# A Multi-Dataset Characterization of Window-based Hyper-parameters for Deep CNN-driven sEMG Pattern Recognition

Frank Kulwa#, *Student Member, IEEE*, Haoshi Zhang#, Oluwarotimi Williams Samuel*, *Senior Member, IEEE*, Mojisola Grace Asogbon, *Member, IEEE*, Erik Scheme, *Senior Member, IEEE*, Rami Khushaba, *Senior Member, IEEE*, Alistair A. McEwan, and Guanglin Li*, *Senior Member, IEEE*

***Abstract*— The control performance of myoelectric prostheses would not only depend on the feature extraction and classification algorithms but also on interactions of dynamic window-based hyper-parameters (WBHP) used to construct input signals. However, the relationship between these hyper-parameters and how they influence the performance of the convolutional neural networks (CNNs) during motor intent decoding has not been studied. Therefore, we investigated the impact of various combinations of WBHP (window length and overlap) employed for the construction of raw 2-dimensional (2D) surface electromyogram signals on the performance of CNNs when used for motion intent decoding. Moreover, we examined the relationship between the window length of the 2D sEMG and three commonly used CNN kernel sizes. To ensure high confidence in the findings, we implemented three CNNs which are variants of the existing models, and a newly proposed CNN model. Experimental analysis was conducted using three distinct benchmark databases, two from upper limb amputees and one from able-bodied subjects. The results demonstrate that the performance of the CNNs improved as the overlap between consecutively generated 2D signals increased, with 75% overlap yielding the optimal improvement by 12.62% accuracy and 39.60% F1-score compared to no overlap. Moreover, the CNNs performance was better for kernel size of seven than three and five across the databases. For the first time, we have established with multiple evidence that WBHP would substantially impact the decoding outcome and computational complexity of deep neural networks, and we anticipate that this may spur positive advancement in myoelectric control and related fields.**

***Index Terms*—Convolution neural network (CNN), Window length, Window overlap, sEMG signal, Prostheses, Upper Limb**

## I. INTRODUCTION

THE inability of amputees to carry out basic and complex arm related tasks reduces their quality of life which may be psychologically distressing. To restore limb functionality in amputees, prostheses driven by biosignals from which limb movement intentions can be decoded, have been widely embraced [1], [2], [3]. Such signals can be collected either in invasive (implantation of sensors within human body [4], [5], [6]) or noninvasive (placement of sensors on the skin surface human body [7], [8], [9]). Though different biosignals exist, surface electromyogram (sEMG) is considered the most appropriate signal source for the control of prostheses and other miniaturized robots (such as exoskeleton for stroke patient rehabilitation) due to its noninvasiveness, ease of acquisition, and capability to offer rich set of motor information [10]. The stochastic nature of sEMG motivated the use of engineered features for characterizing the underlying motion intent (MI) of amputees in traditional machine learning models [11], [12]. Besides, various studies have shown that raw sEMG signals are unsuitable for prosthetics control [8] and re-iterated the need for manually engineered features since they are often in low-dimensional space and can significantly reduce pattern recognition complexity. However, such engineered features are computed based on statistical approaches that sometimes fail to capture relationships across sEMG channels, leading to the loss of underlying motor information [7]. To address this issue, deep learning methods have been proposed for automatic feature learning and can be directly applied to raw 2-dimensional sEMG signals for adequate extraction of relevant features across channels [9][13].

The sEMG-based gesture classification task can be modeled as a 2D signal recognition problem using convolutional neural networks (CNNs), where the input signal to the CNNs has a size of J × W (length x width). Many techniques have been used to construct 2D signals from sEMG which serve as input to deep learning models. For instance, 2D signals have been constructed based on spectrograms obtained from short-time Fourier transforms [14], scalograms obtained from wavelet transforms [15], and construction from engineered features [16]. Other methods include HD-sEMG electrode array topology (where the 2D resembles an array of electrode

This work was supported in part by Key-Area Research and Development Program of Guangdong Province (2020B0909020004), the National Natural Science Foundation of China (#62150410439), Guangdong Basic and Applied Research Foundation (#2023A1515011478), the Science and Technology Program of Guangdong Province (#2022A0505090007), Ministry of Science and Technology, Shenzhen (#QN2022032013L), and the ANSO scholarship for young talented students, "The Belt and Road" Innovative Talent Exchange program for foreign experts (DL2022024002L) and Jinan 5150 Program for Talents Introduction.

F. Kulwa, H. Zhang, O.W. Samuel, M.G. Asogbon, and G. Li are with the CAS Key Laboratory of Human-Machine Intelligence-Synergy Systems and the Shenzhen Institute of Artificial Intelligence and Robotics for Society, Shenzhen Institute of Advanced Technology (SIAT), Chinese Academy of Sciences (CAS), Shenzhen, Guangdong 518055, China.

O.W. Samuel, M.G. Asogbon, and A.A. McEwan are with the School of Computing and Engineering, University of Derby, Derby, DE22 3AW, United Kingdom.

F. Kulwa and H. Zhang are also with the Shenzhen College of Advanced Technology, University of Chinese Academy of Sciences, Shenzhen, Guangdong, 518055.

E. Scheme is with Institute of Biomedical Engineering, University of New Brunswick, Fredericton, NB, Canada.

Rami Khushaba is with the Australian Centre for Field Robotics, Sydney University, Chippendale, NSW, 2008, Australia.

*# The first and second author contributed equally to the work.*

*Correspondence*: Dr. O.W. Samuel, e-mail: o.samuel@derby.ac.uk or samuel@siat.ac.cn and Prof. Dr. G. Li, e-mail: gl.li@siat.ac.cn.

placement during data collection) [17], and from raw sEMG signal segments using overlapping time windows [18] (where the signal width is equal to the number of electrodes and the length matches the segmentation window length). However, the use of spectrogram, scalogram and 2D from engineered features impose extra preprocessing time. While the raw EMG as input to CNNs represents a seamless and direct approach, several studies have leveraged this option towards attaining fairly decent classification performance [16] [18]. This partly motivated us to further explore the potential impact of 2D signals constructed from raw EMG recordings on CNNs for movement intent decoding, which may be necessary in developing time-efficient and intuitive deep learning driven prosthetic control schemes.

Recent research has confirmed that 2D signals' temporal and spatial information can be used by CNN to extract long and short-term patterns present in sEMG recordings [17], [19], [20]. With respect to sparse and high-density surface EMG recordings, previous studies have proposed CNN-based models driven by 2D signals of varied window length and overlap constructed from sEMG recordings. For instance, in developing a CNN model that adopts 2D signals for movement intent decoding, overlaps of 0ms, 90ms, 10ms, 10ms, and 30 data points were applied using window lengths of 150ms, 150ms, 100ms, 300ms, and 150 data points by Atzori et al. [21], Tsinganos et al. [18], Rahimian et al. [22], Gulati et al. [23], and Bakircioglu et al. [16], respectively. From these studies and other related works, it is unclear what combination of parameters in terms of window length and overlap would yield the requisite 2D signals for optimal movement intent decoding (in terms of accuracy and robustness) in CNNs-based pattern recognition schemes for multifunctional prostheses. Notably, previous studies have reiterated that variation in window-based hyper parameters would substantially impact MI decoding and computational complexity of traditional machine learning models that employ engineered features [24], [25], [26], and [27] without addressing deep neural networks. In other words, systematic investigations of the optimal combination of window-based hyper parameters for 2D signal construction in deep neural networks (CNNs) remains an open research question to the best of the authors' knowledge.

Convolution kernels act as the feature extraction phase for the CNN in which the network automatically learn low- and high-level features [28]. The size of the kernel has a great influence on the network's performance as small size kernels are good in mining fine features which could be missed by large size kernels and vice versa. Therefore, it is important to look at their influence in EMG decoding tasks.

In an attempt to fill the above-mentioned research gaps, this paper thoroughly investigated the impact of 2D signals constructed from raw sEMG recordings using varied combinations of windowing parameters on the decoding performance of CNNs. Besides, the CNN models employed are simple and low memory enabling practical deployment in upper limb prostheses. Also, we explored the effects of varying kernel sizes on the classification performance of different CNN models. The experimentation was carried out using sEMG from three different databases comprising amputees and able-bodied individuals that are mostly used for EMG gesture characterization. Moreover, the databases include both HD-sEMG and sparse multichannel sEMG signals and contain various degrees of freedom movements associated with the finger, hand, wrist, and elbow gestures, which ensures proper coverage. The four primary contributions of this work are;
(1) To provide insight on the deployment of novel and efficient deep learning driven prostheses control scheme, this study explored the impact of the raw 2D signals' window length and overlap on the performance of CNNs. (2) We explore the effects of receptive window for different configurations of CNNs and how they influence the feature learning on the raw 2D signal. (3) To aid the optimal selection of window parameters, we derived a rule of thumb on the relationship between kernel sizes and window parameters. (4) We proposed a novel variant of CNN model characterized by low computational complexity with simple yet robust, and accurate decoding performance.

## II. MATERIAL AND METHODS

### *A. Description of the EMG Databases Utilized*

In this study, we utilized three sEMG databases to validate our hypothesis and the performance of the CNNs as follows.

**Database 1 (DB1):** The DB1 is the CapgMyo database (DB-a) [29], and it contains High-density sEMG (HD-sEMG) data acquired at a sampling frequency of 1000 Hz from 18 able-bodied subjects. The acquisition device had 8 modules of matrix-type (8 × 2) differential electrode arrays with silver wet electrodes which were fixed around the right forearm. The first acquisition module was placed on the extensor digitorum communis muscle at the height of the radio-humeral joint and others were equally spaced clockwise. 8 isotonic and isometric hand gestures were involved, and were the subset of NinaPro database (DB3) [30]. There were 10 repetitions of each motion, each held for 3 to 10 seconds. The movements were interspersed with a rest state of around 7 seconds to avoid muscle fatigue, and only the preprocessed data which contains static part of the motion were used to ensure correct match with the labels. To handle the noisy segment, a second-order Butterworth band stop filter of 45–55 Hz was used while other observed artifacts were removed using a band-pass filter of 20-380 Hz.

**Database 2 (DB2):** This database was acquired in-house from four transhumeral amputees. The dataset was collected using the HD-sEMG system (Refa 128 model, TMS International, Netherlands) with 32 monopolar electrodes and an amplifier gain of 26.55. The electrodes were positioned on biceps brachii and triceps brachii of the individuals' residual limbs in a matrix of 4x8 as indicated in Fig. 1. Prior to the data collection, the subjects were given written informed consent and they provided permission for the publication of their data

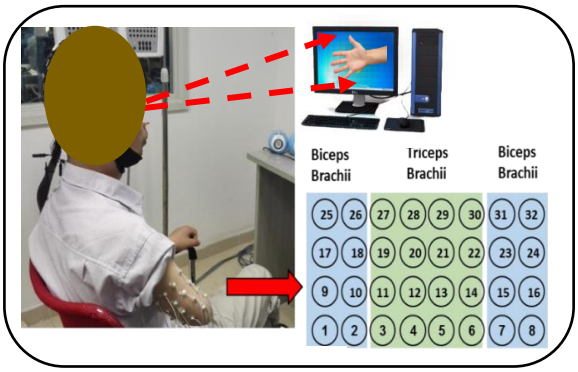



**Fig. 1.** Electrode location and experiment setup on the amputee during sEMG data collection for DB2.

for scientific and educational purposes. During data acquisition, a computer screen was placed in front of the participants and a picture of a particular gesture would be shown for the subject to follow (see Fig. 1). The 5 upper limb gestures involved were wrist supination (WS), wrist pronation (WP), no movement (NM), hand close (HC), and hand open (HO)). Meanwhile, each motion was held for about 5s with a 5s rest between two consecutive tasks. Each gesture was repeated 5 times. The data was sampled at 1024Hz while 50Hz notch filter and band pass filter with cut-off frequencies 10-500Hz were applied to extract requisite signal components. The experiment was approved by the Institutional Review Board of Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences.

