

# Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning

Ulysse Côté-Allard, Cheikh Latyr Fall, Alexandre Drouin,

Alexandre Campeau-Lecours, Clément Gosselin, Kyrre Glette, François Laviolette, and Benoit Gosselin

**Abstract**—In recent years, the use of deep learning algorithms has become increasingly more prominent for their unparalleled ability to automatically learn discriminant features from large amounts of data. However, within the field of electromyography-based gesture recognition, deep learning algorithms are seldom employed as it requires an unreasonable amount of time for a single person, in a single session, to generate tens of thousands of examples.

This work's hypothesis is that general, informative features can be learned from the large amount of data generated by aggregating the signals of multiple users, thus reducing the recording burden imposed on a single person while enhancing gesture recognition. As such, this paper proposes applying transfer learning on the aggregated data of multiple users, while leveraging the capacity of deep learning algorithms to learn discriminant features from large dataset, without the need for in-depth feature engineering. To this end, two datasets are recorded with the Myo Armband (Thalmic Labs), a low-cost, low-sampling rate (200Hz), 8-channel, consumer-grade, dry electrode sEMG armband. These two datasets are comprised of 19 and 17 able-bodied participants respectively. A third dataset, also recorded with the Myo Armband, was taken from the NinaPro database and is comprised of 10 able-bodied participants. This transfer learning scheme is shown to outperform the current state-of-the-art in gesture recognition. It achieves an average accuracy of 98.31% for 7 hand/wrist gestures over 17 able-bodied participants and 65.57% for 18 hand/wrist gestures over 10 able-bodied participants. Finally, a use-case study employing eight able-bodied participants suggests that real-time feedback reduces the degradation in accuracy normally experienced over time.

**Index Terms**—Surface Electromyography, sEMG, Transfer Learning, Domain Adaptation, Deep Learning, Convolutional Networks, Hand Gesture Recognition, Prosthetic Control

## I. INTRODUCTION

Robotics and artificial intelligence can be leveraged to increase the autonomy of people living with disabilities. This is accomplished, in part, by enabling users to seamlessly interact with robots to complete their daily tasks with increased independence. In the context of hand prosthetic control, muscle activity provides an intuitive interface on which to perform hand gesture recognition [1]. This activity can be recorded by surface electromyography (sEMG), a non-invasive technique

widely adopted both in research and clinical settings. The sEMG signals, which are non-stationary, represent the sum of subcutaneous motor action potentials generated through muscular contraction [1]. Artificial intelligence can then be leveraged as the bridge between sEMG signals and the prosthetic behavior.

The literature on sEMG-based gesture recognition primarily focuses on feature engineering, with the goal of characterizing sEMG signals in a discriminative way [1], [2], [3]. Recently, researchers have proposed deep learning approaches [4], [5], [6], shifting the paradigm from feature engineering to feature learning. Regardless of the method employed, the end-goal remains the improvement of the classifier's robustness. One of the main factors for accurate predictions, especially when working with deep learning algorithms, is the amount of training data available. Hand gesture recognition creates a peculiar context where a single user cannot realistically be expected to generate tens of thousands of examples in a single sitting. Large amounts of data can however be obtained by aggregating the recordings of multiple participants, thus fostering the conditions necessary to learn a general mapping of users' sEMG signal. This mapping might then facilitate the hand gestures' discrimination task with new subjects. Consequently, deep learning offers a particularly attractive context from which to develop a transfer learning algorithm to leverage inter-user data by pre-training a model on multiple subjects before training it on a new participant.

As such, the main contribution of this work is to present a new transfer learning scheme employing a convolutional network (ConvNet) to leverage inter-user data within the context of sEMG-based gesture recognition. A previous work [7] has already shown that learning simultaneously from multiple subjects significantly enhances the ConvNet's performance whilst reducing the size of the required training dataset typically seen with deep learning algorithms. This paper extends upon the aforementioned conference paper's work improving the transfer learning algorithm to reduce its computational load and improving its performance. Additionally, two new ConvNet architectures specifically designed for the robust and efficient classification of sEMG signals are presented. Both the short-time Fourier transform-based spectrogram and Continuous Wavelet Transform (CWT) are considered for the characterization of the sEMG signals to be fed to these ConvNets. To the best of the authors' knowledge, this is the first time that CWTs are employed as features for the classification of sEMG-based hand gesture recognition (although they have been proposed for the analyze of myoelectric signal [8]). Another major

Ulysse Côté-Allard\*, Cheikh Latyr Fall and Benoit Gosselin are with the Department of Computer and Electrical Engineering, Alexandre Drouin and François Laviolette are with the Department of Computer Science and Software Engineering, and Alexandre Campeau-Lecours and Clément Gosselin are with the Department of Mechanical Engineering, Université Laval, Québec, Québec, G1V 0A6, Canada.

Kyrre Glette is with the Department of Informatics, University of Oslo, Oslo, 0315, Norway.

\*Contact author email: [ulyссе.cote-allard.1@ulaval.ca](mailto:ulyссе.cote-allard.1@ulaval.ca)

contribution of this article is the publication of a new sEMG-based gesture classification dataset comprised of 36 able-bodied participants. This dataset and the implementation of the ConvNets along with their transfer learning augmented version (Theano and PyTorch implementation available) are made readily available<sup>1</sup>. Finally, this paper presents a use-case experiment on the effect of real-time feedback on the performance of a classifier without recalibration over a period of fourteen days.

