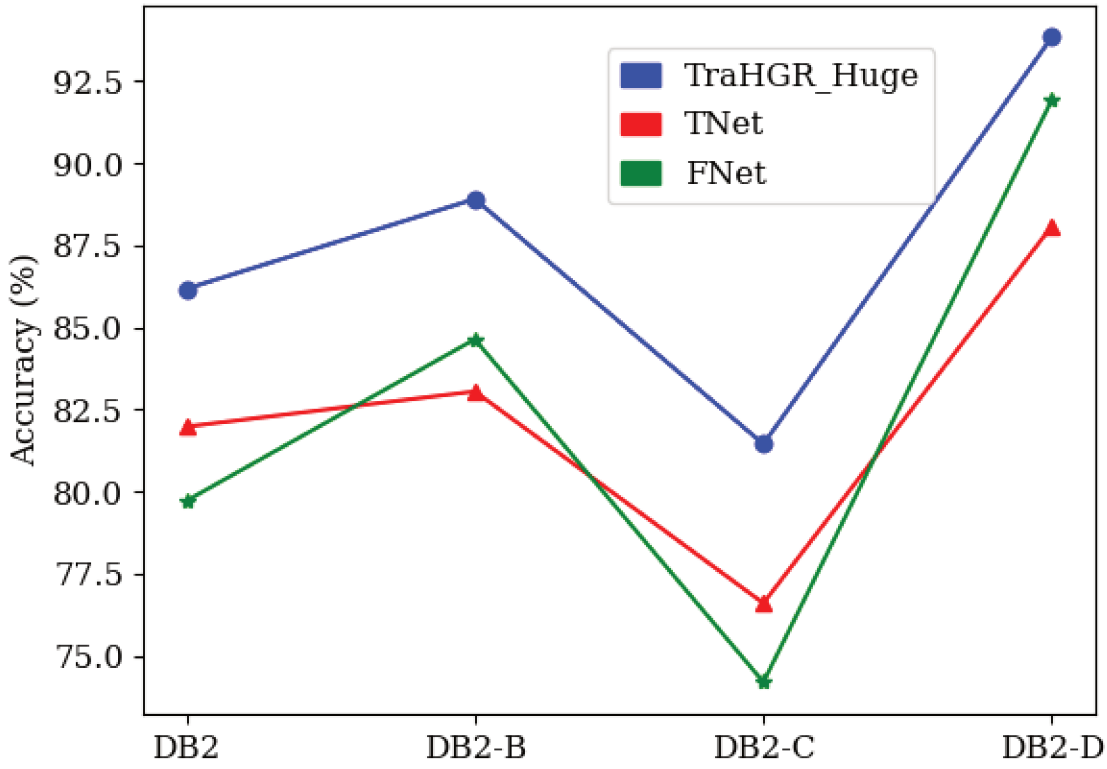

Figure 4.6: The accuracy for TraHGR-Huge, SNet, and FNet when they are trained simultaneously for DB2 (49 gestures) and its sub-exercises, DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures).

4.2.8 Ablation Study

For the proposed hybrid architectures, i.e, TraHGR-Huge, TraHGR-large, and TraHGR-Base, the classification accuracy is calculated using the prediction values y obtained from Eq. (11). To show that our proposed architecture based on a developed hybrid strategy has great potential for improving gesture recognition accuracy, we also calculated the other two accuracies, i.e., y_{SNet} or y_{FNet}, based on the Eq. (10). More specifically, we trained the hybrid architectures by computing the loss function in Eq. (12). However, output y is used to calculate the accuracy of the reported results in the tables. Here, in Fig. 4.6, it is shown that the accuracy obtained using the y is better than those calculated using the y_{SNet} or y_{FNet} for DB2 (49 gestures) and its sub-exercises. In particular, from Fig. 4.6, it can be observed that the hybrid architecture takes advantage of two parallel paths and improved the recognition

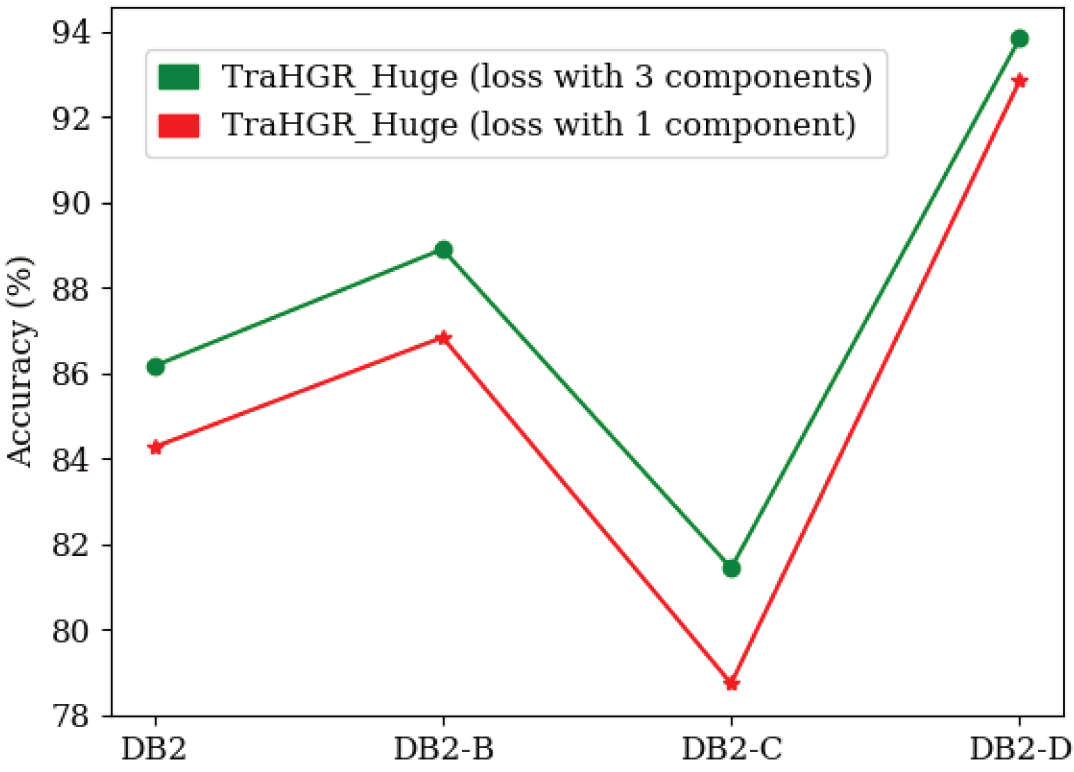

Figure 4.7: Results of the ablation study on loss functions with TraHGR-Huge model which is trained by Eq. 12 (green) and Eq. 13 (red) evaluated on DB2 (49 gestures), DB2-B (17 gestures), DB2-C (23 gestures), and DB2-D (9 gestures).

accuracy.

Evaluation of Multiple Loss Functions: As described in sub-section 4.2.1, our proposed hybrid architectures' parameters are learned by optimizing the loss function $\mathcal{L}$, which consists of three components. To demonstrate the advantage of training our proposed hybrid architecture with loss function $\mathcal{L}$ defined in Eq. (12), we evaluated performance of TraHGR-Huge when the loss function $\mathcal{L}$ has only one component as follows:

$$\mathcal{L} = \mathcal{L}_{\text{TraHGR}}. \tag{13}$$

Fig. 4.7 shows the performance of TraHGR-Huge in DB2 (49 gestures) and its sub-exercises for two different loss functions. We can see that training TraHGR-Huge with a loss function with three components (Eq. (12)) improves the results compared

to the case where the loss function has only one component (Eq. (13)).

4.3 Conclusion

In this chapter, we proposed a hybrid architecture based on the Transformers for the task of hand gesture recognition. We have shown that the proposed hybrid architecture could augment the power of model discrimination resulting in a significant performance improvement in the task at hand. Moreover, we investigated the ability of Transformers for sEMG-based hand gesture recognition as such they revolutionized other fields and applications such as NLP, CV, and speech recognition. In this regard, we compared TraHGR results with traditional ML approaches and DNN-based techniques and demonstrated the outstanding performance of the proposed architecture. However, one major drawback of transformers, especially for wearable devices, is their high computational requirements. This high computational requirement can be a significant challenge for embedding the models in wearable devices, which typically have limited processing power and battery life. Running a transformer model on a wearable device can quickly drain its battery and cause performance issues. To address this issue, we have explored new architecture to reduce the computational requirements of transformers, which is provided in the next chapter.

Chapter 5

Light-weight CNN-Attention based Architecture for sEMG-based Hand Gesture Recognition

Despite extensive research in this area and the fact that academic researchers achieve high classification accuracy in laboratory conditions, there is still a gap between academic research in sEMG pattern recognition and commercialized solutions [18]. For instance, one of the main obstacles in current prosthesis devices is the lack of feedback provided to the user regarding the prosthesis's position or the forces being applied. This can make the control process difficult and less precise for the user, leading to less natural and less efficient interaction with the device. To develop a user-friendly and reliable prosthesis control, providing feedback is crucial [18, 141, 142]. Moreover, there are challenges related to the wearability and portability of the sEMG-based systems, as well as the ease of use, and the robustness against the variations in muscle activation patterns, which may affect the performance of the systems [143]. Academic researches often focus on developing advanced and sophisticated algorithms to improve the performance of sEMG-based prosthesis control, but these methods may be too complex or too expensive to be practical for industrial use from the time and computation perspective [19, 63–65]. All these factors contribute to the existing gap

between academic research in sEMG pattern recognition and commercialized solutions, and further research and development are needed to overcome these limitations and improve the performance and usability of sEMG-based systems for practical applications. In this context, the primary goal of this study is to reduce the gap by developing DNN-based models that not only have high recognition accuracy but also have minimal processing complexity, allowing them to be embedded in low-power devices such as wearable controllers [19, 20]. Furthermore, the designed DNN-based models should be based on the minimum number of electrodes while estimating the desired gestures within an acceptable delay time [18, 55]. Consequently, we develop the novel **Hierarchical Depth-wise Convolution along with the Attention Mechanism (HDCAM)** model for HGR based on sparse sEMG signals to fill this gap by meeting criteria such as improving the accuracy and reducing the number of parameters. The HDCAM is developed based on the Ninapro [87, 144] database, which is one of the most well-known sparse multi-channel sEMG benchmark datasets.