**Database 3 (DB3):** The third database denoted as DB3 is a publicly available and known as NinaPro Database 3 [30] and it contains EMG recordings of 11 transradial amputees. The sparse multi-channel sEMG signals were recorded via 12 electrode channels positioned on the residual arm muscles of the amputee. The electrodes were located around the residual arm. We utilized exercise B which contains 17 classes of movements comprising 9 basic wrist movements and 8 isotonic isometric hand gestures. Each movement gesture lasted 5 seconds with 6 repetitions, followed by 3 seconds of rest. We directly worked with the preprocessed data which was sampled at 2000Hz, filtered with a notch filter of 50Hz and a band pass filter of 20-450 Hz [31]. Table 1. Summarizes the difference in the parameters of the three databases used in this study.

Table 1: The parameters of databases deployed in this study

| Database type | Subjects | Type of subjects | No. of gestures | Type of gestures | No. of electrodes |
|---|---|---|---|---|---|
| DB1 [29] | 18 | Able-bodied | 8 | Hand and finger | 128 |
| DB2 [in-house] | 4 | Amputee | 5 | Hand | 32 |
| DB3 [30] | 11 | Amputee | 17 | Hand, finger, and wrist | 12 |

### *B. 2D Signal Construction and Windowing Technique*

The 2D signals were constructed using a segmentation window of dimension $\boldsymbol{J \times W}$, where $\boldsymbol{W}$ is the window length and $\boldsymbol{J}$ is the number of electrodes corresponding to 128, 32, and 12 for DB1, DB2, and DB3, respectively. Precisely, a combination of various windowing parameters including window lengths (25ms, 50ms, 75ms, 100ms, 125ms, 150ms, 175ms, and 200ms) and different overlaps (0% of W which denotes no overlap, 25% of W, 50% of W, and 75% of W) were employed in constructing varied configuration of 2D signals. The values of the windowing parameters were opted based on previous recommendations [24]. It is worth noting that, in this paper we denote the overlap as $\boldsymbol{V}$. For instance, the overlap of 25% of W is denoted as 0.25V, 50% of W is 0.50V etc. A description of the constructed 2D signals based on combinations of windowing parameters is indicated in Fig. 2.

Subsequently, the 2D signals were partitioned into training and test sets. Where a 5-fold cross validation protocol was followed. And each time, 80% of the data was used for training while the unseen 20% was used for testing to avoid data leakage (a key challenge in building machine learning models). Before training, a Z-score normalization method was applied to the training set to ease the CNN's learning process [32]. For the testing set normalization, the mean value to subtract and the standard deviation to divide as obtained from the training set were utilized for the normalization process of the test set. It is noteworthy that during the preliminary experiment, the Min-Max [33] and Z-score normalization approaches were examined. Our choice on using Z-score normalization is based on its consistent excellent movement intent decoding results. Additionally, it is important to note that data preprocessing and the subsequent experiments were carried out separately for every subject.

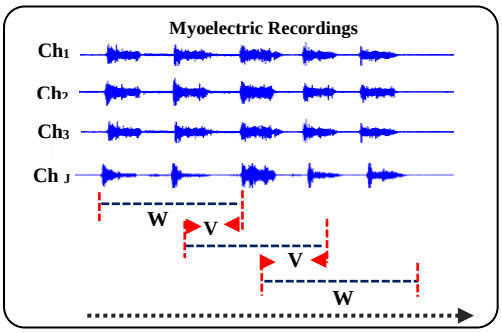



**Fig. 2.** The framework for the construction of 2D signals. The relationship between window length (W) and overlap (V) of any successive data segment exploited for constructing 2D sEMG representation of size JxW is depicted in the figure. Where J is the number of channels i.e., Channel 1 (Ch1), Channel2 (Ch2) up to Ch(J).

### *C. Convolution Neural Networks (CNNs)*

This study built four different CNN models to investigate the impact of varied configurations of 2D sEMG. Among the four CNNs, Model1 has similar architecture to the model implemented in our previous work [34]. Meanwhile, Model2 and Model3 are built as variants of existing models and the fourth (Model4) is entirely proposed by the authors. The architectures of the models are:

***Model1****:* Model1 is built based on the CNN model implemented in our preliminary work [34]. Our decision for considering this network is due to its simple structure and stable performance. In addition, it takes as input a shape that aligns with the shape of our 2D sEMG representation (J x W).

***Model2****:* Inspired by the simplicity of the model proposed by Bakırcıoğlu et al. [16], we adopted its feature learning convolution blocks in building our own CNN variant and replaced the flattening layer in Bakırcıoğlu et al.'s network with a global-max pooling which comparably reduces the network parameters (to enhance its computational cost) and is not prone to overfitting [35]. Thus, the *Model2* consists of three convolution blocks each having 80, 100, and 120 nodes with similar values of kernel sizes, respectively. Succeeding each convolution block is a batch normalization and a max-pooling layers that allows down sampling of the feature maps by half. The lower part of the network consists of the Global max-pooling layer for feature reduction and a dense layer of 512 nodes for high-level feature learning.

***Model3:*** It is an improved version of the network proposed by Maufroy et al. [36] that comprised two convolution layers and a fully connected layer each having 100 nodes designed specifically for 1D EMG signal processing. Thus, we came up with a variant of the network by replacing the 1D convolution layers with 2D convolution layers and discarded the dropout layers after each convolution except after the fully connected layer. Each convolution is followed by a batch normalization and max-pooling layer of 2x2. Similar to Model2, feature

reduction is achieved by using the Global max-pooling layer. Because the networks are trained from scratch, we have applied weights initialization in both Model2 and Model3 to allow quick convergence and optimization of the networks [37]. It should be noted that the classification in both networks (Model2 and Model3) is performed using a softmax layer.

***Model4 (Multi-scale CNN)*** abbreviated as ***MS-CNN:*** The convolution kernels can be treated as view windows of the network, large size kernels meaning that the network can view 2D signal at a large dimension (which includes both spatial information across channels, and temporal information within channels) at once and learn large grained features, likewise, small kernels can model fine features of the signal [38]. Therefore, in this study, we proposed a MS-CNN model which combines the advantage of small and large size kernels to improve the classification performance by learning both fine and large-grained patterns from the signal while keeping the network parameters as low as possible. Moreover, inspired by the dilated convolution kernel concept [39], we have utilized them as core layers to achieve the multi-scale task. The architecture of the proposed network herein referred to as *Model4* is presented in Fig. 3.

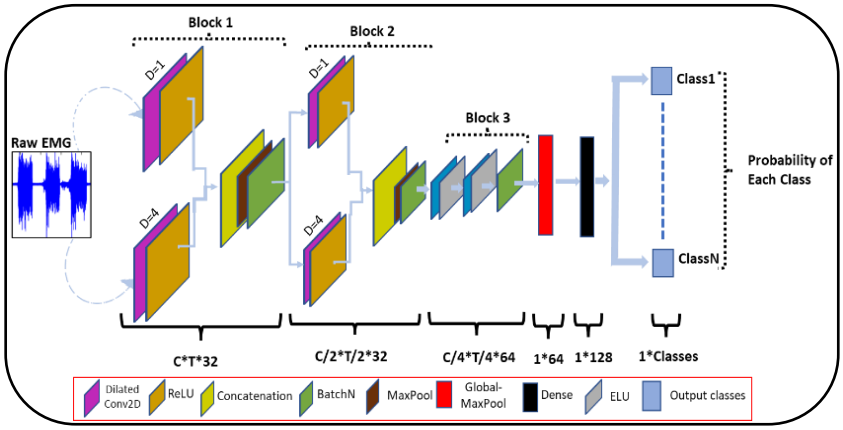



**Fig. 3.** The proposed Multi-scale Convolution neural network (MS-CNN). D represents the dilation rate.

The essential functional layers in the MS-CNN (in Fig. 3) are the multi-scale block 1 and block 2 composed of dilated convolution layers at different scales (dilation rates) to enable an exponential increase of large receptive fields without loss of output resolution [40]. The input to the network is fed to block1 via two means including the upper and lower paths with each having its own scales. Meanwhile, the upper path contains convolution with kernel size=3 and dilation =1 to model fine features. The lower path contains convolution with kernel size=3 and dilation=4 which is equivalent to the conventional kernel of size =9 with output resolution equivalent to 3x3 convolution. Such a large kernel can learn coarse (general) features from the input. The outputs of the two paths are combined by the concatenation layer without dimension reshaping while max-pooling is applied to extract the most essential joint features. The second Multi-scale block (Block 2) has a similar operation to Block 1.

The subsequent Block 3 contains two point-wise 1x1 convolutions. These convolutions enable the network to learn more about the joint features from the multi-scale blocks while keeping the number of network parameters low [41]. The lower part of the network incorporates a global max-pooling layer, one fully connected layer of 128 nodes, and the classification layer that employs a softmax function. Additionally, all the weights were initialized using a random uniform initializer. The key operations of the proposed *Model4* are described in subsection (i) and (ii) below;

*i. Dilated Convolution Layer for Multi-scale and Memory Efficiency*

Dilated convolution layer is a layer that uses larger 2D filters to extend the standard convolution layer. Fig. 4 depicts the conceptual representation of a typical dilation process in the context of the proposed MS-CNN. In order to optimize the number of network parameters, traditional CNNs exploit convolution with small filters ($2 \times 2$ or $3 \times 3$). However, dilated convolution filters are an easy approach to generating a CNN with large filters such as $5 \times 5, 7 \times 7$, etc. while keeping the parameters similar to $2 \times 2$ or $3 \times 3$ kernels. A dilated convolution with an enlargement rate of $d$ places $d-1$ zeros in between successive filter values, which results in enlarging the size of a $L$ filter to $[L+(L-1)(d-1)(d-1)]$. It should be noted that the quantity of nonzero parameters remains equal as before. In addition, the dilated filters significantly expand CNN's receptive field, allowing it to capture more contextual and spatial information [42]. As a result, it enables flexible multi-scale contextual information aggregation while maintaining the same resolution.