This paper is organized as follows. An overview of the related work in hand gesture recognition through deep learning and transfer learning/domain adaptation is given in Section II. Section III presents the proposed new hand gesture recognition dataset with data acquisition and processing details. A presentation of the different state-of-the-art feature sets employed in this work is given in Section IV. Section V thoroughly describes the proposed network's architecture, while Section VI presents the transfer learning techniques used to augment said architecture. Moreover, comparisons of the state-of-the-art in gesture recognition with the proposed deep learning approach are given in Section VII and Section VIII. Finally, a use-case experiment on the ability of users to counteract signal drift from sEMG signal is presented in Section IX.

## II. RELATED WORK

sEMG signals can vary significantly from subject to subject, even when precisely controlling for electrode placement [9]. Regardless, classifiers trained from a user can be applied to new participants achieving slightly better than random performances [9]. As such, more sophisticated techniques have been proposed to leverage inter-user information. For example, research has been done to find a projection of the feature space that bridges the gap between an original subject and a new user [10], [11]. Moreover, several works, such as [12], [13], [14], have proposed leveraging a pre-trained model removing the need to work simultaneously with data from multiple users. These non-deep learning approaches showed important performance gains when the model is augmented through pre-training compared to training from scratch. Although, for some of them, these gains might be due to poor hyperparameter (i.e. parameters whose values are set before the learning process) optimization [15].

Short-Time Fourier Transform (STFT) have been sparsely employed in the last decades for the classification of sEMG data [16], [17]. A possible reason for this limited interest in STFT is that much of the research on sEMG-based gesture recognition focuses on designing feature ensembles derived from multiple methods [2]. Because STFT on its own generates large amounts of features and are relatively computationally expensive, they can be challenging to integrate with other feature types. Additionally, STFTs have also been shown to be less accurate than Wavelet Transforms [16] on their own for the classification of sEMG data. Recently however, STFT features, in the form of spectrograms, have been applied as input feature space for the classification of sEMG data by leveraging ConvNet [4], [6]. A possible reason for this

newfound interest is that ConvNets are particularly well suited for the classification of image-like features.

CWT features have been employed for EMG signals analysis, but mainly for lower limbs body [18], [19], electrocardiogram analysis [20] and electroencephalography [21]. Wavelet-based features have been used in the past for sEMG-based hand gesture recognition [22]. The features employed however are based on the Discrete Wavelet Transform [23] and the Wavelet Packet Transform (WPT) [16] instead of the CWT. This preference might be due to the fact that both DWT and WPT are less computationally expensive than the CWT and are thus better suited to be integrated into an ensemble of features. Similarly to spectrograms however, CWT offers an attractive image-like representation to leverage ConvNets for sEMG signal classification. To the best of the authors' knowledge, this is the first time that CWT is utilized for sEMG-based hand gesture recognition.

In recent years, deep learning techniques and especially ConvNets, have started to be applied to the problem of hand gesture recognition with array [4], [5] and matrix [24] of electrodes. Additionally, other authors applied deep learning in conjunction with domain adaptation techniques [6] but for inter-session classification as opposed to the inter-subject context presented in this paper. To the best of our knowledge, this paper, which is an extension of [7] is the first time inter-user data is leveraged through transfer learning techniques for training deep learning algorithms on sEMG data.

## III. SEMG DATASET ACQUISITION

One of the major contributions of this article is to provide a new, publicly available, sEMG-based hand gesture recognition dataset, referred to as the *Myo Dataset*. This dataset contains two distinct sub-datasets with the first one serving as the *pre-training dataset* and the second as the *evaluation dataset*. The former, which is comprised of 19 able-bodied participants, should be employed to build, validate and optimize classification techniques. The latter, comprised of 17 able-bodied participants, is utilized only for the final testing and comparisons between different methods. To the best of our knowledge, this is the largest dataset published utilizing the commercially available Myo Armband (Thalmic Labs) and it is our hope that this dataset will become a useful tool for the sEMG-based hand gesture classification community, by facilitating the comparison of new classification methods.

The data acquisition protocol was approved by the Laval University Ethics committee (approbation number: 2017-026/21-02-2016).

### A. sEMG Recording

The electromyographic activity of each subject's forearm was recorded with the Myo Armband<sup>2</sup>. The Myo is an 8-channel, dry-electrode, low-sampling rate (200 Hz), low-cost consumer-grade sEMG armband. While the device also incorporates a 9-axis inertial measurement unit (IMU), it was deactivated for the recording of the dataset.

<sup>1</sup><https://github.com/Giguelingueling/MyoArmbandDataset>

<sup>2</sup><https://www.myo.com>

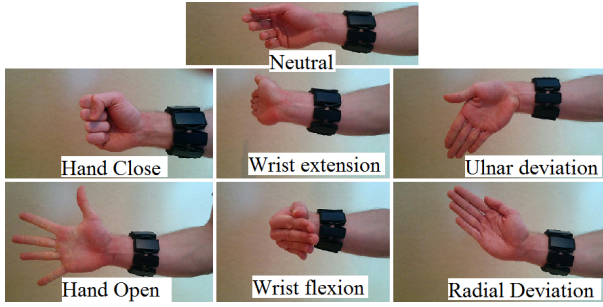


Fig. 1. The 7 hand/wrist gestures considered in this work.