Although recent academic researchers are improving the performance by using Recurrent Neural Networks (RNNs) or hybrid CNN-RNN architectures [20,61,90,92–94], the sequence modeling with recurrent-based architectures has several drawbacks such as consuming high memory, lack of parallelism, and lack of stable gradient during the training [65, 96]. It is demonstrated [95] that sequence modeling using RNN-based models does not always outperform CNN-based designs. Specifically, CNN architectures have several advantages over RNNs such as lower memory requirements and faster training if designed properly [95]. Therefore, in the recent literature [65,96–98], the authors took advantage of 1-D Convolutions developed based on the dilated causal convolutions, where the sequence of sEMG signals can be processed as a whole with lower memory requirement during the training compared to RNNs. Convolution operation in CNNs, however, has two main limitations, i.e., (i) it has a local receptive field, which makes it incapable of modeling global context, and; (ii) their learned weights remain stationary at inference time, therefore, they cannot adapt to changes in input. Attention mechanism [14] can mitigate both of these problems. Consequently, the authors in the recent research papers [19, 63, 99–101] used the attention mechanism combined with CNNs and/or RNNs to improve the performance of sEMG-based HGR. The attention mechanism's major disadvantage is that it is often computationally intensive, necessitating a carefully engineered design to ensure computational

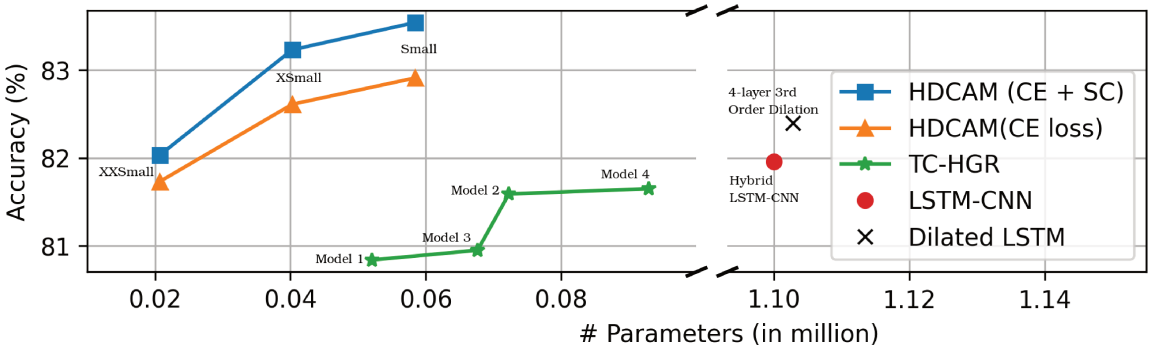

Figure 5.1: Comparing different variants of the proposed HDCAM model with SOTA designs for an input window size of 300 ms. The x-axis shows the number of parameters and the y-axis displays the classification accuracy on the Ninapro DB2 dataset. HDCAM shows a better compute versus accuracy trade-off compared to recent approaches. The square-blue plot shows HDCAM trained with Cross Entropy (CE) and Supervised Contrastive (SC) losses, whereas all other models are trained with only CE loss.

viability, particularly for low-power devices. Therefore, the combination of CNNs and Transformers offers a promising solution by harnessing the strengths of both architectures and addressing their respective limitations, resulting in an effective and efficient model for sEMG-based hand gesture recognition.

In this chapter, we develop HDCAM architecture by effectively combining the complementary advantages of CNNs and the attention mechanisms. Our proposed architecture shows a favorable improvement in terms of parameter reduction and accuracy compared to the state-of-the-art (SOTA) methods for sparse multichannel sEMG-based hand gesture recognition (see Fig. 5.1). The contributions of the HDCAM architecture can be summarized as follows:

- Efficiently combining advantages of Attention- and CNN-based models and reducing the number of parameters (i.e., computational burden).
- Efficiently extracting local and global representations of the sEMG sequence by coupling convolution and attention-based encoders.
- Integration of Depth-wise convolution (*DwConv*) a hierarchical structure in the proposed Hierarchical Depth-wise Convolution (*HD*Conv) encoder, which not only extracts a multi-scale local representation but also increases the receptive field in a single block.

The small version of the proposed HDCAM with $58,441$ parameters achieves 83.54% top-1 classification accuracy on Ninapro DB2 dataset with $18.87\times$ less number of parameters compared to the previous approach [20].

5.1 The Proposed HDCAM Architecture

The primary objective of this study is to build a lightweight hybrid architecture that successfully combines advantages of Attention- and CNN-based models for low-powered devices. In what follows, the HDCAM architecture is explained in detail, and then the training objectives are described.

5.1.1 Overview of HDCAM Architecture

In the proposed framework, a sliding window strategy with the window size of $\boldsymbol{W} \in \{150, 200, 250, 300\, ms\}$ is adapted to use the multi-variate temporal sEMG information, resulting in dataset $\boldsymbol{D} = \{(\boldsymbol{X}_i, \mathrm{y}_i)\}_{i=1}^{N}$. More specifically, $\mathrm{y}_i \in \mathbb{R}$ is the label assigned to the i^{th} segmented sequence $X_i \in \mathbb{R}^{L\times C}$. Here, L is the length of the segmented sequential input corresponding to the number of samples obtained at a frequency of 2 kHz for a window of size $\boldsymbol{W}$, and $\boldsymbol{C}$ denotes the number of channels in the input segment corresponding to the number of input features/sensors. As illustrated in Fig. 5.2, the HDCAM framework has a hybrid design based on CNN and the "Multi-Head Self-Attention (MHA) mechanism" to reap the advantages of both methods for designing a lightweight architecture for low-power devices.

As shown in Fig 5.2(a), the overall HDCAM architecture consists of four different stages, the first three for multi-scale feature extraction and the last one for classification. HDCAM is made up of two primary components, namely "Hierarchical Depth-wise Convolution (*HD*Conv)" encoder and "Multi-Head Self-Attention (*MHS*Atten)" encoder, where the former and latter aim to model the local and global information in the sequential input, respectively. Formally, for a given segmented sequential input $X_i \in \mathbb{R}^{L\times C}$, HDCAM begins with the Stem layer. More specifically, the Stem layer serves as a patching mechanism for the input X_i which applies a 10×1

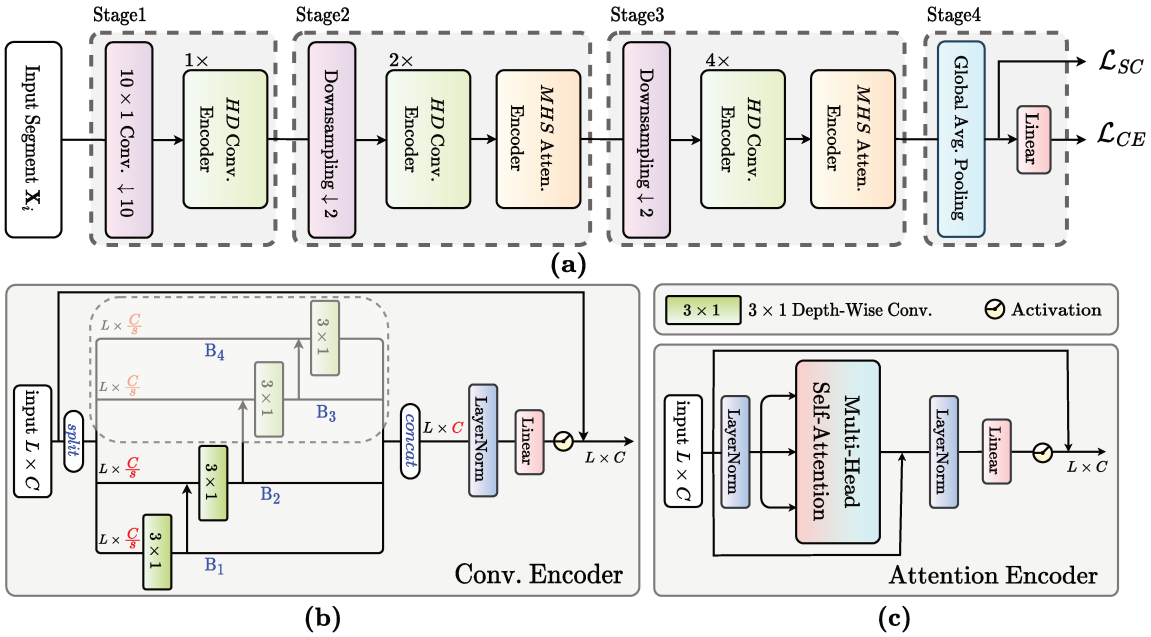

Figure 5.2: The proposed architecture: **(a)** The overall architecture of proposed HDCAM model. At stage 4, the output representations of the Global Average Pooling (GAP) layer are passed to Supervised Contrastive Loss ($\mathcal{L}_{\text{SC}}$), and the output logits of the Linear layer are used in Cross Entropy Loss ($\mathcal{L}_{\text{CE}}$). **(b)** The *HD*Conv Encoder uses Hierarchical Depth-wise Convolution for multi-scale temporal feature mixing followed by a point-wise convolution, i.e. Linear layer, for channel mixing. To expand the receptive field in the deeper layers, the number of active branches (B_i) is increased from Stage 1 to Stage 3. **(c)** The design of the *MHS*Atten Encoder is illustrated, which consists of a Multi-Head Self-Attention (MHA) mechanism to encode the global representation of the input feature maps.

strided convolution with the stride of size 10 followed by a Layer Normalization (LN) to the input. Patching mechanism helps reduce memory and computation requirements in downstream layers resulting in $L/10 \times C_1$ feature maps. Afterward, local features are extracted using a *HD*Conv encoder. Further processing of the feature maps takes place in the second and third stages, which follow almost the same architectural structure. Both of which start with the downsampling layer followed by consecutive *HD*Conv encoders for *local* feature extraction and end with *MHS*Atten block to encode the *global* representations of the input. The downsampling layer consists of an LN followed by a 2×1 strided convolution with stride of size 2, which reduces the sequential feature maps length by half and increases the number channels, resulting $L/20 \times C_2$ and $L/40 \times C_3$ dimensional features for second and third stages,

respectively. In the final stage, a Global Average Pooling (GAP) operation is used to reduce the feature maps' dimension followed by a Linear layer for classification. When Supervised Contrastive (SC) loss is adopted to training model the output of the GAP layer, denoted by $\boldsymbol{z}_i \in \mathbb{R}^{C_3}$, is used as the input $\boldsymbol{X}_i$ representation, further discussed in section 5.1.4. Here, C_k refers to number of channels in k^{th} stage, for $k \in \{1, 2, 3\}$.