*ii. Handling Internal Covariant Shift within the MS-CNN*

A major challenge of non-stationary signals such as the sEMG is the lack of a common distribution which sometimes result in internal covariant shift within the network [43]. To resolve this issue, batch normalization layers were introduced after each convolution block, and the ELU activation function was leveraged afterwards in requisite layers. Thus, the ELU activation function is expressed mathematically as shown in equation. (1):

$$f_{ELU}(x_k) = \begin{cases} x_k & (x_k > 0) \\ \propto (\exp(x_k) - 1) & (x_k \leq 0) \end{cases} \quad (1)$$

Where, $x_k$ is the filter output in the $k^{th}$ channel and $\propto$ represents a parameter to control the value to which an ELU saturates for negative inputs. In this study, the default value $\propto = 1$ was considered. For negative inputs, the ELU activation function returns negative values, as a result its mean (average) outputs approach zero than that of ReLU. Hence the ELU activation function can improve network learning [44]. Thus, in the *MS-CNN*, each point-wise convolution is followed by an ELU activation function.

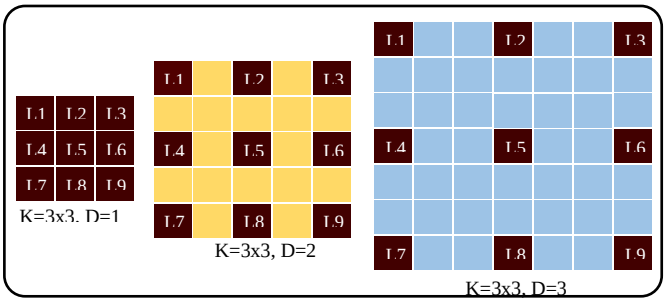



**Fig. 4.** The conceptualization of dilation: The convolution kernel (L) = 3x3 and its dilated filters when dilation rate (D) = 1, 2 and, 3.

### *D. Experimental Setups*

Since the Adam Optimizer [45] demonstrated the best network performance (in our preliminary/pilot experiments of finding the optimal hyper-parameters for the models) compared to the stochastic gradient descent (SGD) optimizer [46], all the described models were trained using Adam Optimizer with a learning rate of 0.0001. It should also be noted that at least one subject from each DB was used to get the optimal hyper-parameters for the models during preliminary experiments. Besides, due to its superior performance in multiclass tasks, the categorical cross-entropy [47] was utilized as the loss function. The number of epochs for each model was set depending on the characteristics of the dataset in each database as indicated in Table 2. Thus, the number of iterations/epochs was varied in every model separately to allow optimal convergence while leaving out only the effects of the studied parameters (overlap, window length and kernel sizes) to be observed.

**Table 2.** The number of epochs for each model and database. DB1 denotes database1, DB2 denotes Database 11 and DB3 denotes database 3.

| SN | Database | Model1 (Epochs) | Model2 (Epochs) | Model3 (Epochs) | Model4 (Epochs) |
|---|---|---|---|---|---|
| 1 | DB1 | 35 | 20 | 30 | 30 |
| 2 | DB2 | 30 | 20 | 30 | 30 |
| 3 | DB3 | 30 | 30 | 38 | 30 |

To analyze the impact of kernel size on the performance of the CNNs per time, the same size of the kernel was applied to all convolution layers of the models. Hence, we utilized kernel sizes of 3x3, 5x5, and 7x7 since they are the basic kernels that have been employed in prior studies and yielded relatively good performances [14], [16], [18], [21], [22]. It should be noted that *Model4*, utilizes kernels with resolution equivalent to 3. Thus, it was not included in experiments involving the variation of kernels 5x5 and 7x7.

**Table 3**. Experimental analyses performed for the three parameters for different models (The parameters were the same for every subject)

| SN | CNN | Window Length | Overlap Size | Kernel Size | Total Analyses |
|---|---|---|---|---|---|
| 1 | Model 1 | 8 | 4 | 3 | 96 |
| 2 | Model 2 | 8 | 4 | 3 | 96 |
| 3 | Model 3 | 8 | 4 | 3 | 96 |
| 4 | Model 4 | 8 | 4 | 1 | 32 |
| | **Total Experimental Analyses** | | | | **320** |

*Evaluation Measures*

To analyze the performance of the different networks, we used benchmark evaluation metrics that include accuracy and F1-score, as they have been widely adopted in the field of myoelectric pattern recognition. Additionally, to verify the statistical significance of the proposed method, a non-parametric Friedman's test was conducted followed by Dunn's post hoc test. The definitions for the metrics are indicated in equations 2 and 3.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \quad (2)$$

$$F1-score = \frac{2TP}{2TP+FP+FN} \quad (3)$$

From equations 2 and 3, true negative (TN) and true positive (TP) are the number of accurately classified negative and positive samples, respectively. The number of positive cases classified as negative is a false negative (FN), while the number of negative samples predicted as positive is false positive (FP).

## III: RESULTS AND ANALYSIS

### *A. Analyzing the impact of window length and overlap on the performance of the CNNs.*

In this section, we investigated whether or not the 2D signals constructed from varied combinations of window length (*W*) and overlap (*V*) would impact the decoding performance of CNNs. For each experiment, the *W* was kept constant while varying the *V* and vice versa. And the corresponding signals were used to train the built CNNs described in **Section II.C**. For instance, 2D signals obtained from *W* of 175ms and an overlap of "75% of *W*" were employed to train and test a CNN with a kernel size of 3. Similarly, the impact of other combinations of *W* and *V* across kernel sizes on the CNNs was examined. Meanwhile, the experiments were carried out for each of the databases described in **Section II.A** and the obtained results are presented as follows. Table 3 summarizes the overall analyses per subject. Because the results for all three kernels (3, 5, and 7) show a similar trend, we only presented the results for kernel size of 3. The average motion classification performance for every combination of window parameters (W vs V) across kernel sizes and subjects for the different CNNs for Database 1 (DB1) are presented in Fig 5 in accuracy and Table 4 in F1-score.

Analyzing the results in (Fig. 5 (a)), it can be seen that the accuracy in every CNN improves when the percentage of overlap (V) increases at constant window length (W) and kernel size (K). For instance, the accuracy of Model1 at W = 25ms increases from 97.22%, 97.79%, 98.32%, to 98.74% when V increases from 0V, 0.25V, 0.50V, to 0.75V, respectively. This indicates a significant improvement upon statistical analysis results when comparing overlap 0V and 0.75V (*p-value: 7x10$^{-6}$*). When investigating the effects of varying W at constant V in a particular model, the results show no considerable difference. For instance, in Model1, the performances at the overlap of 0.75 are 98.74%, 98.88%, 98.84%, 98.86%, 98.84%, 98.74%, 98.79%, and 98.91% at W of 25ms, 50ms, 75ms, 100ms, 125ms, 150ms, 175ms, and 200ms, respectively. To further examine this phenomenon, the F1-score results in Table 4 also show that the model's performance improved by increasing the percentage of overlap at constant window length. Similarly, the average results displayed in Fig. 5 (b) (accuracy) and Table 4 (F1-score), show that the overlap of 75% (0.75V) achieved much better classification results for upper limb MI decoding irrespective of the window length.

To further validate the above results, we conducted the experiments using amputee databases (D2 and DB3) whose results are presented in Fig 6 and 7 in accuracy, and Table 5 and 6 in F1-score, respectively. The accuracy results in Fig. 6 and 7 (for DB2 and DB3 respectively), show that the performance of all the networks increases with an increase in overlap at a fixed window size per time with statistical significance of *p=4.0x10$^{-6}$, 3.0x10$^{-6}$, 3.0x10$^{-6}$, 7.9x10$^{-5}$* for model1, model2, model3, and model4 in DB2. While *p=3.62x10$^{-4}$, 2.4x10$^{-5}$, 2.8 x10$^{-5}$, 8.26x10$^{-3}$* were observed in DB3 for model1, model2, model3, and model4 respectively, when comparing the results of overlap 0v and 0.75. Also, the F1-score values in Table 5 and 6 depict similar trend, further corroborating the phenomenon observed for the accuracy metrics.

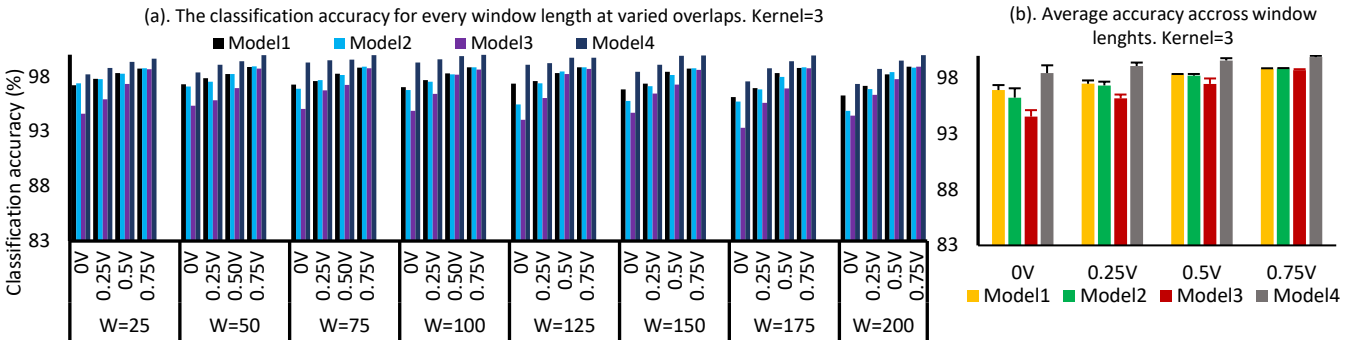



**Fig. 5.** Classification accuracy of Model1, Model2, Model3, and Model4, when they are applied to DB1. (a) Each sub-part in the graphs contains results of all models at the same window length and varied overlaps 0V, 0.25V, 0.50V and 0.75V which are equivalent to 75% of W, 50% of W, 25%of W and 0% of W. **Note: Window length (W) unit is in millisecond (ms). (b)** The average classification results across window lengths.

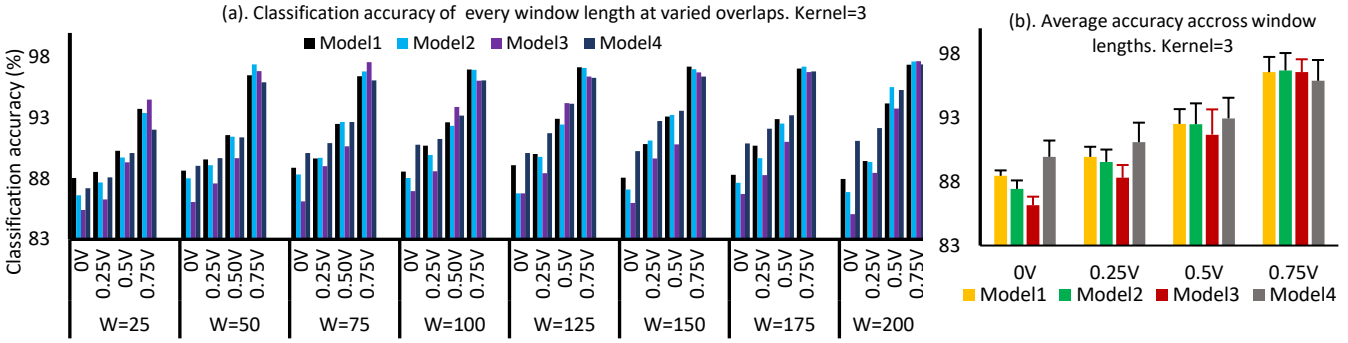



**Fig. 6.** Classification accuracy of Model1, Model2, Model3 and Model4, when they are applied to DB2. (a) Each sub-part in the graphs contains results of all models at the same window length and varied overlaps 0V, 0.25V, 0.50V and 0.75V which are equivalent to 75% of W, 50% of W, 25%of W and 0% of W. **Note: Window length (W) unit is in millisecond (ms). (b)** The average classification results across window lengths.