The Myo is non-intrusive, as the dry-electrodes allow users to simply slip the bracelet on without any preparation. Comparatively, gel-based electrodes require the shaving and washing of the skin to obtain optimal contact between the subject's skin and electrodes. Unfortunately, the convenience of the Myo Armband comes with limitations regarding the quality and quantity of the sEMG signals that are collected. Indeed, dry electrodes, such as the ones employed in the Myo, are less accurate and robust to motion artifact than gel-based ones [25]. Additionally, while the recommended frequency range of sEMG signals is 5-500 Hz [26] requiring sampling frequency greater or equal to 1000 Hz, the Myo Armband is limited to 200 Hz. This information loss was shown to significantly impact the ability of various classifiers to differentiate between hand gestures [27]. As such, robust and adequate classification techniques are needed to process the collected signals accurately.

#### B. Hand/Wrist Gestures

Seven hand/wrist gestures are considered in this work. The first one referred to as *neutral* is the natural posture of the hand of the subject when no significant muscle activity is detected. The six other gestures are: ulnar deviation, radial deviation, hand open, hand close, wrist extension and wrist flexion. Fig 1 depicts all seven gestures.

#### C. Time-Window Length

For real-time control in a closed loop, input latency is an important factor to consider. A maximum time latency of 300 ms was first recommended in [28]. Even though more recent studies suggest that the latency should optimally be kept between 100-125 ms [29], the performance of the classifier should take priority over speed [29], [30]. As is the case in [4] and [7] a latency of 300 ms was selected to achieve a reasonable number of samples between each prediction due to the low frequency of the Myo.

#### D. Recording Labeled Data

For both sub-datasets, the labeled data was created by requiring the user to hold each gesture for five seconds. The recording of the full seven gestures for five seconds is referred to as a *cycle*, with four cycles forming a *round*. In the case of the *pre-training dataset*, a single round is available per



Fig. 2. Examples of the wide range of armband placements on the subjects' forearm

subject. For the *evaluation dataset* three rounds are available with the first round utilized for training (i.e. 140 s of data per participant) and the last two for testing.

For each participant, the armband was systematically tightened to its maximum and slid up the user's forearm until the circumference of the armband matched that of the forearm. This was done in an effort to reduce bias from the researchers, and to emulate the wide variety of armband positions that end-users without prior knowledge of optimal electrode placement might use (see Fig. 2). The raw sEMG data of the Myo is what is made available with this dataset.

Signal processing must be applied to efficiently train a classifier on the data recorded by the Myo armband. The data are first separated by applying sliding windows of 52 samples (260 ms) with an overlap of 235 ms. Employing windows of 260 ms allows 40 ms for the pre-processing and classification process. Note that utilizing sliding windows is viewed as a form of data augmentation in the present context (see Section V-A). This is done for each gesture in each cycle on each of the eight channels. As such, in the dataset, an *example* corresponds to the eight windows associated with their respective eight channels. From there, the processing depends on the classification techniques employed which will be detailed in Section IV and V

### IV. FEATURE EXTRACTION

Traditionally, one of the most researched aspects of sEMG-based gesture recognition comes from feature engineering. In other words, manually finding a representation for sEMG signals that allows easy differentiation between gestures while reducing the variance of examples within the same gestures. Over the years, several efficient combinations of features both in the time and frequency domain have been proposed [31], [32], [33], [34]. Features can be regrouped into different types, mainly: time, frequency and time-frequency domains. This section presents the feature sets used in this work. See the appendix for a description of each feature employed.

#### A. Feature Sets

As this paper's main purpose is to present a deep learning-based transfer learning approach to the problem of sEMG hand gesture recognition, a comparison with current classification methods is essential. As such, four different feature sets were taken from the literature to serve as a comparison basis. The four feature sets will be tested on five of the most common classifiers leveraged for sEMG pattern recognition: Support Vector Machine (SVM) [33], Artificial Neural Networks

(ANN) [35], Random Forest (RF) [33], K-Nearest Neighbors (NN) [33] and Linear Discriminant Analysis (LDA) [34]. As is often the case, dimensionality reduction is applied [1], [3], [36]. LDA was chosen to perform feature projection as it is computationally inexpensive, devoid of hyperparameters and was shown to allow for robust classification accuracy for sEMG-based gesture recognition [34], [37]. The implementation employed for all the classifiers comes from the scikit-learn<sup>3</sup> Python package [38]. The four feature sets employed for comparison purposes are the following:

1) *Time Domain Features (TD)* [32]: This set of features, which is probably the most commonly employed in the literature [27], often serves as the basis for bigger feature sets [1], [34], [39]. As such, TD is particularly well suited to serve as a baseline comparison for new classification techniques. The four features are: Mean Absolute Value (MAV), Zero Crossing (ZC), Slope Sign Changes (SSC) and Waveform Length (WL).

2) *Enhanced TD* [34]: This set of features includes the TD features in combination with Skewness, Root Mean Square (RMS), Integrated EMG (IEMG), Autoregression Coefficients (AR) ( $P=11$ ) and the Hjorth Parameters. It was shown to achieve excellent performance on a setup similar to the one employed in this article.

3) *Nina Pro Features* [33], [39]: This set of features was selected as it was found to perform the best in the article introducing the NinaPro dataset. The set consists of the normalized combination of the following features RMS, Marginal Discrete Wavelet Transform (mDWT) (wavelet=db7,  $S=3$ ), EMG Histogram (HIST) (bins=20, threshold= $3\sigma$ ) and the TD features.