5.1.2 *HD*Conv Encoder

As shown in Fig. 5.2(b), the proposed *HD*Conv block combines depth-wise convolution with a hierarchical structure to extract local features at multi-scales. The proposed multi-scale feature extractor is inspired by the Res2Net [145] module, which combines features with different resolutions. Different from the Res2Net module, we omitted the first point-wise convolution layer and added a 3×1 depth-wise convolution to the first branch. Also, the number of active branches in the hierarchical convolutional structure is dynamic and varies depending on the stage. In *HD*Conv module, input feature maps of shape $L \times C$ is evenly splitted into s subsets/scales, denoted by $\boldsymbol{x}_i$ of shape $L \times C/s$, where $i \in \{1, 2, \ldots, s\}$. Then, 3×1 depth-wise convolution, denoted by $DwConv_i$, is applied on each subset $\boldsymbol{x}_i$ after combining with the previous branch output features, denoted by $\boldsymbol{y}_{i-1}$. Generally, we can write the output features of each branch $\boldsymbol{y}_i$ as follows

$$\boldsymbol{y}_i = \begin{cases} DwConv_i(\boldsymbol{x}_i) & i = 1 \\ DwConv_i(\boldsymbol{x}_i + \boldsymbol{y}_{i-1}) & 2 \leq i \leq s \end{cases} \tag{14}$$

As shown in Fig 5.2(b) and Eq. (14), the hierarchical structure allows each depth-wise convolution $DwConv_i$ receive the information from all previous splits, $\{\boldsymbol{x}_j, j \leq i\}$. The output feature maps of all branches are concatenated and passed through an LN followed by point-wise convolution to enrich the multi-scale local representation, and finally, Gaussian Error Linear Unit (GELU) activation is used for adding non-linearity to the model. For information flow through the network hierarchy, residual connection is used in *HD*Conv encoder. The *HD*Conv encoder can be represented as

follows

$$X_{out} = X_{in} + Linear_{GELU}(LN(HDwConv(X_{in}))) \tag{15}$$

where X_{in} and X_{out} are the *HD*Conv input and output feature maps, both of shape $L \times C$, $Linear_{GELU}$ is point-wise convolution followed by GELU non-linearity LN is Layer Normalization, and $HDwConv$ is hierarchical depth-wise convolution operation. Finally, it worth to note that in order to expand the receptive field in the deeper layers of the network, the number of active branches (B_i in Fig 5.2(b)) in *HD*Conv Encoder is increased from Stage 1 to Stage 3 for the proposed model.

5.1.3 *MHS*Atten Encoder

In [14], the authors showed that the attention mechanism allows a model to present global information in a given input sequence. Furthermore, attention-based architectures [19,63,99–101] have shown promising performance in the context of sEMG-based HGR by extracting particular bits of information from the sequential nature of the sEMG signals. However, most of these models are still heavy-weight to be used in resource-constrained devices. Hence, in the proposed HDCAM architecture, we designed a hybrid architecture that combines convolutions and attention mechanism advantages. Specifically, due to spatial inductive biases in convolution operation, the CNN-based encoder (*HD*Conv) assists our hybrid model to learn local representations with fewer parameters than solely attention-based models. However, to effectively learn global representations, we also used an attention-based encoder (*MHS*Atten). Since computation in the MHA has quadratic relation to input size, we only used the *MHS*Atten encoder in the second and third stages of the HDCAM to efficiently encode the global representation, where the length of the sequential feature maps are 1/20 and 1/40 of the original input of the network, respectively. The *MHS*Atten encoder can be represented as follows

$$X_{out} = Linear_{GELU}(LN(X_{in} + MHA(LN(X_{in})))) + X_{in} \tag{16}$$

where X_{in} and X_{out} are the *MHS*Atten input and output feature maps, both of shape $L \times C$, $Linear_{GELU}$ is point-wise convolution followed by GELU non-linearity LN is Layer Normalization, and *MHA* is Multi-Head Self-Attention mechanism. In *MHA*,

the input feature maps X_{in} of shape $L \times C$ are passed through a Linear projection to create Queries Q, i.e., a matrix with the same shape as the input feature maps. Then, Queries Q is evenly splitted into h subsets, denoted by $\boldsymbol{q}_i$ of shape $L \times C/h$, where $i \in \{1, 2, \ldots, h\}$ and h is number of the heads. In parallel, the same approach has been applied to construct Keys and Values subsets, i.e., $\boldsymbol{k}_i$ and $\boldsymbol{v}_i$. Finally, on each head, the attention block measures the pairwise similarity of each $\boldsymbol{q}_i$ and all $\boldsymbol{k}_h$ to assign a weight to each $\boldsymbol{v}_h$. The entire operation is

$$A_h = Softmax(\frac{\boldsymbol{q}_h \boldsymbol{k}_h^T}{\sqrt{d}})\boldsymbol{v}_h, \tag{17}$$

where d=C/h denotes the dimension of $\boldsymbol{k}_h$ and $\boldsymbol{q}_h$ subsets. Then concatenation of attention feature maps of all heads is projected to get the final attention maps of the MHA mechanism, i.e., $MHA(X_{in})$=$Linear(Concat(A_1, A_2, \ldots, A_h))$. This completes the description of the proposed HDCAM architecture, next, we present the training objectives of the proposed model.

5.1.4 Training Objectives

For model training, we employ a hybrid loss that consists of two-fold: **(i)** Cross Entropy (CE) loss which focuses on identifying the helpful features to perform the classification objective, and **(ii)** Supervised Contrastive (SC) loss which assists to learn more robust and generic features by minimizing the ratio of intra-class to inter-class similarity.

Cross Entropy (CE) Loss: To train a classifier by CE loss, the predicted probability of each sample X_i is compared to the actual expected value y_i, and a loss is calculated to penalize model weights $\boldsymbol{\theta}$ based on how far the prediction is from the actual expected value. Given a training batch $\mathcal{B} = \{(\boldsymbol{X}_i, \mathrm{y}_i)\}_{i=1}^{|\mathcal{B}|}$, the CE is formulated as

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} \mathrm{y}_i \log \mathrm{p}_\theta(\mathrm{y}_i|\boldsymbol{X}_i) \tag{18}$$

where $\mathrm{p}_{\boldsymbol{\theta}}(\mathrm{y}_i|\boldsymbol{X}_i)$ is predicted class probability by the classifier. Although CE loss is the most commonly used objective function to adjust weights of deep classification

models, it has several known issues, such as the lack of robustness to noisy labels [146] and the possibility of inefficient margins [147]. Hence to mitigate these limitations inspired by recent works [148, 149], we added SC loss [150] as a regularization term to the conventional CE objective function.

Supervised Contrastive (SC) Loss: The SC loss is intended to increase the similarity between features resulting from positive sets while simultaneously driving away features of the negative sets. Following [150], to form positive and negative sets, we leverage label information. For instance, given a training batch $\mathcal{B} = \{(\boldsymbol{X}_i, \mathrm{y}_i)\}_{i=1}^{|\mathcal{B}|}$, for a sample $\boldsymbol{X}_i$ (i.e. anchor), the anchor set is all samples in the batch except $\boldsymbol{X}_i$, and the positive set is composed of the samples that are in the same class as $\boldsymbol{X}_i$, i.e., samples with the label y_i. Accordingly, the negative set is defined by the samples that are in the anchor set but not in the positive set. To compute the SC loss, we first embed inputs in lower dimension space to get the representations, denoted by $\boldsymbol{z}_\star$. Then, the SC loss can be computed by

$$\mathcal{L}_{\mathrm{SC}} = -\frac{1}{|\mathcal{B}|} \sum_{\forall i \in \mathcal{I}} \log \frac{\sum_{\forall p \in \mathcal{P}(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_p)}{\sum_{\forall a \in \mathcal{A}(i)} \exp(\boldsymbol{z}_i \cdot \boldsymbol{z}_a)} \tag{19}$$

where $\cdot$ denotes the dot product operation, $\mathcal{I} = \{1, 2, \ldots, |\mathcal{B}|\}$ indicates indices of all samples in the batch, $i \in \mathcal{I}$ is the index of the anchor, $\mathcal{A}(i) \equiv \mathcal{I} \backslash \{i\}$ represents the indices of all batch samples but the anchor, and $\mathcal{P}(i) = \{p \in \mathcal{A}(i) : \mathrm{y}_i = \mathrm{y}_p\}$ is positive set composed of the indices of samples sharing the same class. In our framework, the output feature maps of the GAP layer in stage 4 are used as the representations $\boldsymbol{z}_\star$ of the inputs (see Fig. 5.2(a)).

Hybrid Loss: According to [150], using the conventional SC loss requires two distinct training stages for a classification problem: first, learning the representations using SC loss, and second, training classifier on top of the learned representations with the CE loss. However, SC loss generally demands a relatively high batch size in order to get acceptable and stable performance, while this is not the case for CE loss. Therefore, to take the advantages of both CE and SC losses, we jointly trained the HDCAM with the weighted sum of them as follow

$$\mathcal{L}_{\mathrm{H}} = \mathcal{L}_{\mathrm{CE}} + \lambda \mathcal{L}_{\mathrm{SC}} \tag{20}$$

where λ is a weighting coefficient for balancing the losses. Using weighted sum of losses, we can have end-to-end training and learn more general and robust representation due to the minimization of the intra- to inter-class similarity ratio.

5.2 Experiments and Results

The proposed HDCAM is trained and tested using the second Ninapro dataset [87] referred to as the DB2, which is the most commonly used sparse sEMG benchmark. For a fair comparison, as well as following the recommendations of the database [87] and previous literature [19, 20, 94, 99], we considered two repetitions (i.e., 2 and 5) for testing and the remaining repetitions for training. The DB2 dataset is presented in three sets of exercises (B, C, and D). Following [19, 20, 94], the focus is on Exercise B which consists of 17 hand movements.

5.2.1 Results and Discussions

In this section, a comprehensive set of experiments is conducted to evaluate the performance of the proposed HDCAM architecture. Table 5.1 represents the sequence of the *HD*Conv and *MHS*Atten encoders along with design information of the extra-extra small (XXSmall), extra-small (XSmall), and small (Small) versions of the model. As shown in Table 5.1, the type, number, and sequence of the component blocks in the overall model architecture (illustrated in Fig. 5.2) are maintained across all HDCAM architecture variants. The differentiation between the XXSmall, XSmall, and Small models lies in the number of output channels present in each stage. Since the number of active branches (s) and attention heads (h) in the *HD*Conv and *MHS*Atten encoders is proportional to the number of output channels of the corresponding stage, we maintained the fundamental rule for all model variants, which requires having at least eight channels per-head/per-branch. Additionally, the maximum allowed number of heads/branches is set to four. All models were trained using the Adam optimizer at a learning rate of 10^{-4}. We trained HDCAM with only using CE loss (Eq. 18), and also with hybrid loss (Eq.20) in which we empirically set to $\lambda = 0.25$. Furthermore,

Table 5.1: HDCAM Architecture variants. Description of the models' layers with respect to kernel size, and output channels, repeated n times. We use a hierarchical structure in *HD*Conv Encoder to extract multi-scale local features. Also, *MHS*Atten Encoder is used to extract global representations of the feature maps.