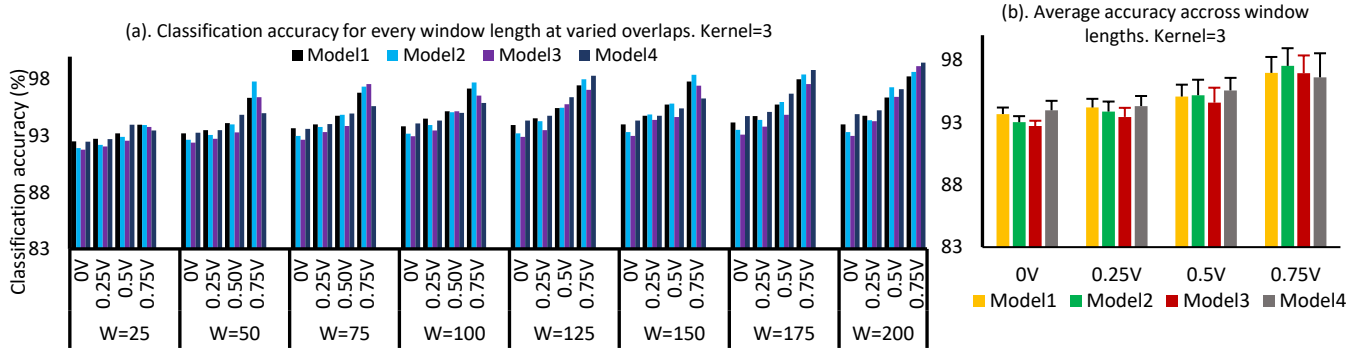



**Fig. 7.** Classification (in accuracy) of Model1, Model2, Model3, and Model4, when they are applied to DB3. (a) Each sub-part in the graphs contains results of all models at the same window length and varied overlaps 0V, 0.25V, 0.50V and 0.75V which are equivalent to 75% of W, 50% of W, 25%of W and 0% of W. **Note: Window length (W) unit is in millisecond (ms). (b)** The average classification results across window lengths.

**Table 4** Classification F1-score of Model1, Model2, Model3, and Model4 at kernel size=3 when applied on DB1. Each part contains classification at different window lengths (W) with varied overlaps (V). **Note: Window length (W) unit is in millisecond (ms)).**

| **(DB1)** | **W=25** | | | | **W=50** | | | | **W=75** | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.50V | 0.75V |
| Model1 | 88.41 | 90.92 | 94.18 | 97.72 | 87.69 | 90.00 | 94.04 | 98.70 | 87.67 | 91.26 | 93.39 | 97.71 |
| Model2 | 89.96 | 92.48 | 95.96 | 98.93 | 89.98 | 93.72 | 95.72 | 99.19 | 88.72 | 94.12 | 95.39 | 99.08 |
| Model3 | 80.86 | 86.31 | 90.99 | 97.61 | 83.47 | 85.59 | 91.45 | 98.60 | 79.20 | 88.64 | 91.22 | 98.90 |
| Model4 | 92.60 | 95.04 | 97.38 | 98.54 | 93.31 | 96.18 | 97.44 | 99.85 | 96.79 | 97.88 | 98.22 | 99.79 |
| | **W=100** | | | | **W=125** | | | | **W=150** | | | |
| | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| Model1 | 88.22 | 89.77 | 93.94 | 97.70 | 86.29 | 90.12 | 94.59 | 97.48 | 82.67 | 86.22 | 93.32 | 98.24 |
| Model2 | 88.09 | 91.95 | 95.35 | 98.59 | 88.22 | 90.61 | 96.35 | 98.19 | 85.30 | 89.83 | 93.96 | 97.66 |
| Model3 | 81.78 | 86.44 | 94.61 | 98.19 | 80.52 | 85.33 | 95.37 | 97.83 | 79.06 | 88.63 | 88.99 | 98.13 |
| Model4 | 96.94 | 98.10 | 99.47 | 99.91 | 96.34 | 96.85 | 98.79 | 98.89 | 93.31 | 95.89 | 99.55 | 99.76 |
| | **W=175** | | | | **W=200** | | | | **Mean±Std** | | | |
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| Model1 | 84.80 | 84.89 | 92.71 | 98.00 | 83.11 | 85.69 | 92.48 | 97.68 | 86.10±2.31 | 88.60±2.56 | 93.57±0.73 | 97.90±0.40 |
| Model2 | 85.22 | 88.89 | 94.49 | 98.22 | 82.04 | 87.73 | 96.90 | 98.35 | 87.19±2.77 | 91.16±2.29 | 95.51±0.95 | 98.52±0.52 |
| Model3 | 77.59 | 82.77 | 89.87 | 98.50 | 78.47 | 86.23 | 94.84 | 98.92 | 80.11±1.92 | 86.24±1.88 | 92.16±2.44 | 98.33±0.48 |
| Model4 | 94.92 | 94.95 | 97.52 | 99.68 | 92.67 | 94.85 | 97.43 | 99.89 | 94.60±1.87 | 96.21±1.29 | 98.22±0.93 | 99.53±0.52 |

**Table 5.** Classification F1-score of Model1, Model2, Model3, and Model4 at kernel size=3 when applied on DB2.

| (DB2) | W=25 | | | | W=50 | | | | W=75 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.50V | 0.75V |
| Model1 | 69.83 | 72.27 | 75.95 | 82.99 | 70.59 | 74.19 | 78.21 | 90.56 | 70.48 | 74.02 | 79.94 | 88.69 |
| Model2 | 67.45 | 70.16 | 77.52 | 84.75 | 69.84 | 72.59 | 78.84 | 93.08 | 69.57 | 76.29 | 80.78 | 92.47 |
| Model3 | 63.62 | 66.77 | 72.54 | 83.94 | 64.79 | 68.13 | 71.63 | 90.48 | 65.23 | 72.05 | 72.82 | 90.56 |
| Model4 | 67.59 | 70.22 | 75.29 | 80.13 | 72.60 | 74.04 | 78.19 | 89.97 | 75.03 | 77.19 | 81.70 | 90.35 |
| | W=100 | | | | W=125 | | | | W=150 | | | |
| | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| Model1 | 71.81 | 74.40 | 78.52 | 89.08 | 70.35 | 73.85 | 78.37 | 89.72 | 67.97 | 74.47 | 78.61 | 87.57 |
| Model2 | 68.59 | 74.46 | 80.33 | 91.62 | 68.63 | 73.83 | 79.59 | 91.96 | 66.90 | 77.80 | 80.87 | 92.55 |
| Model3 | 65.98 | 68.82 | 82.27 | 86.98 | 65.86 | 70.69 | 84.09 | 88.20 | 64.08 | 73.38 | 74.99 | 88.42 |
| Model4 | 76.75 | 78.12 | 83.04 | 90.25 | 75.23 | 78.63 | 85.40 | 90.73 | 75.60 | 81.53 | 83.91 | 91.01 |
| | W=175 | | | | W=200 | | | | Mean±Std | | | |
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| Model1 | 66.89 | 73.20 | 78.43 | 90.10 | 67.60 | 71.80 | 81.96 | 89.69 | 69.44±1.73 | 73.52±1.01 | 78.74±1.70 | 88.54±2.43 |
| Model2 | 65.24 | 71.99 | 80.71 | 92.81 | 66.59 | 70.39 | 91.23 | 94.07 | 67.85±1.58 | 73.43±2.70 | 81.23±4.20 | 91.66±2.89 |
| Model3 | 63.51 | 67.89 | 71.83 | 89.36 | 64.33 | 72.87 | 83.63 | 96.03 | 64.67±0.95 | 70.07±2.51 | 76.72±5.58 | 89.24±3.47 |
| Model4 | 76.92 | 80.36 | 82.91 | 91.99 | 77.46 | 80.81 | 88.20 | 93.54 | 74.64±3.23 | 77.61±3.82 | 82.32±4.04 | 89.74±4.06 |

T**able 6. C**lassification F1-score of Model1, Model2, Model3, and Model4 at kernel size=3 when applied on DB3.

| (DB3) | W=25 | | | | W=50 | | | | W=75 | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.50V | 0.75V |
| **Model1** | 37.96 | 39.97 | 42.47 | 47.48 | 43.27 | 46.31 | 49.56 | 57.81 | 45.65 | 48.88 | 53.21 | 62.42 |
| **Model2** | 34.45 | 36.95 | 43.25 | 49.69 | 40.50 | 43.79 | 49.67 | 74.86 | 42.68 | 49.64 | 55.09 | 73.08 |
| **Model3** | 28.41 | 30.81 | 33.28 | 38.33 | 32.11 | 34.69 | 37.47 | 46.25 | 33.69 | 38.35 | 40.20 | 58.74 |
| **Model4** | 35.42 | 37.48 | 40.24 | 44.74 | 42.47 | 44.58 | 47.63 | 57.07 | 45.71 | 48.95 | 53.26 | 62.56 |
| | **W=100** | | | | **W=125** | | | | **W=150** | | | |
| | 0V | 0.25V | 0.50V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| **Model1** | 47.06 | 51.18 | 55.28 | 65.42 | 46.74 | 52.19 | 57.61 | 67.09 | 47.01 | 53.13 | 58.23 | 69.29 |
| **Model2** | 44.78 | 50.41 | 57.11 | 74.68 | 43.77 | 52.35 | 59.51 | 77.33 | 46.06 | 57.34 | 62.11 | 80.03 |
| **Model3** | 36.09 | 38.92 | 45.74 | 51.90 | 36.22 | 38.84 | 48.63 | 54.35 | 37.06 | 42.09 | 45.95 | 57.95 |
| **Model4** | 49.16 | 51.41 | 57.46 | 64.77 | 51.45 | 55.11 | 60.99 | 68.28 | 51.45 | 55.11 | 60.99 | 68.28 |
| | **W=175** | | | | **W=200** | | | | **Mean±Std** | | | |
| | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V | 0V | 0.25V | 0.5V | 0.75V |
| **Model1** | 47.79 | 52.19 | 58.22 | 71.32 | 45.25 | 50.80 | 60.55 | 72.77 | 45.09±3.21 | 49.33±4.36 | 54.39±5.92 | 64.20±8.31 |
| **Model2** | 46.68 | 53.05 | 63.49 | 81.80 | 45.28 | 53.54 | 74.28 | 84.88 | 43.02±3.99 | 49.63±6.43 | 58.06±9.34 | 74.54±10.81 |
| **Model3** | 39.24 | 40.76 | 46.01 | 58.73 | 36.40 | 43.10 | 52.70 | 63.53 | 34.89±3.39 | 38.44±4.02 | 43.74±6.32 | 53.72±8.11 |
| **Model4** | 55.00 | 57.70 | 63.49 | 72.41 | 54.74 | 58.85 | 66.32 | 77.87 | 48.17±6.69 | 51.14±7.26 | 56.29±8.78 | 64.49±10.14 |