4) *SampEn Pipeline* [31]: This last feature combination was selected among fifty features that were evaluated and ranked to find the most discriminating ones. The SampEn feature was ranked first amongst all the others. The best multi-features set found was composed of: SampEn( $m=2$ ,  $r=0.2*\sigma$ ), Cepstral Coefficient (order=4), RMS and WL.

## V. DEEP LEARNING CLASSIFIERS OVERVIEW

Deep learning algorithms, including ConvNets, are prone to overfitting especially on small datasets [40], [41]. Furthermore, deep networks tend to be computationally expensive and thus ill-suited for embedded systems, such as those required when guiding a prosthetic. However, in recent years, algorithmic improvements and new hardware architectures have allowed for complex networks to run on very low power systems. An overview of these advances is given in Section V-E.

As previously mentioned, due to inherent limitations from the context of sEMG-based hand gesture recognition, the proposed ConvNets will have to contend with a limited amount of data from any single individual. To address the over-fitting issue, Monte Carlo Dropout [41], Batch Normalization [42], and early stopping are employed and detailed in the following subsections. Considering the stochastic nature of the algorithms presented in this paper, unless stated otherwise, all experiments are reported as an average of 20 runs.

### A. Data Augmentation

The idea behind data augmentation is to augment the size of the training set, with the objective of achieving better generalization. This is generally accomplished by adding realistic noise to the training data, which tends to induce a robustness to noise into the learned model. In many cases, this has been shown to lead to better generalization [43], [44]. In this paper's context, data augmentation techniques can thus be viewed as part of the solution to reduce the overfitting from training a ConvNet on a small dataset. When adding noise to the data, it is important to ensure that the noise does not change the label of the examples. Hence, for image datasets, the most common and often successful techniques have relied on affine transformations [44].

Unfortunately, for sEMG signals, most of these techniques are unsuitable and cannot be applied directly. As such, specific data augmentation techniques must be employed. In this work, five data augmentation techniques are tested on the *pre-training dataset* as they are part of the architecture building process. Note that this comparison was made with the ConvNet architecture presented in [7], which takes as input a set of eight spectrograms (one for each channel of the Myo Armband).

Examples are constructed by applying non-overlapping windows of 260 ms. This non-augmented dataset is referred to as the *Baseline*. Consequently, an intuitive way of augmenting sEMG data is to apply overlapping windows (i.e. temporal translation) when building the examples. A major advantage of this technique within the context of sEMG signals - and time signals in general - is that it does not create any synthetic examples in the dataset compared to the affine transformation employed with images. Furthermore, with careful construction of the dataset, no new mislabeling occurs. In this work, this technique will be referred to as *Sliding Window* augmentation.

Second, the effect of muscle fatigue on the frequency response of muscles fibers [45] can be emulated, by altering the calculated spectrogram. The idea is to reduce the median frequency of a channel with a certain probability, by systematically redistributing part of the power of a frequency bin to an adjacent lower frequency one and so on. This was done in order to approximate the effect of muscle fatigue on the frequency response of muscles fibers [45]. In this work, this technique will be referred to as *Muscle Fatigue* augmentation.

The third data augmentation technique employed aims at emulating electrode displacement on the skin. This is of particular interest, as the dataset was recorded with a dry electrode armband, for which this kind of noise is to be expected. The data augmentation technique consists of shifting part of the power spectrum magnitude from one channel to the next. In other words, part of the signal energy from each channel is sent to an adjacent channel emulating electrode displacement on the skin. In this work, this approach will be referred to as *Electrode Displacement* augmentation.

For completeness, a fourth data augmentation technique which was proposed in a paper [5] employing a ConvNet for sEMG gestures classification is also considered in this work. The approach consists of adding a white Gaussian noise to the signal, with a signal-to-noise ratio of 25. This technique will

<sup>3</sup>scikit-learn v.1.13.1

be referred to as *Gaussian Noise* augmentation.

Finally, the application of all these data augmentation methods simultaneously is referred to as the *Aggregated Augmentation* technique.

Data from these augmentation techniques will be generated from the *pre-training dataset*. The data will be generated on the first two cycles, which will serve as the training set. The third cycle will be the validation set and the test set will be the fourth cycle. All augmentation techniques will generate double the amount of training examples compared to the baseline dataset.

Table I reports the average test set accuracy for the 19 participants over 20 runs. Furthermore, considering each participant as a separate dataset allows for the application of the Wilcoxon signed-rank test [46] to compare the *Baseline* with all the other data augmentations techniques. The null hypothesis is that the medians of the difference between two group samples are equal ( $n = 19$ ). The results of the statistical test are summarized in Table I. The two techniques that produce significantly different results from the *Baseline* are the *Gaussian Noise* (worst accuracy) and the *Sliding Window* (improves accuracy). As such, as described in Section III-D the only data augmentation technique employed in this work is the sliding windows.

#### B. Monte Carlo Dropout

Dropout is a regularization method that attempts to avoid overfitting by reducing the co-adaptation of hidden units [40]. During training, for each sample of the mini-batch, hidden units are randomly deactivated with probability  $p$  (hyperparameter). Backpropagation is then applied normally on the thinned network (i.e. the network with the dropped units) for each sample. Importantly, Dropout is not applied at test time.