	Layer (n)	#Layers	Kernel Size	Output Channels		
				XXSmall	**XSmall**	**Small**
Stage1	Stem	1	10×1	16	24	24
	*HD*Conv Encoder	1	3×1	$16(s = 2)$	$24(s = 3)$	$24(s = 3)$
Stage2	Downsampling	1	2×1	24	32	32
	*HD*Conv Encoder	2	3×1	$24(s = 3)$	$32(s = 4)$	$32(s = 4)$
	*MHS*Atten Encoder	1	–	$24(h = 3)$	$32(h = 4)$	$32(h = 4)$
Stage3	Downsampling	1	2×1	32	48	64
	*HD*Conv Encoder	4	3×1	$32(s = 4)$	$48(s = 4)$	$64(s = 4)$
	*MHS*Atten Encoder	1	–	$32(h = 4)$	$48(h = 4)$	$64(h = 4)$
	Global Avg. Pooling	1	–	32	48	64
	Linear	1	1	17	17	17
	Model Parameters			$20,689$	$40,281$	$58,441$

during training, the number of samples for each class in a batch is set to 32, leading to a balanced batch of size 544 samples. In the following sections, we conducted several experiments to evaluate our proposed HDCAM model. It is worth mentioning that in certain experiments, models were exclusively trained using CE loss to exclude the influence of the SC loss on the outcomes.

Impact of Contrastive Loss

Table 5.2 illustrates the average recognition accuracy of different variants of HDCAM over all subjects with and without SC loss function involvement in model training. It can be observed that the performance of all variants of HDCAM for all window sizes is improved when SC loss was involved in training. For instance, the best performance is archived for Small model with a window size of 300 ms in Table 5.2 which is 82.91%,

Table 5.2: Accuracy of HDCAM variants trained with hybrid loss (λ=0.25) and only CE loss over different window sizes ($\boldsymbol{W}$).

	Model ID	**XXSmall**	**XSmall**	**Small**	**XXSmall**	**XSmall**	**Small**
	Loss	$\mathcal{L}_H = \mathcal{L}_{CE} + \lambda * \mathcal{L}_{SC}$			$\mathcal{L}_{CE}$		
$\boldsymbol{W}$= 150 ms	Accuracy (%)	80.82	82.01	82.44	80.53	81.51	82.21
	STD (%)	6.6	6.2	6.3	6.9	6.6	6.7
$\boldsymbol{W}$= 200 ms	Accuracy (%)	81.34	82.66	82.86	81.10	81.77	82.28
	STD (%)	6.7	6.7	6.5	6.8	6.8	6.6
$\boldsymbol{W}$= 250 ms	Accuracy (%)	81.73	82.82	83.13	81.26	82.17	82.57
	STD (%)	6.8	6.6	6.6	6.8	6.7	6.6
$\boldsymbol{W}$= 300 ms	Accuracy (%)	82.03	83.23	83.54	81.73	82.61	82.91
	STD (%)	6.6	6.8	6.3	6.7	6.6	6.5

while this value increased by 0.63% when SC is used along with CE loss. These performance improvements demonstrate the usefulness of the SC loss in improving the quality of the learned representation.

The Model's Dimension

This experiment analyzes the recognition accuracy of the HDCAM by varying the number of channels in each stage, yielding XXSmall, XSmall, and Small models. In this regard, Table 5.2 shows the results for all variants of the proposed architecture for different window sizes. For the same arrangement of component layers, it can be seen from Tables 5.1 and 5.2 that the accuracy of the model is improved by increasing the dimensions of the stages regardless of training with hybrid loss or sole CE loss. More specifically, the dimension of the stage 3 is the only difference between the XSmall and Small architectures, resulting in more informative high-level features in the Small model, which leads to better performance. Comparing XXSmall versus two other variants, the dimension of all stages has reduced leading to lower performance. From Table 5.1 and 5.2, it can be observed that there is a trade-off between the complexity of the model and the accuracy.

The Effect of Window Size

As shown in Table 5.2, sliding window strategy with window of size $\boldsymbol{W} \in \{150, 200, 250, 300\, ms\}$ is adapted to evaluate the performance of the HDCAM. It is worth mentioning that W is required to be under 300 ms to have a real-time response in peripheral human machine intelligence systems [55]. Comparing outcomes in each column of Table 5.2 shows that increasing window size (W) led to better performance for all model variants for both losses. According to this observation, the proposed HDCAM architecture is capable of extracting/utilizing information from longer sequences of inputs. For instance, for XXSmall, XSmall, and Small architectures, increasing W from 150 ms to 300 ms resulted in accuracy improvements of 1.21%, 1.22%, and 1.1% when trained with hybrid loss, respectively, These values are changed to 1.2%, 1.1%, and 0.7% when models are trained withe sole CE loss. Although the number of parameters for a specific version of the model does not change for different W, larger W leads to longer sequential feature maps in the second and third stages, which leads to more memory requirements in the attention mechanism. We would like to emphasize that our proposed hybrid architecture is still far superior to the sole attention-based approach since the sequence lengths in the second and third stages are significantly decreased, as previously noted.

Comparison with State-of-the-Art (SOTA)

All of the SOTA methods mentioned in Table 5.3 are trained by the CE loss. In Table 5.3, HDCAM is compared with recent SOTA recurrent (Dilated LSTM) [20], convolutional (CNN), hybrid LSTM-CNN [94], and hybrid attention-CNN [19] models on Ninapro DB2 dataset [87]. Overall, our model demonstrates better accuracy versus the number of parameters compared to other methods regardless of training objective function. As shown in Table 5.3, for the window size of 200 ms, all variants of the proposed model outperform other SOTA approaches with and without SC loss. For instance, our XXSmall model has 53.3 times less parameter than Dilated LSTM, but obtains a 2.1% (2.34%) gain in the top-1 accuracy when trained with sole CE loss (hybrid loss). Compared to the best performing TC-HGR model (Model 4), our XXSmall and Small models trained with CE loss improve the accuracy for

0.38% and 1.56% with 4.59 and 1.62 times less number of parameters, respectively. Moreover, as shown in Table 5.3, for the window size of 300 ms, XSmall and Small variants of HDCAM, trained with CE loss, obtain 82.61% and 82.91% top-1 accuracy respectively, both surpassing all previous SOTA methods with fewer number of parameters (see Fig. 5.1). Dilation-based LSTM [20], as the previous SOTA model on DB2 dataset, reached 82.4% top-1 accuracy with $1,102,801$ number of parameters, while our XSmall model attains better accuracy (82.61%) with only $40,281$ parameters, i.e., 27.38 times fewer. It is worth noting that our Small model achieved 82.91% top-1 accuracy with $58,441$ parameters trained with CE loss. Small model reaches a new SOTA performance even with sole CE loss training that demonstrates the effectiveness and the generalization of our design.

In Table 5.4, the computation reduction of the proposed model is investigated. More specifically, Table 5.4 provides average inference time for different variants of the HDCAM and TC-HGR. It is important to note that the processing time can vary depending on the hardware used. In this study, we utilized a GeForce GTX 1080 Ti Graphics Cards to obtain the average inference computation time of each model per input sample. The results, as shown in Table 5.4, demonstrate that the inference computation time of all variants of the HDCAM model are smaller in comparison to computation times the TC-HGR model variants, while improving performance as shown in Table 5.3.

Effectiveness of the Multi-scales Local Representation

To extract multi-scale local features, we integrated depth-wise convolution (*DwConv*) with a hierarchical structure in the proposed *HD*Conv encoder. The hierarchical structure besides the multi-scale feature extraction increases the receptive field in a single block. As shown in Table 5.5, replacing the *"hierarchical" DwConv* structure in *HD*Conv with a standard *DwConv* layer degrades the accuracy in all variants of HDCAM, indicating its usefulness in our design. As an example, the top-1 accuracy of the Small model decreased by 0.56% in its non-hierarchical variant.

Table 5.3: Comparing the performance of the proposed HDCAM models with state-of-the-art (SOTA) models on Ninapro DB2 dataset [87]. Our model in the number of parameters and accuracy outperforms the SOTA models.

Model	**Model's Variant**	$W = 200$ **ms**		$W = 300$ **ms**	
		Parameters↓	**Accuracy**↑ (%)	**Parameters**↓	**Accuracy**↑ (%)
Dilated LSTM [20]	4-layer 3rd Order Dilation	1,102,801	79.0	1,102,801	82.4
	4-layer 3rd Order Dilation (pure LSTM)	_	_	466,944	79.7
LSTM-CNN [94]	CNN	_	_	$\approx 1.4M$	77.30
	Hybrid LSTM-CNN	_	_	$\approx 1.1M$	81.96
TC-HGR [19]	Model 1	49,186	80.29	52,066	80.84
	Model 2	68,445	80.63	72,285	81.59
	Model 3	69,076	80.51	67,651	80.95
	Model 4	94,965	80.72	92,945	81.65
HDCAM (CE loss)	XXSmall	**20,686**	81.10	**20,686**	81.73
	XSmall	40,281	81.77	40,281	82.61
	Small	58,441	**82.28**	58,441	**82.91**
HDCAM (Hybrid loss)	XXSmall	**20,686**	81.34	**20,686**	82.03
	XSmall	40,281	82.66	40,281	83.23
	Small	58,441	**82.86**	58,441	**83.54**

Table 5.4: Comparing average process time of different variants of HDCAM and TC-HGR for hand gesture recognition on window size of 200 ms. The process times are reported in millisecond (ms).

Model	**Model's Variant**	**Process Time Per-Sample** ↓ **(ms)**
TC-HGR [19]	Model 1	3.094
	Model 2	3.305
	Model 3	3.425
	Model 4	3.540
HDCAM	XXSmall	2.317
	XSmall	2.626
	Small	2.859

Table 5.5: Evaluating the effectiveness of the multi-scales local representation extraction in the *HD*Conv encoder for window size 300 ms. For Hierarchical models, the scale values (s) for each stage are provided in Table 5.1. For Non-hierarchical models, s is equal to 1 at all stages.

***HD*Conv Encoder**	**Accuracy (%)**		
	XXSmall	XSmall	Small
Hierarchical structure	81.73	82.61	82.91
Non-hierarchical structure	81.27	82.21	82.35

Importance of Using *MHS*Atten Encoders

To examine the importance of *MHS*Atten encoder, we conducted two ablation studies using this encoder at different stages of the network for W=300 ms. In Table 5.6, we kept the total number of *HD*Conv and *MHS*Atten encoders fixed to $[1, 3, 5]$ for experiment 1 to 4. While in experiment 5 to 8, the number of *HD*Conv encoder is set to $[1, 2, 4]$ for all stages, and *MHS*Atten encoder is progressively added to the end of stages. According to both experiments, adding the *MHS*Atten encoder gradually in the last two stages increases accuracy and the number of parameters. In addition, adding a global *MHS*Atten encoder to the first stage is not beneficial since the features in this stage are not mature enough. When at least one *MHS*Atten encoder is used in the network architecture, the best trade-off between accuracy and the number of parameters obtained for the Small model in both experiments, the highlighted rows in Table 5.6. Furthermore, we conducted another experiment to investigate the impact of using the *MHS*Atten encoder at the beginning (after downsampling) versus the end of each stage on the HDCAM architecture. As shown in Table 5.7, better performance is achieved by using the *MHS*Atten encoder as the final block of the stages. In other words, it is more beneficial to encode global representations after extracting local representations rather than the other way around.