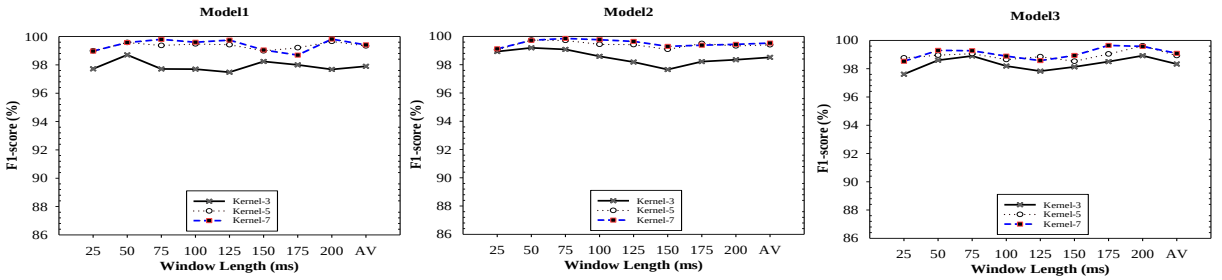
### *B. Analyzing the Effect of Network Receptive Fields against the Generated 2D sEMG Signals.*

The effect of each model's kernel size on different 2D sEMG signals constructed using varied windowing parameters was investigated in this section. The experiments were carried out by varying kernel sizes (3, 5, and 7) while keeping each window length constant and the networks were configured to utilize a specific kernel size per training phase. Due to the relatively stable and higher performances observed when the overlap was 75%, only this overlap was reported in this section for brevity. Moreover, due to the architecture of Model4 which only incorporates kernels with a resolution equivalent to 3, it was excluded from the analysis in the section for a fair comparison. Thus, the F1-score results for three models at different kernel sizes when 2D signals were generated from DB1, DB2, and DB3 are presented in Fig. 8.

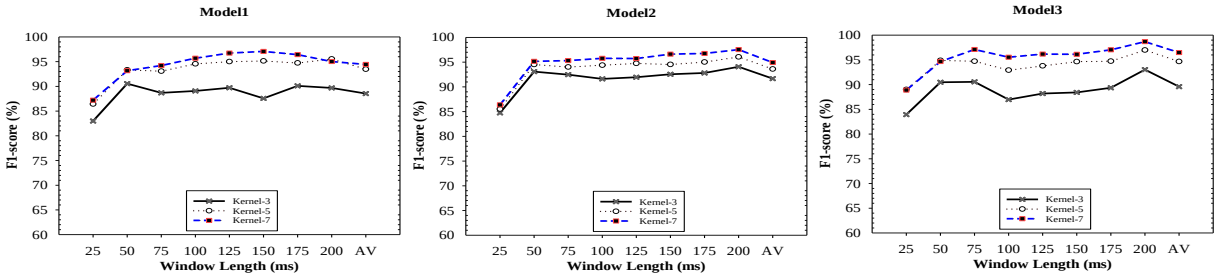
Based on analysis of the results for DB1 presented in Fig. 8(a), in general, the kernel size of 7 yields better classification performance across the networks (Model1, Model2, and Model3) as the window length increases from 25ms to 200ms. Also, the trend of increment for kernel sizes of 3, 5, and 7 can be observed to be similar in all the models except at window 175 in Model1 and Model2, and window 125 in Model3, where kernel 5 recorded an average F1-score value that is better than that of kernel 7. The average performance shows that kernel size of 3 led to the least classification results in all the CNNs. Besides, the statistical analysis between kernel 3 and 7 show a significant improvement of $p=0.9.92x10^{-4}$, $1.30x10^{-3}$, and $1.83x10^{-2}$. Although the impact is realized when varying kernel sizes, there is no significant and consistent (increase) effect when varying window length (at constant kernel) in the network's performance in this database (DB1) except from window 25 to 50ms for Model1, and from 25 to 75ms for Model2 and Model3.

By closely analyzing the variation of kernel sizes and window lengths for DB2 in Fig. 8(b), an increase in the performance of every model is realized when the size of kernels increases from 3, 5 to 7. The observation in this database shows that the use of large-size kernels improves the CNN's performance significantly (with $p<0.05$ in all models) regardless of the window length. Moreover, the variation of window lengths shows an impact in the performance of CNNs especially in Model2 and Model3 where the graphs show a positive gradient with the increase in window lengths for all window sizes (from 25ms to 200ms) in Model2, and from 100ms to 200ms in Model2 at constant kernel size.

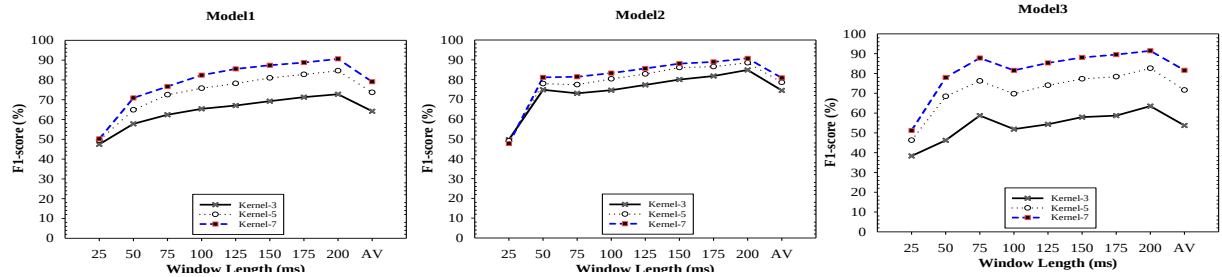
From the results illustrated in Fig. 8(c) for DB3, the performance of every model increases with the increase in kernel size at constant window length. Although the database contains signals from amputees, kernel=7 has achieved good performance. Comparing the results of kernels 7 and 3 a




(a) Variation of kernel sizes and window lengths in DB1




(b) Variation of kernel sizes and window lengths in DB2




(c) Variation of kernel sizes and window lengths in DB3

**Fig. 8**. The classification performance of Model1, Model2, and Model3 while varying the network kernels (3, 5 and, 7) at different 2D signal lengths (W) of 25, 50, 75, 100, 125, 150, 175, and 200 of DB1, DB2, and DB3. AV is the average results across all window lengths. **Note: Window length (W) unit is in millisecond (ms).**

significant increase in the performance of the models was observed for DB3 with $p<0.05$ in Model 1 and Model3. In that respect the maximum improvement of 18.46% is observed in Model1 with a window length of 125ms, while 8.35% is observed in Model2 with a window of 75ms, and 31.75% in Model3 with a window of 50ms. Contrary to previous databases (DB1 and DB2), the effects of increasing window length at constant kernel significantly increase the performance of the networks when DB3 was utilized. For example, the results in Model1 with a kernel of 3 improved from 47.48%, 57.81%, 62.42%, 65.42%, 67.09%, 69.29%, 71.32%, to 72.77% F1-score at window lengths of 25, 50, 75, 100, 125, 150, 175, and 200ms, respectively. This observation correlates with the previous study, which shows that the impact of window length is more evident in a small number of channels [26].

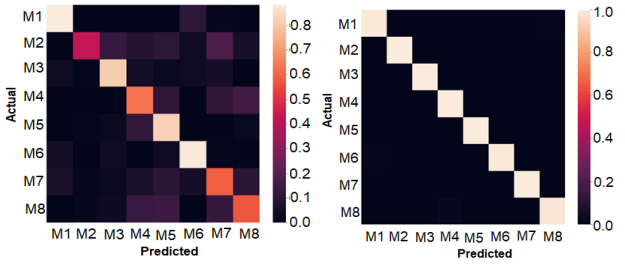
### *C. Analysis of Individual Limb Motion Intention (MI) across Window Overlaps and Kernel Sizes*

The analysis in the previous sections only presents an overview of motion classification outcomes with respect to varied combinations of window lengths, overlaps (V), and kernel sizes (K). Thus, it is necessary to look into the effects of these parameters on the classification performance of individual classes of movement. Hence, for each of the studied databases, the impact of kernel size and overlap on the categorization of a certain class of motion is described in this section. It is noteworthy that the results of W=200 were considered due to its high performances (as indicated in section III-B) and only Model1 was presented because the results in other models follow the same trend. Additionally, the confusion matrices for the average results across all subjects are presented for the lowest performing combination (Kernel=3 and overlap of 0V) and highest performing combination K=7 and overlap=0.75V. The impacts of window parameters on each MI classification were examined in DB1, DB2 and DB3 and the results are presented in Fig. 9.

Considering the analysis in Fig. 9 (a) for DB1, it can be seen that most of the individual gestures have attained results below 80.00% at the lowest combination (K=3 and overlap of 0%), however, the application of kernel=7 and overlap of 75% led to significant improvement in MI classification performance (above 98.00%) in all gestures.

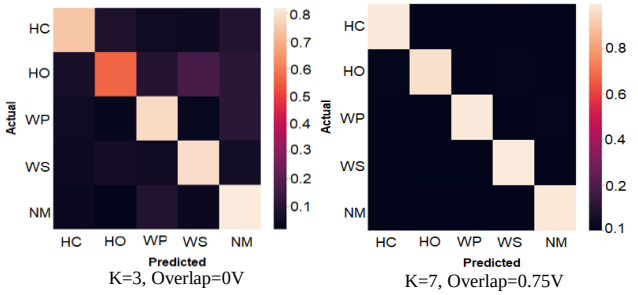
Likewise, the investigation across individual limb motion intention (MI) task in the amputee database (DB2) in Fig. 9 (b) shows that the classification performances of all gestures were improved from below 74.00% to above 88.00% for all gestures when the lowest combination was replaced by the highest combination.

The same trend is observed in Fig. 9 (c) for DB3, where the application of 75% overlap and kernel of 7 has achieved performances above 86.00%. This demonstrates that networks at kernel =7 and the 2D sEMG signals with a 75% overlap would be a perfect combination for characterizing sEMG signal patterns associated with these classes of hand gestures.



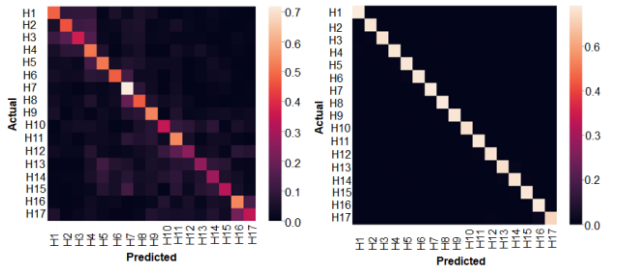

K=3, Overlap=0V K=7, Overlap=0.75V

**(a).** Classification performance for every motion intent in DB1. M1 – M8 represent the 8 motion intents in DB1.




K=3, Overlap=0V K=7, Overlap=0.75V

**(b).** Classification performance for every motion intent in DB2. Where, the output motions in DB2 are, wrist supination (WS), wrist pronation (WP), no movement (NM), hand close (HC), and hand open (HO).