Monte Carlo Dropout (MC Dropout) is an extension of traditional dropout and is aimed at modeling uncertainty in deep learning [41], [47]. The technique was in part developed to reduce overfitting when training ConvNets with a small dataset [47]. At training time, MC Dropout is functionally identical to traditional dropout. At test time however, MC Dropout remains active. The network prediction is obtained by sending the example  $M$  times through the ConvNet and averaging these  $M$  outcomes of the  $M$  resulting thinned networks. Note that MC Dropout is applied both on the convolutional and fully connected layers. On the *pre-training dataset*, MC Dropout marginally outperforms traditional dropout with an average accuracy of 97.30% and 97.11% respectively. Considering these results, MC Dropout is employed in this work, although traditional dropout remains a valid choice.

#### C. Batch Normalization

Batch Normalization (BN) is a regularization technique that aims to maintain a standard distribution of hidden layer activation values throughout training [42]. BN accomplishes this by normalizing the mean and variance of each dimension of a batch of examples. To achieve this, a linear transformation based on two learned parameters is applied to each dimension. This process is done independently for each layer of the

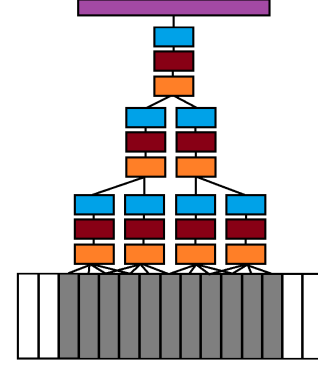


Fig. 3. Typical slow-fusion ConvNet architecture [49]. In this graph, the input, which is represented by the grey rectangles is a video (e.g. a sequence of images). The video is separated into subsets that each go through parallel convolutional layers, which are slowly merged together along the network.

network. Once training is completed, a post-training step is necessary before performing inference. This step consists of feeding the whole dataset through the network to compute the final normalization parameters in a layerwise fashion. At test time, these parameters are applied to normalize the layer activations. BN was shown to yield faster training times whilst allowing better generalization.

#### D. Proposed Convolutional Network Architecture

Videos are a representation of how spatial information (images) changes through time. Previous works have combined this representation with ConvNets to address classification tasks [48], [49]. One successful architecture for leveraging video-based information is called the *slow fusion model* [49]. This model's architecture is illustrated in Fig. 3. The slow-fusion model separates the temporal part of the examples into disconnected parallel layers, which are then slowly fused together throughout the network. By doing so, deeper layers have access to increasingly more temporal information. This model was also applied successfully for multi-view image fusion [50].

When calculating the spectrogram of a signal, the information is structured in a Time x Frequency fashion (Time x Scale for CWT). When the signal comes from an array of electrodes, these examples can naturally be structured as Time x Spatial x Frequency (Time x Spatial x Scale for CWT). As such, the motivation for using a slow-fusion architecture based ConvNets in this work is due to the similarities between videos data and the proposed characterization of sEMG signals, as both representations have analogous structures (i.e. Time x Spatial x Spatial for videos) and can describe non-stationary information. Additionally, the proposed architectures inspired by the slow-fusion model were by far the most successful of the ones tried on the pre-training dataset.

1) *Pooling Layers*: Pooling layers, which are comprised of pooling units, produce features that are resistant to information shifts in the image [51]. A pooling unit computes a function (e.g. max, mean, sum) of contiguous units from a feature map (or a few feature maps) [51]. Due to the nature of the data fed into the ConvNet, the proposed architecture does not



TABLE I

COMPARISON OF THE FIVE DATA AUGMENTATION TECHNIQUES PROPOSED. THE VALUES REPORTED ARE THE AVERAGE ACCURACIES FOR THE 19 PARTICIPANTS OVER 20 RUNS. THE WILCOXON SIGNED RANK TEST IS APPLIED TO COMPARE THE TRAINING OF THE CONVNET WITH AND WITHOUT ONE OF THE FIVE DATA AUGMENTATION TECHNIQUES. THE NULL HYPOTHESIS IS ACCEPTED WHEN  $H_0 = 1$  AND REJECTED WHEN  $H_0 = 0$  (WITH  $p = 0.05$ ).

	Baseline	Gaussian Noise	Muscle Fatigue	Electrode Displacement	Sliding Window	Aggregated Augmentation
Accuracy	95.62%	93.33%	95.75%	95.80%	<b>96.14%</b>	95.37%
STD	5.18%	7.12%	5.07%	4.91%	<b>4.93%</b>	5.27%
Rank	4	6	3	2	<b>1</b>	5
$H_0$	-	<b>0</b>	1	1	<b>0</b>	1

apply pooling layers. In the present context, learning spatially independent features would be detrimental as the position of the channels on the participant's forearm is important for predicting the hand/wrist gesture being performed. Additionally, pooling layers in the dimension of the frequency/scaling are also unused due to the limited sampling rate of the Myo and the small window size of the examples. In the case of high-frequency sEMG apparatus, 1D pooling layers on the frequency axis might actually lead to better results and faster inferences, but this would require further investigation.

2) *ConvNet for Spectrograms*: The spectrograms, which are fed to the ConvNet, were calculated with Hann windows of length 28 and an overlap of 20 yielding a matrix of 4x15. The first frequency band was removed in an effort to reduce baseline drift and motion artifact. As the armband features eight channels, eight such spectrograms were calculated, yielding a final matrix of 4x8x14 (after the axis swap described in Section V-D).