Table 5.6: Evaluating the impact of using *MHS*Atten encoder at a different stage of the network for the window size of 300 ms. The listed values show the number of the corresponding encoder in stages 1 to 3 in order. Highlighted rows indicate the Small model.

ID:	Model Configuration	Accuracy↑ (%)	Parameters↓
1 :	*HD*Conv $= [1, 3, 5]$, *MHS*Atten $= [0, 0, 0]$	81.56	$37,673$
2 :	*HD*Conv $= [1, 3, 4]$, *MHS*Atten $= [0, 0, 1]$	82.45	$54,249$
3 :	*HD*Conv $= [1, 2, 4]$, *MHS*Atten $= [0, 1, 1]$	82.91	$58,441$
4 :	*HD*Conv $= [0, 2, 4]$, *MHS*Atten $= [1, 1, 1]$	82.26	$60,817$
5 :	*HD*Conv $= [1, 2, 4]$, *MHS*Atten $= [0, 0, 0]$	81.93	$31,785$
6 :	*HD*Conv $= [1, 2, 4]$, *MHS*Atten $= [0, 0, 1]$	82.55	$52,969$
7 :	*HD*Conv $= [1, 2, 4]$, *MHS*Atten $= [0, 1, 1]$	82.91	$58,441$
8 :	*HD*Conv $= [1, 2, 4]$, *MHS*Atten $= [1, 1, 1]$	82.48	$61,585$

Table 5.7: Evaluating the impact of using *MHS*Atten encoder at the beginning vs. end of each stage for the window size of 300 ms.

***MHS*Atten Encoder**	**Accuracy (%)**		
	XXSmall	XSmall	Small
Fist block of Stage (*MHS*Atten $= [0, 1, 1]$)	81.31	82.37	82.70
Latest block of Stage (*MHS*Atten $= [0, 1, 1]$)	81.73	82.61	82.91

5.3 Conclusion

In this chapter, a novel resource-efficient architecture, referred to as the HDCAM, is developed for HGR from sparse multichannel sEMG signals. In comparison to SOTA methods, HDCAM is more effective in terms of both parameters and performance. Its lightweight design is a key step toward incorporating DNN models into wearable for immersive HMI. HDCAM is developed by effectively combining the advantages of Attention-based and CNN-based models for low-powered devices. Specifically, HDCAM is empowered with convolution and attention-based encoders, namely *HD*Conv and *MHS*Atten, to efficiently extract local and global representations of the input sEMG sequence. We showed that by proper design of convolution-based architectures, we not only can extract a multi-scale local representation but also can increase the receptive field in a single block.

Chapter 6

Conclusion and Remaining Works

Given the significant advancements made in the domain of wearable technologies, there has been a surge of interest in the development of intelligent algorithms capable of inferring valuable information from physiological biosignals collected from these devices using Machine Learning (ML) techniques, especially Deep Neural Networks (DNNs) [13]. To date, many wearable devices collect biomedical data from the human body, including Electrocardiogram (ECG), Photoplethysmogram (PPG), and surface Electromyogram (sEMG) biosignals, which are among the most widely monitored signals in clinical settings [9]. Utilizing these widely used signals and capitalizing on the significant advances in deep learning, the primary focus of the proposed thesis is on the development of advanced ML algorithms based on DNNs to increase the accuracy of wearable devices in specific applications.

An inevitable increase in the population of seniors makes continuous BP monitoring essential as it provides invaluable information about individuals' cardiovascular conditions. In Chapter 3, we first identified two key drawbacks associated with the existing continuous BP estimation models, i.e., (i) Relying heavily on extraction of hand-crafted features, i.e., ignoring the real potential of deep learning in utilization of the intrinsic features (deep features) and instead using representative hand-crafted features prior to extraction of deep features, and; (ii) Lack of a benchmark dataset for evaluation and comparison of developed deep learning-based BP estimation algorithms. To alleviate these issues, we proposed an efficient algorithm, referred to

as the BP-Net, based on the deep learning techniques for the continuous, cuff-less, and alignment-free prediction of systolic and diastolic BP. In the proposed BP-Net architecture, raw ECG and PPG signals are utilized without extraction of PAT features to explore the real potential of deep learning in utilization of intrinsic features (deep features). The proposed BP-Net architecture is more accurate than canonical recurrent networks in the BP estimation task. Moreover, by capitalizing on the significant importance of continuous BP monitoring and the fact that datasets used in recent literature are not unified and properly defined, a benchmark data set is constructed from the MIMIC-I and MIMIC-III databases to provide a unified base for evaluation and comparison of deep learning-based BP estimation algorithms. The proposed BP-Net architecture is evaluated based on this benchmark dataset demonstrating promising results. The dataset can be accessed through the link provided in Reference [126].

To further improve potential applicability of the prepared dataset, several avenues for future research could be explored, including:

- Extending the dataset by extracting and including relevant clinical information for the patients (such as age and gender). These updates will be appended to the dataset, which can be accessed via the link provided in Reference [126].
- Furthermore, adding more labels, such as annotating Heart Rates (HR) of patients from collected ECG records, which would allow us to investigate a new set of Neural Networks for HR prediction using ECG or even PPG signals, with extremely useful applications in healthcare wearable devices, particularly in clinical and fitness industries. More specifically, the early and correct diagnosis of cardiac abnormalities (such as Atrial Fibrillation (A-fib), Tachycardia, Bradycardia, and Pause) can increase the chances of successful treatments or possibly allow the caregiver to take appropriate action in an emergency.
- One general challenge in BP estimation task is the time- and user-dependent nature of the ECG, PPG, and blood pressure signals. Variations among the probability distribution of these biomedical signals across different subjects make the experience gained on an unseen person difficult. Therefore, domain adaptation methods are highly recommended in this field of study, where learning

methods focus on transferring information between a source and a target domain despite the existence of a distribution shift among them. In this study, the training of our network is fully supervised. Therefore, following previous studies [48,50,52,56], the data for each patient is divided into train, validation, and test sets. We consider this as a limitation for our current study and a future research direction.

In Chapter 4, by capitalizing on the fact that sEMG signals have found technical applications in the development of HMI systems such as VR/AR environments and neural rehabilitation devices, including multifunction prostheses, we proposed investigation of Transformers-based architectures' capacity to improve the analysis of sparse sEMG signals and bridge the gap between recent academic research and clinical/industrial settings. In this context, we presented our proposed architecture based on the Transformers (named as TraHGR), which achieved state-of-the-art performance in the most common dataset (DB2 (49 gestures)) for sEMG-based hand gesture recognition. While Transformers have shown promise for hand gesture recognition, their high computational demands present a challenge when it comes to embedding these models in wearable devices. Wearables have limited processing power and battery life, which makes it difficult to run transformer models without causing performance issues or draining the battery. One possible solution is to rely on cloud services to offload the computational burden. However, to tackle the challenge of directly embedding transformer models into wearable devices, alternative architectures could be explored to reduce the computational requirements of transformers while still achieving high performance.

In Chapter 5, we introduce a novel light-weight architecture, the HDCAM, for hand gesture recognition (HGR) using sparse multichannel sEMG signals. The key objective behind the design of HDCAM was to ensure its resource efficiency while maintaining comparable or better performance than the current state-of-the-art methods. By using a lightweight design, the HDCAM aims to enable the integration of deep neural network models into wearable devices for human-machine interaction. The architecture leverages the benefits of both attention-based and CNN-based models, specifically by utilizing the *HD*Conv and *MHS*Atten encoders to extract both local and global representations of the input sEMG sequence in an efficient manner.

The HDCAM represents a significant step forward in the development of low-power HGR models for wearable applications.

There are several avenues for future research that can be pursued to further improve the performance of sEMG-based hand gesture recognition:

- One possible direction is to explore the potential of incorporating additional modalities, such as accelerometers or gyroscopic sensors, to enhance the accuracy of hand gesture recognition. This approach can potentially provide complementary information to sEMG signals and further improve the model's ability to recognize complex hand movements.
- The impact of various factors on the distribution of sEMG signals, such as the time variability between days and the type of amputation, were not explored in this thesis but represent important areas for future investigation. Additionally, the misplacement or displacement of sensors can also significantly influence the distribution of sEMG signals, which remains an open research question yet to be addressed. Therefore, these limitations of the current study highlight promising directions for future research.
- A potential research direction could be exploring the capabilities of spatio-temporal Transformers for HGR tasks by leveraging the high-resolution spatio-temporal data provided by High-Density sEMG (HD-sEMG) signals compared to those obtained from sparse electrodes. The recent release of HD-sEMG datasets such as [151, 152] can facilitate this research. Utilizing spatio-temporal Transformers enables the extraction of both local and long-range temporal features from the HD-sEMG records, leading to a more comprehensive understanding of the signal patterns and potentially improving HGR performance. Furthermore, developing novel pooling mechanisms that encourage the network to focus on representative frames and drop non-informative features along the temporal dimension could be a promising direction to reduce computation and optimize the spatio-temporal model.
- It is worth noting that some studies with HD-sEMG systems have found that the relationship between accuracy and the number of electrodes is not necessarily a monotonic function, and increasing the number of electrodes beyond an optimal point can cause the system to lose accuracy due to increased noise

and over-fitting. Thus one use of HD-sEMG arrays is to select optimal electrode placement [71]. Other studies record data with the full array but only utilize a subset of the electrode channels (frames) for the control algorithm. To enhance the performance of the spatio-temporal model, instead of discarding some information such as using only odd frames, an intriguing direction would be to explore novel pooling mechanisms. These mechanisms would not only enable the network to concentrate on essential frames but also eliminate unimportant features/frames along the temporal dimension, thereby reducing computation. This could potentially be a fruitful area for future research to improve the spatio-temporal model

- Finally, the use of Transformers-based architecture to develop an adaptive learning method with a focus on increasing the robustness of sEMG classifiers and improving inter-subject accuracy will be an interesting direction for our future research. More specifically, for a future direction utilize contrastive learning for learning subject-independent representation, resulting in a more robust model for sEMG classifiers. In general, contrastive learning is usually applied in the field of unsupervised learning [156–158], which allows deep models to learn an informative representation by attracting positive pairs from an anchor and pulling negative pairs away from it [156]. To this end, developing a new strategy that uses contrastive learning to develop subject-independent representations by increasing the similarity of features resulting from different subjects but belonging to the same class could be a valuable contribution.

Bibliography

[1] J.L. Helbostad, B. Vereijken, C. Becker, C. Todd, K. Taraldsen, m. Pijnappels, K. Aminian, and S. Mellone, "Mobile health applications to promote active and healthy ageing," *Sensors*, vol. 17, p. 622, 2017.

[2] J.W. Lee, and K.S. Yun, "ECG monitoring garment using conductive carbon paste for reduced motion artifacts," *Polymers*, vol. 9, p. 439, 2017.