K=3, Overlap=0V K=7, Overlap=0.75V

**(c).** Classification performance of Model1 for every motion in DB3. H1 – H17 represents the 17 gestures in DB3.

**Fig. 9.** The confusion matrices for Model1 on classification performance across each motion intention in DB1, DB2, and DB3

**Table 7.** Delay time (in seconds) of a single trial of movement on DB1. V is the overlap, K is the kernel size, and W is the window size

| DB1 | | W=25 | | | W=50 | | | W=75 | | | W=100 | | | W=125 | | | W=150 | | | W=175 | | | W=200 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V/K | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 |
| Model1 | 0.75V | 162 | 120 | 120 | 226 | 221 | 223 | 200 | 214 | 226 | 122 | 119 | 120 | 149 | 148 | 148 | 169 | 169 | 172 | 188 | 190 | 191 | 210 | 213 | 208 |
| | 0.50V | 94 | 58 | 56 | 109 | 102 | 103 | 107 | 105 | 107 | 103 | 97 | 99 | 115 | 115 | 115 | 135 | 136 | 136 | 153 | 153 | 153 | 172 | 172 | 172 |
| | 0.25V | 82 | 46 | 46 | 87 | 79 | 83 | 84 | 86 | 88 | 80 | 80 | 82 | 96 | 96 | 96 | 112 | 113 | 113 | 127 | 127 | 127 | 143 | 143 | 143 |
| | 0V | 66 | 37 | 37 | 65 | 68 | 67 | 73 | 77 | 73 | 66 | 67 | 68 | 79 | 79 | 82 | 92 | 93 | 93 | 104 | 104 | 107 | 116 | 117 | 117 |
| Model2 | 0.75V | 147 | 117 | 119 | 205 | 212 | 218 | 204 | 206 | 164 | 128 | 127 | 126 | 152 | 151 | 151 | 174 | 174 | 172 | 195 | 196 | 195 | 214 | 214 | 219 |
| | 0.50V | 90 | 58 | 60 | 105 | 103 | 105 | 105 | 105 | 99 | 99 | 99 | 98 | 116 | 117 | 117 | 136 | 137 | 135 | 154 | 159 | 154 | 172 | 174 | 172 |
| | 0.25V | 79 | 45 | 25 | 83 | 81 | 63 | 82 | 88 | 88 | 81 | 82 | 82 | 96 | 97 | 97 | 114 | 114 | 113 | 129 | 131 | 128 | 144 | 144 | 144 |
| | 0V | 68 | 36 | 20 | 72 | 68 | 48 | 74 | 75 | 75 | 67 | 68 | 69 | 80 | 80 | 80 | 93 | 94 | 92 | 105 | 106 | 105 | 117 | 117 | 117 |
| Model3 | 0.75V | 142 | 103 | 119 | 193 | 190 | 220 | 194 | 196 | 222 | 125 | 125 | 128 | 149 | 150 | 150 | 170 | 168 | 175 | 195 | 191 | 194 | 215 | 214 | 218 |
| | 0.50V | 85 | 52 | 58 | 95 | 94 | 102 | 101 | 103 | 107 | 98 | 97 | 98 | 116 | 116 | 116 | 135 | 135 | 135 | 153 | 153 | 154 | 172 | 173 | 173 |
| | 0.25V | 75 | 44 | 46 | 78 | 77 | 82 | 82 | 82 | 86 | 81 | 81 | 81 | 96 | 97 | 96 | 112 | 112 | 112 | 128 | 128 | 128 | 143 | 144 | 144 |
| | 0V | 63 | 34 | 39 | 64 | 62 | 67 | 75 | 73 | 71 | 67 | 67 | 67 | 79 | 79 | 79 | 92 | 92 | 92 | 104 | 104 | 104 | 117 | 117 | 117 |
| Model4 | 0.75V | 165 | | | 234 | | | 234 | | | 130 | | | 153 | | | 172 | | | 193 | | | 218 | | |
| | 0.50V | 101 | | | 123 | | | 122 | | | 100 | | | 118 | | | 136 | | | 155 | | | 175 | | |
| | 0.25V | 84 | | | 96 | | | 97 | | | 83 | | | 99 | | | 114 | | | 130 | | | 145 | | |
| | 0V | 83 | | | 88 | | | 86 | | | 70 | | | 81 | | | 93 | | | 106 | | | 120 | | |

**Table 8.** Average delay time (seconds) of a single trial of movement in DB2.

| DB2 | | W=25 | | | W=50 | | | W=75 | | | W=100 | | | W=125 | | | W=150 | | | W=175 | | | W=200 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V/K | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 |
| Model1 | 0.75V | 162 | 120 | 120 | 226 | 221 | 223 | 200 | 214 | 226 | 153 | 157 | 159 | 178 | 178 | 180 | 197 | 203 | 206 | 215 | 225 | 233 | 233 | 231 | 238 |
| | 0.50V | 94 | 58 | 56 | 109 | 102 | 103 | 107 | 105 | 107 | 114 | 116 | 115 | 133 | 133 | 133 | 151 | 176 | 152 | 169 | 169 | 171 | 186 | 187 | 187 |
| | 0.25V | 82 | 46 | 46 | 87 | 79 | 83 | 84 | 86 | 88 | 100 | 100 | 100 | 115 | 115 | 115 | 130 | 132 | 143 | 145 | 153 | 147 | 160 | 160 | 160 |
| | 0V | 66 | 37 | 37 | 65 | 68 | 67 | 73 | 77 | 73 | 86 | 87 | 87 | 101 | 101 | 99 | 110 | 110 | 110 | 123 | 123 | 123 | 134 | 134 | 134 |
| Model2 | 0.75V | 147 | 117 | 119 | 205 | 212 | 218 | 204 | 206 | 164 | 156 | 158 | 162 | 180 | 180 | 199 | 193 | 205 | 219 | 214 | 219 | 230 | 243 | 238 | 251 |
| | 0.50V | 90 | 58 | 60 | 105 | 103 | 105 | 105 | 105 | 99 | 118 | 119 | 121 | 135 | 135 | 142 | 156 | 172 | 163 | 174 | 177 | 183 | 203 | 188 | 192 |
| | 0.25V | 79 | 45 | 25 | 83 | 81 | 63 | 82 | 88 | 88 | 102 | 102 | 104 | 117 | 117 | 124 | 132 | 135 | 137 | 147 | 149 | 153 | 162 | 160 | 164 |
| | 0V | 68 | 36 | 20 | 72 | 68 | 48 | 74 | 75 | 75 | 88 | 89 | 91 | 100 | 100 | 103 | 110 | 111 | 112 | 123 | 124 | 127 | 137 | 136 | 138 |
| Model3 | 0.75V | 142 | 103 | 119 | 193 | 190 | 220 | 194 | 196 | 222 | 158 | 162 | 158 | 192 | 192 | 213 | 195 | 215 | 230 | 216 | 220 | 230 | 243 | 244 | 242 |
| | 0.50V | 85 | 52 | 58 | 95 | 94 | 102 | 101 | 103 | 107 | 117 | 119 | 119 | 154 | 154 | 163 | 157 | 164 | 157 | 175 | 177 | 182 | 190 | 191 | 201 |
| | 0.25V | 75 | 44 | 46 | 78 | 77 | 82 | 82 | 82 | 86 | 101 | 102 | 103 | 123 | 123 | 120 | 133 | 136 | 138 | 147 | 149 | 153 | 161 | 162 | 162 |
| | 0V | 63 | 34 | 39 | 64 | 62 | 67 | 75 | 73 | 71 | 88 | 90 | 88 | 100 | 100 | 105 | 111 | 118 | 112 | 123 | 124 | 126 | 136 | 137 | 138 |
| Model4 | 0.75V | 165 | | | 234 | | | 234 | | | 165 | | | 191 | | | 200 | | | 220 | | | 239 | | |
| | 0.50V | 101 | | | 123 | | | 122 | | | 122 | | | 139 | | | 160 | | | 178 | | | 191 | | |
| | 0.25V | 84 | | | 96 | | | 97 | | | 105 | | | 119 | | | 134 | | | 149 | | | 163 | | |
| | 0V | 83 | | | 88 | | | 86 | | | 92 | | | 103 | | | 112 | | | 124 | | | 138 | | |

**Table 9.** Average delay time (seconds) of a single trial of movement in DB3.

| DB3 | | W=25 | | | W=50 | | | W=75 | | | W=100 | | | W=125 | | | W=150 | | | W=175 | | | W=200 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | V/K | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 | K=3 | K=5 | K=7 |
| Model1 | 0.75V | 55 | 34 | 12 | 66 | 68 | 47 | 71 | 77 | 79 | 87 | 90 | 91 | 95 | 96 | 98 | 109 | 109 | 110 | 118 | 120 | 121 | 123 | 123 | 132 |
| | 0.50V | 32 | 20 | 20 | 36 | 38 | 38 | 43 | 43 | 45 | 60 | 61 | 62 | 69 | 69 | 70 | 78 | 79 | 80 | 86 | 87 | 88 | 94 | 95 | 95 |
| | 0.25V | 26 | 16 | 16 | 30 | 28 | 31 | 37 | 34 | 38 | 49 | 50 | 51 | 57 | 57 | 57 | 64 | 65 | 66 | 73 | 72 | 73 | 78 | 78 | 79 |
| | 0V | 24 | 13 | 14 | 28 | 25 | 26 | 34 | 29 | 32 | 41 | 41 | 41 | 48 | 47 | 48 | 54 | 53 | 54 | 61 | 61 | 60 | 65 | 65 | 65 |
| Model2 | 0.75V | 57 | 35 | 37 | 68 | 73 | 76 | 76 | 83 | 84 | 87 | 91 | 92 | 97 | 101 | 103 | 110 | 109 | 111 | 118 | 124 | 125 | 124 | 128 | 136 |
| | 0.50V | 34 | 21 | 21 | 39 | 40 | 41 | 45 | 43 | 49 | 61 | 62 | 64 | 68 | 72 | 72 | 79 | 79 | 80 | 87 | 88 | 88 | 95 | 96 | 101 |
| | 0.25V | 28 | 17 | 17 | 34 | 30 | 34 | 40 | 32 | 42 | 50 | 53 | 51 | 57 | 59 | 58 | 65 | 66 | 65 | 73 | 73 | 73 | 79 | 79 | 83 |
| | 0V | 24 | 15 | 14 | 28 | 26 | 31 | 31 | 27 | 37 | 41 | 42 | 42 | 46 | 49 | 48 | 53 | 54 | 53 | 60 | 61 | 62 | 65 | 65 | 67 |
| Model3 | 0.75V | 57 | 38 | 41 | 69 | 75 | 77 | 74 | 79 | 82 | 87 | 90 | 91 | 99 | 103 | 107 | 112 | 114 | 116 | 120 | 124 | 123 | 125 | 128 | 126 |
| | 0.50V | 33 | 22 | 22 | 37 | 41 | 43 | 44 | 44 | 47 | 60 | 61 | 62 | 71 | 73 | 74 | 80 | 82 | 83 | 88 | 89 | 90 | 94 | 94 | 95 |
| | 0.25V | 27 | 18 | 17 | 31 | 32 | 33 | 37 | 36 | 38 | 49 | 50 | 50 | 58 | 59 | 60 | 66 | 67 | 67 | 73 | 74 | 74 | 78 | 80 | 79 |
| | 0V | 24 | 15 | 16 | 26 | 27 | 29 | 32 | 30 | 31 | 40 | 41 | 41 | 48 | 49 | 49 | 54 | 55 | 57 | 60 | 61 | 60 | 64 | 64 | 64 |
| Model4 | 0.75V | 59 | | | 75 | | | 85 | | | 93 | | | 101 | | | 112 | | | 123 | | | 129 | | |
| | 0.50V | 34 | | | 40 | | | 49 | | | 63 | | | 71 | | | 80 | | | 89 | | | 97 | | |
| | 0.25V | 27 | | | 32 | | | 40 | | | 52 | | | 59 | | | 65 | | | 73 | | | 80 | | |
| | 0V | 25 | | | 29 | | | 35 | | | 43 | | | 49 | | | 55 | | | 61 | | | 66 | | |