The implementation of the spectrogram ConvNet architecture, which is illustrated in Fig. 4 was created with Theano [52] in conjunction with Lasagne [53], allowing training and inference to be computed on a GPU. As usual in deep learning, the architecture was created in a trial and error process taking inspiration from previous architectures (primarily [4], [6], [49], [7]). The non-linear activation function is the parametric exponential linear unit (PELU) [54] in conjunction with PReLU [55]. ADAM [56] is utilized for the optimization of the ConvNet (with a learning rate of 0.00681292). The deactivation rate for MC Dropout is set at 0.5 and the batch size at 128. Finally, to further reduce overfitting, early stopping is employed by saving randomly 10% of the training data at the beginning of the optimization process as a validation set that will periodically test the network being trained. Note that learning rate annealing is applied with a factor of 5 when the validation loss stops improving. The training stops when two consecutive decays occurred with no network performance amelioration on the validation set. All hyperparameter values were found by a random search on the pre-training dataset.

3) *ConvNet for Continuous Wavelet Transforms*: The architecture for the CWT ConvNet, illustrated in Fig. 5, was built in a similar fashion as the spectrogram ConvNet one. Both the *Morlet* and *Mexican Hat* wavelet were considered for this work due to their previous application in EMG-related work [57], [58]. In the end, the Mexican Hat wavelet was selected, as it was the best performing during cross-validation on the *pre-training dataset*. The CWT were calculated with 32 scales yielding a 32x52 matrix. Downsampling is then

applied at a factor of 0.25 employing spline interpolation of order 0 to reduce the computational load of the ConvNet during training and inference. Following downsampling, the last row and column of the calculated CWT were removed as their noise-to-signal ratio were too high, similarly to the first frequency band of the spectrogram. The final matrix shape is thus 12x8x7 (after applying the axis swap described in Section V-D). The MC Dropout deactivation rate, batch size, optimization algorithm, and activation function remained unchanged. The learning rate was set at 0.0879923 (found by cross-validation).

#### E. Deep Learning on Embedded Systems

Within the context of sEMG-based gesture recognition, an important consideration is the feasibility of implementing the proposed ConvNets on embedded systems. As such, important efforts were deployed when designing the ConvNets architecture to ensure attainable implementation on currently available embedded systems. With the recent advent of deep learning, hardware systems particularly well suited for neural networks training/inference have been made commercially available. Graphics processing units (GPUs) such as the Nvidia Volta GV100 from *Nvidia* (50 GFLOPs/s/W) [59], field programmable gate arrays (FPGAs) such as the Stratix 10 from *Altera* (80 GFLOPs/s/W) [60] and mobile system-on-chips (SoCs) such as the Nvidia Tegra from *Nvidia* (100 GFLOPs/s/W) [61], are commercially available platforms that target the need for portable, computationally efficient and low-power systems for deep learning inference. Additionally, dedicated Application-Specific Integrated Circuits (ASICs) have arisen from research projects capable of processing ConvNet orders of magnitudes bigger than the ones proposed in this paper at a throughput of 35 frames/s at 278mW [62]. Pruning and quantizing network architectures are further ways to reduce the computational cost when performing inference with minimal impact on accuracy [63], [64].

Efficient CWT implementation employing the Mexican Hat wavelet has already been explored for embedded platforms [65]. These implementations are able to compute the CWT of larger input sizes than those required in this work in less than 1ms. Similarly, in [66], a robust time-frequency distribution estimation suitable for fast and accurate spectrogram computation is proposed. To generate a classification, the proposed CNN-Spectrogram and CNN-CWT architectures (including the transfer learning scheme proposed in Section VI) require approximately 14 728 000 and 2 274 000 floating

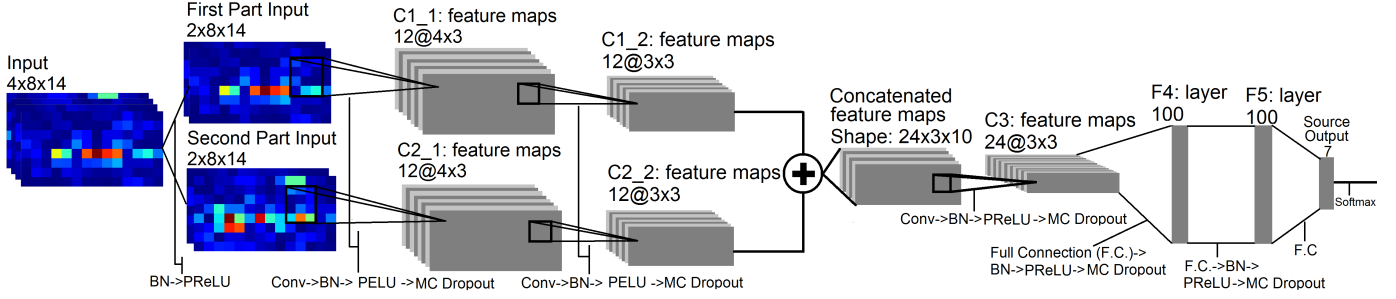


Fig. 4. The proposed spectrogram ConvNet architecture to leverage spectrogram examples employing 67 179 learnable parameters. To allow the slow fusion process, the input is first separated equally into two parts with respect to the time axis. The two branches are then fused together by element-wise summing the feature maps together. In this figure, *Conv* refer to *Convolution* and *F.C.* to *Fully Connected* layers.