[3] S. Zhao, J. Liu, Z. Gong, Y. Lei, X. OuYang, C.C. Chan, and S. Ruan, "Wearable physiological monitoring system based on electrocardiography and electromyography for upper limb rehabilitation training," *Sensor*, vol. 20, no. 17, p. 4861, 2020.

[4] M. Tomasini, S. Benatti, B. Milosevic, E. Farella, and L. Benini, "Power line interference removal for high-quality continuous biosignal monitoring with low-power wearable devices.," *IEEE Sensors Journal*, vol. 16, no. 10, pp. 3887-3895, 2016.

[5] C. Wong, Z.-Q. Zhang, B. Lo, and G.-Z. Yang, "Wearable sensing for solid biomechanics: A review," *IEEE Sensors Journal*, vol. 15, no. 5, pp. 2747–2760, 2015.

[6] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.

[7] J. Kim, et al., "Wearable biosensors for healthcare monitoring," *Nature biotechnology*, vol. 37, no. 4, pp. 389-406, 2019.

[8] R.B. Reilly, and T.C. Lee, "Electrograms (ecg, eeg, emg, eog)," *Technology and Health Care*, vol. 18, no. 6, pp. 443-458, 2010.

[9] M. Pflugradt, et al., "Multi-modal signal acquisition using a synchronized wireless body sensor network in geriatric patients," *Biomedical Engineering/Biomedizinische Technik*, vol. 61, no. 1, pp. 57-68, 2016.

[10] J. Chou, R.T. Llamas, and J. Ubrani, "Wearable Devices Market Share (Updated: 22 March 2022), " Available online: `https://www.idc.com/promo/wearablevendor`, accessed on 20 April 2022.

[11] P.J. Soh, G.A. Vandenbosch, M. Mercuri, and D.M.P. Schreurs, "Wearable wireless health monitoring: Current developments, challenges, and future trends," *IEEE microwave magazine*, vol. 16, no. 4, pp. 55-70, 2015.

[12] K. Guk, et al., "Evolution of wearable devices with real-time disease monitoring for personalized healthcare," *Nanomaterials*, vol. 9, no. 6, p. 813, 2019.

[13] A.R. Dargazany, P. Stegagno, and K. Mankodiya, "WearableDL: wearable internet-of-things and deep learning for big data analytics—concept, literature, and future," *Mobile Information Systems*, 2018.

[14] A. Vaswani, *et al.*, "Attention is All You Need," *arXiv preprint arXiv:1706.03762*, 2017a.

[15] T.B. Brown., et al., "Language Models are Few-shot Learners," *arXiv preprint arXiv:2005.14165*, 2020.

[16] J. Devlin, M.W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[17] A. Dosovitskiy, *et al.*, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," *arXiv preprint arXiv:2010.11929*, 2020.

[18] D. Farina, N. Jiang, H. Rehbaum, A. Holobar, B. Graimann, H. Dietl, and O.C. Aszmann, "The Extraction of Neural Information from the Surface EMG for the Control of Upper-Limb Prostheses: Emerging Avenues and Challenges," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp.797–809, 2014.

[19] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S.F. Atashzar, and A. Mohammadi, "Hand Gesture Recognition Using Temporal Convolutions and Attention Mechanism," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2022, pp. 1196-1200.

[20] T. Sun, Q. Hu, P. Gulati, and S.F. Atashzar, "Temporal Dilation of Deep LSTM for Agile Decoding of sEMG: Application in Prediction of Upper-limb Motor Intention in NeuroRobotics.," *IEEE Robotics and Automation Letters*, 2021.

[21] H. Yoon, and S.H. Park, "A non-touchscreen tactile wearable interface as an alternative to touchscreen-based wearable devices," *Sensors*, vol. 20, no. 5, p.1275, 2020.

[22] M.H.U. Rehman, C.S. Liew, T.Y. Wah, J. Shuja, and B. Daghighi, "Mining personal data using smartphones and wearable devices: A survey," *Sensors*, vol. 15, no. 2, pp.4430-4469, 2015.

[23] "World Population Ageing," *Department of Economics and Social Affairs, Population Devision*, United Nations, 2020.

[24] W.H. Lin, F. Chen, Y. Geng, N. Ji, P. Fang, and G. Li, "Towards Accurate Estimation of Cuffless and Continuous Blood Pressure using Multi-order Derivative and Multivariate Photoplethysmogram Features," *Biomedical Signal Processing and Control*, 63, p.102198., 2021.

[25] M.R. Mohebbian, A. Dinh, K. Wahid, and M.S. Alam, "Blind, cuff-less, Calibration-free and Continuous blood Pressure Estimation using Optimized Inductive Group Method of Data Handling" *Biomedical Signal Processing and Control*, 57, p.101682, 2020.

[26] A. Chandrasekhar, *et al.*, "Smartphone-based Blood Pressure Monitoring via the Oscillometric Finger-pressing Method," *Sci. Transl. Med.,* vol. 10, 2018.

[27] F. Miao, Z. Liu, J. Liu, B. Wen and Y. Li, "Multi-sensor Fusion Approach for Cuff-less Blood Pressure Measurement," *IEEE J. Biomed. & Health Inf.*, 2019. In Press.

[28] C. El-Hajj, and P.A. Kyriacou, "A Review of Machine Learning Techniques in Photoplethysmography for the Non-invasive Cuff-less Measurement of Blood Pressure," *Biomedical Signal Processing and Control*, vol. 58, pp. 101870, 2020.

[29] Y. Qiu, *et al.*, "Cuffless Blood Pressure Estimation based on Composite Neural Network and Graphics Information," *Biomedical Signal Processing and Control*, vol. 70, 2021.

[30] M.S. Tanveer, and M.K. Hasan, "Cuffless Blood Pressure Estimation from Electrocardiogram and Photoplethysmogram using Waveform based ANN-LSTM Network", *Biomedical Signal Processing and Control*, vol. 51, pp. 382-392, 2019.

[31] B. Zhang, J. Ren, Y. Cheng, B. Wang and Z. Wei, "Health Data Driven on Continuous Blood Pressure Prediction Based on Gradient Boosting Decision Tree Algorithm," *IEEE Access*, vol. 7, pp. 32423-32433, 2019.

[32] S.G. Khalid, J. Zhang, F. Chen, D. Zheng, "Blood Pressure Estimation using Photoplethysmography Only: Comparison Between Different Machine Learning Approaches," *J. Healthc. Eng.*, pp. 1-13, 2018.

[33] M. Simjanoska, M. Gjoreski, M. Gams, A.M. Bogdanova, "Non-invasive Blood Pressure Estimation from ECG using Machine Learning Techniques," *Sensors*, vol. 18, no. 4, 2018.

[34] M. Kachuee, M.M. Kiani, H. Mohammadzade, M. Shabany, "Cuffless Blood Pressure Estimation Algorithms for Continuous Health-care Monitoring," *IEEE Trans. Biomed. Eng.*, vol. 64, no. 4, pp. 859-869, 2017.

[35] F.P.W. Lo, C.X.T. Li, J. Wang, J. Cheng, and M.Q.H. Meng, "Continuous Systolic and Diastolic Blood Pressure Estimation Utilizing Long Short-Term Memory Network," *IEEE Int. Conf. Eng. in Med. & Biol. Society (EMBC)*, pp. 1853-1856, 2017.

[36] H. Xiao, M. Butlin, I. Tan, A. Qasem and A. P. Avolio, "Estimation of Pulse Transit Time From Radial Pressure Waveform Alone by Artificial Neural Network," *IEEE J. Biomed. & Health Inf.*, vol. 22, no. 4, pp. 1140-1147, July 2018.

[37] Y. Ma, *et al.* "Relation between Blood Pressure and Pulse Wave Velocity for Human Arteries," *Proceedings of the National Academy of Sciences*, vol. 115, no. 44, pp. 11144-11149, 2018.

[38] T.H. Huynh, R. Jafari, W.Y. Chung, "Noninvasive Cuffless Blood Pressure Estimation using Pulse Transit Time & Impedance Plethysmography," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 4, pp. 967-976, 2019.

[39] R. Mukkamala, and J.O. Hahn, "Toward Ubiquitous Blood Pressure Monitoring via Pulse Transit Time: Predictions on Maximum Calibration Period and Acceptable Error Limits," *IEEE Trans. Biomed. Eng.*, vol. 65, no. 6, pp. 1410-1420, 2018.

[40] S. Ahmad, *et al.* "Electrocardiogram-assisted Blood Pressure Estimation," *IEEE Trans. Biomed. Eng.*, vol. 59, no. 3, pp. 608-618, 2012.

[41] J.M. Huttunen, L. Kärkkäinen, and H. Lindholm, "Improving Pulse Transit Time Estimation of Aortic PWV and Blood Pressure using Machine Learning and Simulated Training Data," *arXiv preprint arXiv*, 1903.02262, 2019.

[42] Y. Yoon *et al.*, "Cuff-Less Blood Pressure Estimation Using Pulse Waveform Analysis and Pulse Arrival Time," *IEEE J. Biomed. & Health Inf.*, vol. 22, no. 4, pp. 1068-1074, July 2018.

[43] Z. Tang *et al.*, "A Chair–Based Unobtrusive Cuffless Blood Pressure Monitoring System Based on Pulse Arrival Time," *IEEE J. Biomed. & Health Inf.*, vol. 21, no. 5, pp. 1194-1205, Sept. 2017.

[44] H. Gesche, D. Grosskurth, G, Küchler, and A. Patzak, "Continuous Blood Pressure Measurement by using the Pulse Transit Time: Comparison to a Cuff-based Method," *European J. Applied Physiology*, vol. 112, no. 1, pp. 309-315, 2012.

[45] I. Sharifi, S. Goudarzi, and M.B. Khodabakhshi, "A Novel Dynamical Approach in Continuous Cuffless Blood Pressure Estimation based on ECG and PPG Signals," *Artificial Intell. Medic.*, pp. 143-151, 2019.

[46] A. Strin, "Improvements in Indirect Blood Pressure Estimation via Electrocardiography and Photoplethysmography," 2016.

[47] M. Rehman, *et al.*, "Multiday EMG-based Classification of Hand Motions with Deep Learning Techniques," *Sensors*, vol. 18, no. 8, 2018.

[48] P. Su, X.R. Ding, Y.T. Zhang, J. Liu, F. Miao, and N. Zhao, "Long-term Blood Pressure Prediction with Deep Recurrent Neural Networks," *IEEE EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 323-328, 2018.

[49] S. Lee, and J.H. Chang, "Oscillometric Blood Pressure Estimation based on Deep Learning," *IEEE Trans. Ind. Informat.*, vol. 13, no. 2, pp. 461-472, 2017.

[50] M.S. Tanveer, and M.K. Hasan, "Cuffless Blood Pressure Estimation from Electrocardiogram and Photoplethysmogram using Waveform based ANN-LSTM Network," *Biomedical Signal Processing and Control*, vol. 51, no. 1, pp. 382-392, 2019.