### D. *Analysis of Time Delay of the CNNs for varied Window Parameters of 2D sEMG Signals.*

In this section, we quantified the corresponding latency efficiency of the CNNs when the constructed 2D sEMG signals were applied. In this case, the response time ($T_e$) is an essential parameter that reflects the delay observed when the CNN model is deployed in a prosthesis for use in practical settings. This parameter could be seen as a function of the amount of time taken to decode a MI task from a single trial evaluation. The previous studies investigated the microcontroller delay time when the prosthesis is in use and found that it can be computed using the relationship in equation 4. [26], [24], [25].

$$DT = \frac{1}{2}W + \frac{1}{2}V + T_e \qquad (4)$$

Where, $DT$ is the microcontroller delay time, $W$ is the window length, $V$ is the overlap, and $T_e$ is the response time of CNN (from preprocessing the sEMG and generation of 2D signals to yielding a decision). Thus, equation 4 was utilized to compute the latency (program execution time) of the CNNs built based on varied combination of window parameters across the kernel sizes as shown in Tables 7, 8, and 9 for the three databases (DB1, DB2, and DB3), respectively.

By closely analyzing the results across one trial of movement in Tables 7, 8, and 9, it can be seen that the delay time decreases with a decrease in the overlap. The overlaps of 75% have the highest computation times. Moreover, the increase in kernel size also increases the inference time. DB2 (in Table 8) has the highest times relative to other databases. While the delay times in DB1 (Table 7) are all (except for W= 50, 75, and 200ms at overlap 0.75V) within an acceptable range for the practical application of the prostheses, which is stated to be below 200ms [24]. All delay times in DB3 (Table 9) are within the acceptable range. Therefore, when considering the combination of both accuracy and time efficiency of the CNNs for prostheses-based applications, networks with kernel =7 and an overlap of 50% would yield reliable operations in all databases (below 200ms) and 75% overlap for DB3.

## IV. DISCUSSION

This work investigated the relationship between the length of the 2D sEMG signal window and the overlap, which can produce the network's best performance. We also studied how CNN kernel sizes affect the network's performance on classification tasks when raw 2D sEMG signals constructed using different combinations of

window length and overlaps were used as input to the CNN. The investigation considered four different kinds of CNN models. It should be noted that, although this study focused on raw EMG as input to the CNN, we also acknowledge the potential of using other forms of inputs to the CNN such as spectrograms [48].

Specifically, the results presented in section III.A demonstrated the effects of window overlap and window length of the generated 2D signals on the performance of the CNNs. In general, the results in Fig. 5 & Table 4, Fig. 6 & Table 5, and Fig. 7 & Table 6 for DB1, DB2, and DB3, respectively, show that the performance of the CNNs is greatly influenced by the amount of overlap in the generated 2D signals. Interestingly, the highest overlap achieved the maximum improvement of 20.91% in DB1, 31.70% in DB2, and 39.60% F1-score in DB3. Further analysis is presented in section III. C (Fig. 9), which depicts the effects of varying window overlaps and kernel sizes on the individual motion intention in Fig. 9 (a) (for DB1), Fig. 9 (b) (for DB2), and Fig. 9 (c) (for DB3). The classification performances across individual motion intents show that 75% (0.75V) of signal overlap has achieved higher performances in all gestures compared to 0% overlap (0V). This, emphasizes that the use of 2D sEMG signal with large overlaps can achieve reasonable and practical results for CNN.

On the other hand, the results in section III. B show that the effect of window length of 2D sEMG signals becomes significant when the number of channels decreases. In that regard, the impacts of varying window lengths were not realized when DB1 with 128 channels (signals of size Wx128) was used as indicated in Fig. 8 (a). The effect started to show up in DB2 (32 channels) in Fig. 8 (b). A consistent and positive impact of window length is observed in Fig. 8 (c) when DB3 (which uses 2D signals from 12 channels) was deployed. This finding corroborates the previous work [26], on engineered features which also showed that the effects of window length increase with the decrease in the number of channels. This relationship is likely described by the feature space in non-dense channels (i.e., 12 channels in DB3) that provide less spatial information which on the other hand can be compensated by additional temporal information provided by longer window lengths. But longer window lengths provide less benefit to large-channel clusters (i.e., 128 channels in DB1 and 32 channels in DB2) that are already more adequately distributed in feature space.

The CNN's kernel can be thought of as the network's eye; the larger the kernel, the larger the network's receptive field, allowing it to see more of the signal and identify more widespread spatial patterns (features), whereas a smaller kernel size identifies finer features [22]. The results attained in Fig. 8 indicate that the increase in kernel size improves the CNN's performance by a significant margin at constant window length, with the peak performance attained by kernel =7. This implied that the patterns of the sEMG signals can be modelled over a large spatial dimension of the signal [49]. Besides, a wide field of view allows the network to learn hidden temporal connectivity within the signal [22]. Hence, it can model both spatial and temporal patterns of the signal. This provides us with an insight that it may be essential to consider shallow networks but with large kernels to have classifiers with high accuracy which are time and memory efficient.

Additionally, when we compare the general results across different databases between the amputees-based database (DB2 and DB3) and the database containing the able-bodied subjects' data (DB1). The results obtained in DB1 in all experiments are relatively higher than in DB2 and DB3. This could be due to the high quality of the signals collected from able-bodied subjects contrary to the amputee subjects, whose signal qualities are normally low due to limited residual muscles [50]. However, the application of a high percentage of overlaps (particularly 75%) in the generated 2D signals has improved the classification results of DB2 and DB3 to an acceptable level in all Models. Therefore, a combination of 75% overlap and large kernels such as 7 may have the potential to realize the practical use of CNN-based prostheses.

Furthermore, we estimated the possible delay time of the prosthesis microcontroller based on CNNs in relation to the window length, overlap, and kernel size on a single trial of movement as indicated in Tables 7, 8, and 9. The trend shows that the delay time increases with an increase in overlap and kernel sizes. Besides, we discovered that when considering efficiency in time and accuracy, an overlap of 50% and a kernel size of 7 can give an ideal response in all tested databases but with slightly low accuracy compared to 75% overlap which gives the highest accuracy (in all databases) with slightly low efficiency in time, particularly in DB1 and DB2. Hence such a trade-off between overlaps of 50% and 75% could be taken into account by the prosthesis manufacturers. In general, the analysis presented in this study leads to the following rule of thumb: *The overall decoding performance of convolutional neural networks is observed to be highly dependent on the window based hyperparameters including overlap and kernel size which validates our hypothesis***.**

Lastly, we compared the relative performance of the proposed model MS-CNN (Model4) and other models which have been used in the study (Model1, Model2, and Model3). Considering the average classification results depicted in Fig. 5 (b), Fig. 6 (b), and Fig. 7 (b), it can be observed that the proposed Model4 achieved the highest performances in all overlaps in DB1 (Fig. 5 (b)), in DB2 (Fig. 6 (b)) for all overlaps except overlap of 75%), and in DB3 (Fig. 7(b)) all overlaps except for overlap of 75%). The stable performances exhibited by Model4 across all databases suggest that the proposed Model4 would be more suitable for decoding sEMG based motion intents.

## CONCLUSION

This work has extensively examined the impacts of varying the segmentation window parameters for raw 2D sEMG signal construction and kernel sizes on the performance of CNNs for upper limb movement classification. Moreover, we have designed a less complex, accurate and memory-efficient CNN model whose key operations are based on dilation convolutional layers. From the experimental results, we found that the combination of kernel size of 7 and overlap of 75% achieved higher classification performance than other combinations with slightly low computational time. However, a combination of kernel size of 7 and an overlap of 50% gives acceptable computational time with slightly lower classification accuracy. Therefore, when manufacturing the prostheses, a combination of 75% overlap with shallow CNN but large kernel sizes (such as 7) would provide considerably high accuracy, time, and memory efficiency. These findings will help assist prostheses manufacturers in developing more efficient sEMG-based gesture recognition algorithms. The newly proposed CNN model yielded significantly better motor decoding performance than others tested, thus, its deployment in prosthetics and other fields such as rehabilitation control systems could facilitate multifunctional and intuitive EMG-PR control schemes. Finally, for the first time, we have been able to establish with substantial evidence through multiple experiments that window-based hyper-parameters substantially impact the decoding outcome and computational complexity of deep neural networks. In general, we anticipate that this novel finding may spur positive development in myoelectric control and related fields.

Transfer learning, where a pre-trained model is fine-tuned on a small dataset from the end-application, is an important technique when using little data (especially human data such as with EMG)

for training deep learning models. In that respect, due to their simplicity and good performance, we anticipate that the CNN models proposed in this study can be useful for transfer learning of parsimonious models in EMG-related fields/tasks such as inter-subject and inter limb/hand domain adaptation.

Despite the merits of the findings observed in this study, the analyses were based on offline experiments. Consequently, there is still room for improvement, particularly in the aspect of incorporation of hyperparameters and investigating the variation of window parameters in the performance of the CNN models in an online setting. In our future study, we hope to further validate the findings of the study and the performance of the proposed CNN network in an online setting, taking factors that may impact the real-time control performance of the multifunctional prostheses into consideration.

## REFERENCES

[1] Nsugbe, Ejay, Carol Phillips, Mike Fraser, and Jess McIntosh. "Gesture recognition for transhumeral prosthesis control using EMG and NIR." IET Cyber-Systems and Robotics 2, no. 3 (2020): 122-131.

[2] Pancholi, Sidharth, and Amit M. Joshi. "Electromyography-based hand gesture recognition system for upper limb amputees." IEEE Sensors Letters 3, no. 3 (2019): 1-4.

[3] Unanyan, Narek N., and Alexey A. Belov. "Design of upper limb prosthesis using real-time motion detection method based on EMG signal processing." Biomedical Signal Processing and Control 70 (2021): 103062. .