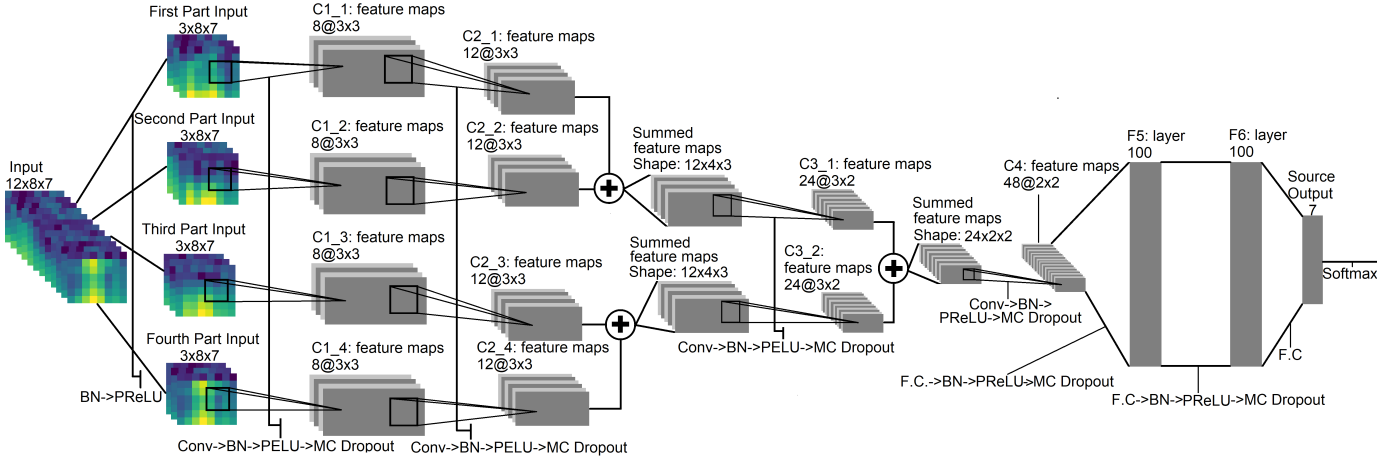


Fig. 5. The proposed CWT ConvNet architecture to leverage CWT examples using 30 219 parameters. To allow the slow fusion process, the input is first separated equally into four parts with respect to the time axis. The four branches are then slowly fused together by element-wise summing the feature maps together. In this figure, *Conv* refers to *Convolution* and *F.C.* to *Fully Connected* layers.

point operations (FLOPs) respectively. Considering a 40ms inference processing delay, hardware platforms of 3.5 and 0.5 GFLOPs/s/W will be suitable to implement a 100mW embedded system for sEMG classification. As such, adopting hardware-implementation approaches, along with state-of-the-art network compression techniques will lead to a power-consumption lower than 100mW for the proposed architectures, suitable for wearable applications.

## VI. TRANSFER LEARNING

One of the main advantages of deep learning comes from its ability to leverage large amounts of data for learning. In the context of this work, it would be too time-consuming for a single individual to record a sufficient amount of data to generate tens of thousands of examples. However, by aggregating the data of multiple individuals, such a goal is easily attainable. The main challenge thus becomes to find a way to leverage data from multiple users, with the objective of achieving higher accuracy with less data. Transfer learning techniques are well suited for such a task, allowing the ConvNets to generate more general and robust features that can be applied to a new subject's sEMG activity.

As the method employed to record the data was purposefully as unconstrained as possible, the armband's orientation from one subject to another can vary widely. As such, to allow for

the use of transfer learning, automatic *alignment* is necessary as a first step. The alignment for each subject was made by identifying the most active channel for each gesture on the first subject. On subsequent subjects, the channels were then circularly shifted until their activation for each gesture matched those of the first subject as closely as possible.

### A. Progressive Neural Networks

Fine-tuning is the most prevalent transfer learning technique in deep learning [67], [68]. It consists of training a model on a *source domain* (abundance of labeled data) and using the trained weights as a starting point when presented with a new task. However, fine-tuning can suffer from *catastrophic forgetting* [69] where relevant and important features learned during pre-training are lost on the *target domain* (i.e. new task). Moreover, by design, fine-tuning is ill-suited when significant differences exist between the source and the target, as it can bias the network into poorly adapted features for the task at hand. Progressive Neural Networks (PNN) [69] attempt to address these issues by pre-training a model on the source domain and freezing its weights. When a new task appears, a new network, with random initialization, is created and connected in a layer-wise fashion to the original network. This connection is done via non-linear lateral connections. For convolutional layers, a dimensionality reduction is performed

through 1x1 convolutions, whereas a multi-layer perceptron is utilized for fully connected layers (detailed in [69]).

### B. Adaptive Batch Normalization

In the paper that introduces the PNN architecture, multiple source tasks are each assigned a separate network. Each of the *Source Networks* are connected together in a layer-wise fashion through a special dimensionality reduction layer called *adapter* (detailed in [69]). Each adapter layer's output is then fed to the corresponding layer of the target's network. In the present context, this solution is unscalable as each new participant would require their own network.