[51] A. Paviglianiti *et al.*, "A Comparison of Deep Learning Techniques for Arterial Blood Pressure Prediction," *J. Cognitive Computation*, pp. 1-22, 2021.

[52] X. Fan *et al.*, "An Adaptive Weight Learning-Based Multitask Deep Network for Continuous Blood Pressure Estimation Using Electrocardiogram Signals," *J. Sensors*, vol. 21, no. 5, p. 1595, 2021.

[53] K. Qin, W. Huang, and T. Zhang, "Deep Generative Model with Domain Adversarial Training for Predicting Arterial Blood Pressure Waveform from Photoplethysmogram Signal," *J. Biomedical Signal Processing and Control*, 70, p. 102972, 2021.

[54] M. Saeed M,*et al.*, "Multiparameter Intelligent Monitoring in Intensive Care II (MIMIC-II): a public-access intensive care unit database." *Critical care medicine*, vol. 39, no. 5, p. 952, 2011.

[55] J.C. Ruiz-Rodríguez *et al.*, "Innovative Continuous Non-invasive Cuffless Blood Pressure Monitoring based on Photoplethysmography Technology," *Intensive care medicine*, vol. 39, no. 9, pp. 1618-1625, 2013.

[56] H. Eom *et al.*, "End-to-end Deep Learning Architecture for Continuous Blood Pressure Estimation Using Attention Mechanism," *Sensors*, vol. 20, no. 8, p. 2338, 2020.

[57] W. Wei, *et al.*, "Surface Electromyography-based Gesture Recognition by Multi-view Deep Learning," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 10, pp. 2964-2973, 2019.

[58] Y. Hu, *et al.*, "A Novel Attention-based Hybrid CNN-RNN Architecture for sEMG-based Gesture Recognition," *PloS one 13*, no. 10, 2018.

[59] W. Geng, *et al.*, "Gesture Recognition by Instantaneous Surface EMG Images," *Scientific Reports*, 6, p. 36571, 2016.

[60] Y. Qu, H. Shang, J. Li, and S. Teng, "Reduce Surface Electromyography Channels for Gesture Recognition by Multitask Sparse Representation and Minimum Redundancy Maximum Relevance," *Journal of Healthcare Engineering*, 2021.

[61] A. Toro-Ossaba, *et al.*, "LSTM Recurrent Neural Network for Hand Gesture Recognition Using EMG Signals," *Applied Sciences*, vol. 12, no. 9, p.9700, 2022.

[62] M. Ergeneci, *et al.*, "An embedded, eight channel, noise canceling, wireless, wearable sEMG data acquisition system with adaptive muscle contraction detection," *IEEE Trans. Biomed. Circuits Syst.*, vol. 12, no. 1, pp. 68–79, Feb. 2018.

[63] E. Rahimian, S. Zabihi, A. Asif, S.F. Atashzar, and A. Mohammadi, "Few-Shot Learning for Decoding Surface Electromyography for Hand Gesture Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1300-1304.

[64] E. Rahimian, S. Zabihi, F. Atashzar, A. Asif, A. Mohammadi, "XceptionTime: Independent Time-Window XceptionTime Architecture for Hand Gesture Classification," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1304-1308, 2020.

[65] P. Tsinganos, B. Cornelis, J. Cornelis, B, Jansen, and A. Skodras, "Improved Gesture Recognition Based on sEMG Signals and TCN," *International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 1169-1173, 2019.

[66] S.E. Ovur, *et al.*, "A Novel Autonomous Learning Framework to Enhance sEMG-based Hand Gesture Recognition using Depth Information," *Biomedical Signal Processing and Control*, vol. 66, p.102444, 2021.

[67] C.L. Toledo-Peral, *et al.*, "Virtual/Augmented Reality for Rehabilitation Applications Using Electromyography as Control/Biofeedback: Systematic Literature Review. Electronics," *Electronics*, vol. 14, no. 11, p.2271, 2022.

[68] L. Guo, Z. Lu, and L. Yao, "Human-machine Interaction Sensing Technology Based on Hand Gesture Recognition: A Review," *IEEE Transactions on Human-Machine Systems*, 2021.

[69] A. Mongardi, *et al.*, "Hand Gestures Recognition for Human-Machine Interfaces: A Low-Power Bio-Inspired Armband," *IEEE Transactions on Biomedical Circuits and Systems*, 2022.

[70] B. Han and H. D. Schotten, "Multi-Sensory HMI for Human-Centric Industrial Digital Twins: A 6G Vision of Future Industry," *IEEE Symposium on Computers and Communications (ISCC)*, pp. 1-7, 2022.

[71] H. Daley, K. Englehart, L. Hargrove, U. Kuruganti "High density electromyography data or normally limbed and transradial amputee subjects for multifunction prosthetic control," *Journal of Electromyography and Kinesiology*, vol. 22, 2012.

[72] M. Twardowski, S. Roy, Z. Li, P Contessa, G. De Luca, and J. Kline, "Motor Unit Drive: A Neural Interface for Real-time Upper Limb Prosthetic Control," *Journal of Neural Engineering*, vol. 16, 2019.

[73] M. Simao, N. Mendes, O. Gibaru, and P. Neto "A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction," *IEEE Access*, vol. 7, pp. 39564-39582, 2019.

[74] U. Côté-Allard, *et al.*, "Interpreting Deep Learning Features for Myoelectric Control: A Comparison with Handcrafted Features," *Frontiers in bioengineering and biotechnology*, 8, p.158, 2020.

[75] D. Esposito, *et al.*, "A Piezoresistive Array Armband with Reduced Number of Sensors for Hand Gesture Recognition," *Frontiers in Neurorobotics*, vol. 13, p. 114, 2020.

[76] M. Tavakoli, C. Benussi, P.A. Lopes, L.B. Osorio, and A.T. de Almeida, "Robust Hand Gesture Recognition with a Double Channel Surface EMG Wearable Armband and SVM Classifier," *Biomedical Signal Processing and Control*, vol. 46, pp. 121-130, 2018.

[77] G.R. Naik, A.H. Al-Timemy, H.T. Nguyen, "Transradial Amputee Gesture Classification using an Optimal Number of sEMG Sensors: an Approach using ICA Clustering," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 8, pp. 837–846, 2015.

[78] S. Pizzolato, *et al.*, "Comparison of Six Electromyography Acquisition Setups on Hand Movement Classification Tasks," *PLoS ONE*, vol. 12, no. 10, pp. 1-7, 2017.

[79] M. Atzori, M. Cognolato, and H. Müller, "Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands," *Frontiers in neurorobotics 10*, p.9, 2016.

[80] A. K. Clarke *et al.*, "Deep Learning for Robust Decomposition of High-Density Surface EMG Signals," *IEEE Trans. Biomed. Eng.*, 2020, In Press.

[81] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Surface EMG-Based Hand Gesture Recognition via Hybrid and Dilated Deep Neural Network Architectures for Neurorobotic Prostheses," *Journal of Medical Robotics Research*, 2020, pp. 1-12.

[82] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Semg-based Hand Gesture Recognition via Dilated Convolutional Neural Networks," *Global Conference on Signal and Information Processing, GlobalSIP*, 2019.

[83] E. Rahimian, S. Zabihi, A. Asif, S.F. Atashzar, and A. Mohammadi, "Few-Shot Learning for Decoding Surface Electromyography for Hand Gesture Recognition," *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021, pp. 1300-1304.

[84] W. Wei, *et al.*, "A Multi-stream Convolutional Neural Network for sEMG-based Gesture Recognition in Muscle-computer Interface," *Pattern Recognition Letters*, 119, pp. 131-138, 2019.

[85] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using mutual components analysis for multifunction prosthetic fingers control," *12th Int. Conf. Control Autom. Robot. Vis.(ICARCV)*, 2012, pp. 1534–1539.

[86] X. Zhai, B. Jelfs, R. H. Chan, and C. Tin, "Self-recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control based on Convolutional Neural Network," *Frontiers in neuroscience*, 11, p.379, 2017.

[87] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.G.M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Muller, "Electromyography Data for Non-Invasive Naturally-controlled Robotic Hand Prostheses," *Scientific data*, vol. 1, no. 1, pp.1-13, 2014.

[88] Z. Ding, *et al.*, "sEMG-based Gesture Recognition with Convolution Neural Networks," *Sustainability 10*, no. 6, p. 1865, 2018.

[89] W. Wei, Y.Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface.," *Pattern Recognition Letters*, 2017.

[90] M. Simao, P. Neto, and O. Gibaru, "EMG-based Online Classification of Gestures with Recurrent Neural Networks," *Pattern Recognition Letters*, pp.45-51, 2019.

[91] F. Quivira et al., "Translating sEMG Signals to Continuous Hand Poses Using Recurrent Neural Networks.," *in Proc. IEEE EMBS Int. Conf. Biomed. Health Informat.*, 2018, pp. 166–169.

[92] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Surface EMG-Based Hand Gesture Recognition via Hybrid and Dilated Deep Neural Network Architectures for Neurorobotic Prostheses," *Journal of Medical Robotics Research*, pp. 1-12, 2020.

[93] N.K. Karnam, S.R. Dubey, A.C. Turlapaty, and B. Gokaraju, "EMGHandNet: A Hybrid CNN and Bi-LSTM Architecture for Hand Activity Classification using Surface EMG Signals," *Biocybernetics and Biomedical Engineering*, vol. 42, no. 1, pp .325-340, 2022.

[94] P. Gulati, Q. Hu, and S.F. Atashzar, "Toward Deep Generalization of Peripheral Emg-based Human-Robot Interfacing: A Hybrid Explainable Solution for Neurorobotic Systems," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp .2650-2657, 2021.

[95] S. Bai, J.Z. Kolter, and V. Koltun, " An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[96] E. Rahimian, S. Zabihi, S. F. Atashzar, A. Asif, and A. Mohammadi, "Semg-based Hand Gesture Recognition via Dilated Convolutional Neural Networks," *Global Conference on Signal and Information Processing, GlobalSIP*, 2019.

[97] P. Tsinganos, B. Jansen, J. Cornelis, and A. Skodras, " Real-Time Analysis of Hand Gesture Recognition with Temporal Convolutional Networks," *Sensors*, vol. 22, no. 5, p. 1694, 2022.

[98] E. Rahimian, S. Zabihi, A. Asif, S.F. Atashzar, and A. Mohammadi, "Trustworthy Adaptation with Few-Shot Learning for Hand Gesture Recognition," *IEEE International Conference on Autonomous Systems (ICAS)*, pp. 1-5, 2021.

[99] E. Rahimian, S. Zabihi, A. Asif, D. Farina, S.F. Atashzar, and A. Mohammadi, "FS-HGR: Few-shot Learning for Hand Gesture Recognition via ElectroMyography," *IEEE Trans. Neural Syst. Rehabil. Eng.*, 2021.

[100] S. Wang, *et al.*, "Improved Multi-Stream Convolutional Block Attention Module for sEMG-Based Gesture Recognition," *Frontiers in Bioengineering and Biotechnology*, 10, 2022.