[4] Hahne, Janne M., Dario Farina, Ning Jiang, and David Liebetanz. "A novel percutaneous electrode implant for improving robustness in advanced myoelectric control." Frontiers in Neuroscience 10 (2016): 114.

[5] Davis, Tyler S., Heather AC Wark, D. T. Hutchinson, David J. Warren, K. O'neill, T. Scheinblum, Gregory A. Clark, Richard A. Normann, and Bradley Greger. "Restoring motor control and sensory feedback in people with upper extremity amputations using arrays of 96 microelectrodes implanted in the median and ulnar nerves." Journal of neural engineering 13, no. 3 (2016): 036001.

[6] Hebert, Jacqueline S., Jaret L. Olson, Michael J. Morhart, Michael R. Dawson, Paul D. Marasco, Todd A. Kuiken, and K. Ming Chan. "Novel targeted sensory reinnervation technique to restore functional hand sensation after transhumeral amputation." IEEE Transactions on Neural Systems and Rehabilitation Engineering 22, no. 4 (2013): 765-773.

[7] Yang, W., Yang, D., Liu, Y. and Liu, H., 2019. Decoding simultaneous multi-DOF wrist movements from raw EMG signals using a convolutional neural network. IEEE Transactions on Human-Machine Systems, 49(5), pp.411-420.

[8] Hargrove, L.J., Scheme, E.J., Englehart, K.B. and Hudgins, B.S., 2010. Multiple binary classifications via linear discriminant analysis for improved controllability of a powered prosthesis. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 18(1), pp.49-57.

[9] Côté-Allard, U., Campbell, E., Phinyomark, A., Laviolette, F., Gosselin, B. and Scheme, E., 2020. Interpreting deep learning features for myoelectric control: A comparison with handcrafted features. Frontiers in Bioengineering and Biotechnology, 8, p.158

[10] Zhou, H., Zhang, Q., Zhang, M., Shahnewaz, S., Wei, S., Ruan, J., Zhang, X. and Zhang, L., 2021. Toward hand pattern recognition in assistive and rehabilitation robotics using emg and kinematics. Frontiers in Neurorobotics, 15, p.659876.

[11] Li, X., Xu, R., Samuel, O.W. et al. 2016, August. A new approach to mitigate the effect of force variation on pattern recognition for myoelectric control. In 2016 38th Annual International Conference of the IEEE EMBS (EMBC) (pp. 1684-1687). IEEE.

[12] Samuel, O.W., Asogbon, M.G., Geng, Y., Li, X., Pirbhulal, S., Chen, S., Ganesh, N., Feng, P. and Li, G., 2019, July. Spatio-temporal based descriptor for limb movement-intent characterization in EMG-pattern recognition system. In 2019 41st Annual International Conference of the IEEE EMBS (EMBC) (pp. 2637-2640). IEEE.

[13] Khushaba, R.N. and Nazarpour, K., 2021. Decoding HD-EMG Signals for Myoelectric Control-How Small Can the Analysis Window Size be?. IEEE Robotics and Automation Letters, 6(4), pp.8569-8574.

[14] Zhai, X., Jelfs, B., Chan, R.H. and Tin, C., 2017. Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network. Frontiers in neuroscience, 11, p.379.

[15] Oh, D.C. and Jo, Y.U., 2021. Classification of hand gestures based on multi-channel EMG by scale Average wavelet transform and convolutional neural network. International Journal of Control, Automation and Systems, 19(3), pp.1443-1450.

[16] BAKIRCIOĞLU, K. and Özkurt, N., 2020. Classification of EMG signals using convolution neural network. International Journal of Applied Mathematics Electronics and Computers, 8(4), pp.115-119.

[17] Zanghieri, M., Benatti, S., Burrello, A., et al., 2019. Robust real-time embedded EMG recognition framework using temporal convolutional networks on a multicore IoT processor. IEEE transactions on biomedical circuits and systems, 14(2), pp.244-256.

[18] Tsinganos, P., Cornelis, B., Cornelis, J., Jansen, B. and Skodras, A., 2018, September. Deep Learning in EMG-based Gesture Recognition. In PhyCS (pp. 107-114).

[19] Olsson, A., Malešević, N., Björkman, A. and Antfolk, C., 2019, July. Exploiting the intertemporal structure of the upper-limb sEMG: Comparisons between an LSTM network and cross-sectional myoelectric pattern recognition methods. In 2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 6611-6615). IEEE.

[20] Rahimian, E., Zabihi, S., Asif, A., Farina, D., Atashzar, S.F. and Mohammadi, A., 2021. Fs-hgr: Few-shot learning for hand gesture recognition via electromyography. IEEE transactions on neural systems and rehabilitation engineering, 29, pp.1004-1015.

[21] Atzori, M., Cognolato, M. and Müller, H., 2016. 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.

[22] Rahimian, E., Zabihi, S., Atashzar, S.F., Asif, A. and Mohammadi, A., 2019, November. Semg-based hand gesture recognition via dilated convolutional neural networks. In 2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP) (pp. 1-5).

[23] Gulati, P., Hu, Q. and Atashzar, S.F., 2021. Toward deep generalization of peripheral EMG-Based Human-Robot interfacing: A hybrid explainable solution for NeuroRobotic systems. IEEE Robotics and Automation Letters, 6(2), pp.2650-2657.

[24] Asogbon, M.G., Samuel, O.W., Jiang, Y., Wang, L., Geng, Y., Sangaiah, A.K., Chen, S., Fang, P. and Li, G., 2020. Appropriate feature set and window parameters selection for efficient motion intent characterization towards intelligently smart EMG-PR system. Symmetry, 12(10), p.1710.

[25] Farrell, T. and Weir, R.F., 2008. Analysis window induced controller delay for multifunctional prostheses. Myoelectric Symposium.

[26] Smith, L.H., Hargrove, L.J., Lock, B.A. and Kuiken, T.A., 2010. Determining the optimal window length for pattern recognition-based myoelectric control: balancing the competing effects of classification error and controller delay. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 19(2), pp.186-192.

[27] Kamavuako, E.N., Scheme, E.J. and Englehart, K.B., 2016. Determination of optimum threshold values for EMG time domain features; a multi-dataset investigation. Journal of neural engineering, 13(4), p.0460

[28] Öztürk, Ş., Özkaya, U., Akdemir, B. and Seyfi, L., 2018, November. Convolution kernel size effect on convolutional neural network in histopathological image processing applications. In *2018 International Symposium on Fundamentals of Electrical Engineering (ISFEE)* (pp. 1-5). IEEE.

[29] Geng, W., Du, Y., Jin, W., Wei, W., Hu, Y. and Li, J., 2016. Gesture recognition by instantaneous surface EMG images. Scientific reports, 6(1), pp.1-8.

[30] Atzori, M., Gijsberts, A., Castellini, C., Caputo, B., Hager, A.G.M., Elsig, S., Giatsidis, G., Bassetto, F. and Müller, H., 2014. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Scientific data, 1(1), pp.1-13.

[31] Chang, J., Phinyomark, A. and Scheme, E., 2020, July. Assessment of EMG benchmark data for gesture recognition using the ninapro database. In 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC) (pp. 3339-3342). IEEE.

[32] Shin, S., Baek, Y., Lee, J., Eun, Y. and Son, S.H., 2017, November. Korean sign language recognition using EMG and IMU sensors based on group-dependent NN models. In 2017 IEEE Symposium Series on Computational Intelligence (SSCI) (pp. 1-7). IEEE.

[33] Al Shalabi, L. and Shaaban, Z., 2006, May. Normalization as a preprocessing engine for data mining and the approach of preference matrix. In *2006 International conference on dependability of computer systems* (pp. 207-214). F25mIEEE.

[34] Kulwa, F., Samuel, O. W., Asogbon, M. G., Obe, O. O., & Li, G. (2022, June). Analyzing the Impact of Varied Window Hyper-parameters on Deep CNN for sEMG based Motion Intent Classification. In 2022 IEEE International Workshop on Metrology for Industry 4.0 & IoT (MetroInd4. 0&IoT) (pp. 81-86). IEEE.

[35] Jia, W., Qu, H., Ma, J., Xia, Y., Cui, D., Liu, Y. and Li, L., 2022. Adjacent age classification algorithm of yellow-feathered chickens based on multi-scale feature fusion. Computers and Electronics in Agriculture, 200, p.107264.

[36] Maufroy, C. and Bargmann, D., 2018, October. CNN-Based detection and classification of grasps relevant for worker support scenarios using sEMG signals of forearm muscles. In 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 141-146). IEEE.

[37] Narkhede, M.V., Bartakke, P.P. and Sutaone, M.S., 2022. A review on weight initialization strategies for neural networks. Artificial intelligence review, 55(1), pp.291-322.

[38] Bai, N., Wang, Z., & Meng, F. (2020). A stochastic attention CNN model for rumor stance classification. IEEE Access, 8, 80771-80778.

[39] Yu, F. and Koltun, V., 2015. Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122.

[40] Bai, S., Kolter, J.Z. and Koltun, V., 2018. An empirical evaluation of generic convolutional and recurrent networks for sequence modeling. arXiv preprint arXiv:1803.01271.

[41] Chen, P.H. and Lee, C.Y., 2018, March. Uavnet: an efficient obstacel detection model for uav with autonomous flight. In 2018 International Conference on Intelligent Autonomous Systems (ICoIAS) (pp. 217-220). IEEE

[42] Zhang, X., Zou, Y. and Shi, W., 2017, August. Dilated convolution neural network with LeakyReLU for environmental sound classification. In 2017 22nd international conference on digital signal processing (DSP) (pp. 1-5). IEEE.

[43] Hasan, M.J., Shon, D., Im, K., Choi, H.K., Yoo, D.S. and Kim, J.M., 2020. Sleep state classification using power spectral density and residual neural network with multichannel EEG signals. Applied Sciences, 10(21), p.7639.

[44] Ide, H. and Kurita, T., 2017, May. Improvement of learning for CNN with ReLU activation by sparse regularization. In 2017 International Joint Conference on Neural Networks (IJCNN) (pp. 2684-2691). IEEE.

[45] Diederik P. Kingma and Jimmy Lei Ba. Adam: a Method for Stochastic Optimization. International Conference on Learning Representations, pages 1–13, 2015.

[46] Ruder, S., 2016. An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

[47] Zhang, Z. and Sabuncu, M., 2018. Generalized cross entropy loss for training deep neural networks with noisy labels. *Advances in neural information processing systems*, *31*.

[48] Côté-Allard, U., Fall, C.L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F. and Gosselin, B., 2019. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE transactions on neural systems and rehabilitation engineering*, *27*(4), pp.760-771.

[49] Stango, A., Negro, F. and Farina, D., 2014. Spatial correlation of high-density EMG signals provides features robust to electrode number and shift in pattern recognition for myocontrol. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 23(2), pp.189-198.

[50] Samuel, O.W., Asogbon, M.G., Geng, Y., Al-Timemy, A.H., Pirbhulal, S., Ji, N., Chen, S., Fang, P. and Li, G., 2019. Intelligent EMG pattern recognition control method for upper-limb multifunctional prostheses: advances, current challenges, and future prospects. Ieee Access, 7, pp.10150-10165.