[101] Y. Hu, *et al.*, "A Novel Attention-based Hybrid CNN-RNN Architecture for sEMG-based Gesture Recognition," *PloS one*, vol. 13, no. 10, p.e0206049, 2018.

[102] T. Bao, S. Q. Xie, P. Yang, P. Zhou and Z.Q. Zhang, "Toward Robust, Adaptiveand Reliable Upper-Limb Motion Estimation Using Machine Learning and Deep Learning–A Survey in Myoelectric Control," *IEEE Journal of Biomedical and Health Informatics*, vol. 26, no. 8, pp. 3822-3835, 2022.

[103] M. Kim, W. K. Chung, and K. Kim, "Subject-Independent sEMG Pattern Recognition by Using a Muscle Source Activation Model," *IEEE Robotics and Automation Letters*,vol. 5, no. 4, pp. 5175-5180, 2020.

[104] K. Watanabe, M. Kouzaki, M. Ogawa, H. Akima, and T. Moritani, "Relationships between muscle strength and multi-channel surface EMG parameters in eighty eight elderly," *European Review of Aging and Physical Activity*, vol. 15, 2018.

[105] E. C. Hill et al., "Effect of sex on torque, recovery, EMG, and mmg responses to fatigue," *J. Musculoskelet Neuronal Interact*, vol. 16, no. 4 pp. 310–317, 2016.

[106] J. He, D. Zhang, N. Jiang, X. Sheng, D. Farina, and X. Zhu, "User adaptation in long-term, open-loop myoelectric training: Implications for EMG pattern recognition in prosthesis control," *J. Neural Eng.*, vol. 12, no. 4, 2015.

[107] 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.*, vol. 12, 2015

[108] M. Jochumsen, A. Waris, and E. N. Kamavuako, "The effect of arm position on classification of hand gestures with intramuscular EMG," *Biomed. Signal Process. Control*, vol. 43, pp. 1–8, 2018.

[109] K.T. Kim, C. Guan, and S.W. Lee, "A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 94-103, 2020.

[110] A. Ameri, M. A. Akhaee, E. Scheme, and K. Englehart, "A deep transfer learning approach to reducing the effect of electrode shift in EMG pattern recognition-based control," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 2, pp. 370-379, 2020.

[111] Z. Yu, J. Zhao, Y. Wang, L. He, and S. Wang, "Surface EMG-based instantaneous hand gesture recognition using convolutional neural network with the transfer learning method," *Sensors*, vol. 21, no. 7, 2021.

[112] F. Demir, V. Bajaj, M. C. Ince, S. Taran, and A. ¸Sengür "Surface EMG signals and deep transfer learning-based physical action classification," *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8455–8462, 2019.

[113] 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. 54789–54801, 2020.

[114] U. Côté-Allard et al., "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760–771, 2019.

[115] A.L. Goldberger, L.A. Amaral, L. Glass, J.M.Hausdorff, P.C. Ivanov, R.G. Mark, J.E. Mietus, G.B. Moody, C.K. Peng, and H.E. Stanley, "PhysioBank, PhysioToolkit, and PhysioNet: Components of a New Research Resource for Complex Physiologic Signals," *Circulation*, vol. 101, no. 23, pp. e215-e220, 2000.

[116] A.E. Johnson, *et al.*, " MIMIC-III, a freely accessible critical care database", *Scientific data*, vol. 3, no. 1, pp. 1-9, 2016.

[117] F.N. Fritsch, and R.E. Carlson, "Monotone piecewise cubic interpolation," *SIAM Journal on Numerical Analysis*, vol. 17, no. 2, pp.238-246, 1980.

[118] M. Abbas *et al.*, "Positivity-preserving C2 rational cubic spline interpolation," *ScienceAsia*, vol. 39, no. 2, pp.208-213, 2013.

[119] S. Butt, K.W. Brodlie, "Preserving positivity using piecewise cubic interpolation," *Computers and Graphics*, vol. 17, no. 1, pp.55-64, 1993.

[120] G. Slapnicar, *et al.*, "Blood pressure estimation from photoplethysmogram using a spectro-temporal deep neural network," *Sensors*, vol. 19, no. 15, pp.3420, 2019.

[121] J. Esmaelpoor, M.H. Moradi, and A. Kadkhodamohammadi, "A multistage deep neural network model for blood pressure estimation using photoplethysmogram signals," *Computers in Biology and Medicine*, vol. 120, p. 103719, 2020.

[122] F. Miao, *et al.*, "Continuous blood pressure measurement from one-channel electrocardiogram signal using deep-learning techniques," *Artificial Intelligence in Medicine*, vol. 108, p. 101919, 2020.

[123] S.S Mousavi, *et al.*, "Blood pressure estimation from appropriate and inappropriate PPG signals using A whole-based method," *Biomedical Signal Processing and Control*, vol. 47, pp.196-206, 2019.

[124] M. Panwar, *et al.*, "PP-Net: A deep learning framework for PPG-based blood pressure and heart rate estimation," *IEEE Sensors Journal*, vol. 20, no. 17, pp.10000-10011, 2020.

[125] Y.H. Li, *et al.*, "Real-time cuffless continuous blood pressure estimation using deep learning model," *Sensors*, vol. 20, no. 19, pp.5606, 2020.

[126] S. Zabihi, *et al.*, "BP-Net Dataset", *https://osf.io/n69ym/?view_only=f74f6a70b5754efab108ce6763fa07d9*, 2022.

[127] A.V.D. Oord and *et al.* "Wavenet: A Generative Model for Raw Audio," *ArXiv preprint arXiv:1609.03499*, 2016.

[128] T. Sercu and G. Vaibhava, "Dense Prediction on Sequences with Time-Dilated Convolutions for Speech Recognition," *arXiv preprint arXiv:1611.09288*, 2016.

[129] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *IEEE Conf. Comput. Vision & Pattern Recognition*, pp. 770-778, 2016.

[130] D. A. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUS)," *arXiv preprint arXiv:1511.07289*, 2015.

[131] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *J. Machine Learning Research*, vol. 15, no. 1, pp. 1929-1958, 2014.

[132] T. Salimans and D. P. Kingma, "Weight Normalization: A Simple Reparameterization to Accelerate Training of Deep Neural Networks," *Advances in Neural Information Process. Sys.*, pp. 901-909, 2016.

[133] Y. Wang, et. al., "Transformer-based Acoustic Modeling for Hybrid Speech Recognition," *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6874-6878.

[134] A. Gijsberts, M. Atzori, C. Castellini, H. Müller, and B. Caputo, "Movement Error Rate for Evaluation of Machine Learning Methods for sEMG-based Hand Movement Classification," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 735-744, 2014.

[135] JL Ba, JR Kiros, and G.E. Hinton, "Layer normalization," *arXiv preprint arXiv:1607.06450*, 2016.

[136] DP. Kingma, and J. Ba, "Adam: A Method for Stochastic Optimization," *ICLR*, 2015.

[137] M. Atzori, *et al.*, "Electromyography Data for Non-Invasive Naturally-Controlled Robotic Hand Prostheses," *Scientific data 1*, no. 1, pp. 1-13, 2014.

[138] U. Côté-Allard, *et al.*, "Deep Learning for Electromyographic Hand Gesture Signal Classification using Transfer Learning," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 4, pp. 760-771, 2019.

[139] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*,vol. 1, no. 6, pp. 80–83, 1945.

[140] B. Hudgins, P. Parker, and R.N. Scott, "A New Strategy for Multifunction Myoelectric Control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, p.82-94, 1993.

[141] C. Castellini, *et al.*, "Proceedings of the first workshop on peripheral machine interfaces: Going beyond traditional surface electromyography," *Frontiers in neurorobotics,* 8, p.22, 2014.

[142] G. S. Dhillon and K. W. Horch, "Direct neural sensory feedback and control of a prosthetic arm," *IEEE Transactions on Neural Systems and Rehabilitation Engineering,* vol. 13, no. 4, pp. 468-472, 2005.

[143] B. Milosevic, S. Benatti and E. Farella, "Design challenges for wearable EMG applications," *Design, Automation and Test in Europe Conference and Exhibition,* pp. 1432-1437, 2017.

[144] M. Atzori, A. Gijsberts, I. Kuzborskij, S. Heynen, A.G.M Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo, "A Benchmark Database for Myoelectric Movement Classification," *Transactions on Neural Systems and Rehabilitation Engineering*, 2013.

[145] S.H. Gao, M.M. Cheng, K. Zhao, X.Y. Zhang, M.H. Yang, and P. Torr, "Res2net: A new multi-scale backbone architecture," *IEEE transactions on pattern analysis and machine intelligence*, vol. 43, no. 2, pp.652-662, 2019.

[146] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural net-works with noisy labels," *Advances in neural information processing systems*, pp.8778–8788, 2018.

[147] W. Liu, Y. Wen, Z. Yu, and M. Yang, "Large-margin softmax loss for convolutional neural networks," *International Conference on Machine Learning (ICML)*, vol. 2, p.7, 2016.

[148] G. Huang and F. Ma, "ConCAD: Contrastive Learning-Based Cross Attention for Sleep Apnea Detection," *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 68-84, 2021.

[149] S. Jeon, K. Hong, P. Lee, J. Lee,and H. Byun, "Feature stylization and domain-aware contrastive learning for domain generalization," *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 22-31, 2021.

[150] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, D. Krishnan, "Supervised contrastive learning," *Advances in Neural Information Processing Systems*, vol. 33, pp. 18661-18673, 2020.

[151] X.Jiang, et. al., "Open access dataset, toolbox and benchmark processing results of high-density surface electromyogram recordings," *Transactions on Neural Systems and Rehabilitation Engineering*, vol. 29, pp. 1035-1046, 2021.

[152] N. Malešević, et. al., "A database of high-density surface electromyogram signals comprising 65 isometric hand gestures. Scientific Data," *Scientific Data*, vol. 8, no. 1, pp. 1-10, 2021.

[153] Z. Liu, Y. Lin, Y. Cao, H. Hu, Y. Wei, Z. Zhang, S. Lin, and B. Guo, "Swin transformer: Hierarchical vision transformer using shifted windows," *IEEE/CVF International Conference on Computer Vision*, pp. 10012-10022, 2021.

[154] Z. Liu, H. Mao, C.Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A ConvNet for the 2020s," *arXiv preprint arXiv:2201.03545*, 2022.

[155] S. Mehta, and M. Rastegari, "Mobilevit: light-weight, general-purpose, and mobile-friendly vision transformer," *arXiv preprint arXiv:2110.02178*, 2021.

[156] A. v. d. Oord, Y. Li, and O. Vinyals, "Representation learning with contrastive predictive coding," *arXiv preprint arXiv:1807.03748*, 2018.

[157] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *In International conference on machine learning (PMLR)*, pp. 1597–1607, 2020.

[158] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," *In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 9729-9738, 2020.