This problem is addressed by applying AdaBatch [70]. In opposition to the PNN architecture, which uses a different network for the source and the target, AdaBatch employs the same network for both tasks. The transfer learning occurs by freezing all the network's weights (learned during pre-training) when training on the target, except for the parameters associated with BN. The hypothesis behind this technique is that the label-related information (i.e. gestures) rests in the network model weights whereas the domain-related information (i.e. subjects) is stored in their BN statistic. In the present context, this idea can be generalized by applying a multi-stream AdaBatch scheme [6]. Instead of employing one *Source Network* per subject during pre-training, a single network is shared across all participants. However, the BN statistics from each subject are calculated independently from one another, allowing the ConvNet to extract more general and robust features across all participants. By fixing the parameters of the network, except those related to BN, the model is then able to naturally adapt to a new user without modifying the connection weights of the original model. Note that a single-stream scheme (i.e. all subjects share statistics and BN parameters are also frozen on the *Source Network*) was also tried. As expected, this scheme's performances started to rapidly worsen as the number of source participants augmented, lending more credence to the initial AdaBatch hypothesis.

### C. Proposed Transfer Learning Architecture

The main tenet behind transfer learning is that similar tasks can be completed in similar ways. The difficulty in this paper's context is then to learn a mapping  $F(x)$  between the source and target task as to leverage information learned during pre-training. Training one network per source-task (i.e. per participant) for the PNN is not scalable in the present context. However, by augmenting the PNN architecture with the multi-stream AdaBatch, the scaling problem in the current context vanishes. The proposed transfer learning scheme thus consists of a single *Source Network* (presented in Section V) shared across all participants from the *pre-training dataset*, along with a second network to learn on the target task. This second network will hereafter be referred to as the *Second Network*. The architecture of the *Second Network* is almost identical to the *Source Network*. The difference being in the activation functions employed. The *Source Network* leveraged a combination of PReLU and PELU (see Fig. V-D and Fig. 5 for details), whereas the *Second Network* only

employed PELU. This architecture choice was made through trial and error and cross-validation on the training dataset. Additionally, the weights of both networks are trained and initialized independently. During pre-training, only the *Source Network* is trained to represent the information of all the participants in the *pre-training dataset*. The parameters of the *Source Network* are then frozen once pre-training is completed, except for the BN parameters as they represent the domain-related information and thus must retain the ability to adapt to new users.

Due to the application of the multi-stream AdaBatch scheme, the source task in the present context is to learn the *general* mapping between muscle activity and gestures. One can see the problem of learning such mapping between the target and the source task as learning a residual of the source task. For this reason, the *Source Network* shares information with the *Second Network* through an element-wise summation in a layer-by-layer fashion (see Fig. 6). The idea behind the merging of information through element-wise summation is two-fold. The first reason is that compared to concatenating the features maps (as in [7]) or employing non-linear lateral connections (like in [69]), element-wise summation minimizes the computational impact of connecting the *Source Network* and the *Second Network* together. As such, the deployment of the proposed algorithm on embedded systems would be made easier. The second reason is to provide a mechanism that fosters residual learning as inspired by Residual Networks [71]. Thus, the *Second Network* only needs to learn weights that express the difference between the new target and source task. All outputs from the *Source Network* layers to the *Second Network* are multiplied by learnable coefficients before the sum-connection. This scalar layer provides an easy mechanism to neuter the *Source Network's* influence on a layer-wise level. This is particularly useful if the new target task is so different that for some layers the information from the *Source Network* actually hinders learning.

The combination of the *Source Network* and *Second Network* will hereafter be referred to as the *Target Network*. An overview of the final proposed architecture is presented in Fig. 6. During training of the *Source Network* (i.e. pre-training), MC Dropout rate is set at 35% and when training the *Target Network* the rate is set at 50%. Note that different architecture choices for the *Source Network* and *Second Network* were required to augment the performance of the system as a whole. This seems to indicate that the two tasks (i.e. learning a general mapping of hand gestures and learning a specific mapping), might be different enough that even greater differentiation through specialization of the two networks might increase the performance further.

## VII. CLASSIFIERS COMPARISON ON THE NOVEL DATASET

A comparison between the proposed deep learning models and a variety of classifiers trained on features set that were found to be state-of-art for sEMG-based gesture classification (Section IV-A for details) is given in Table II. The accuracies are given for one, two, three and four cycles of training. For conciseness, only the best performing classifier is reported for each type of features set and training cycles.



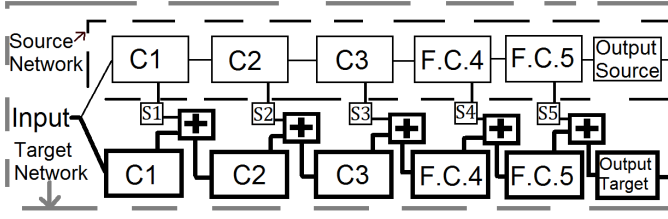


Fig. 6. The PNN-inspired architecture. This figure represents the case with the spectrogram ConvNet (see Fig. 4 for the *Source Network* details). Note that the transfer learning behavior is the same for the CWT-based ConvNet. C1,2,3 and F.C.4,5 correspond to the three stages of convolutions and two stages of fully connected layers respectively. The  $S_i$  ( $i=1..5$ ) boxes represent a layer that scales its inputs by learned coefficients. The number of learned coefficients in one layer is the number of channels or the number of neurons for the convolutional and fully connected layers respectively. For clarity's sake, the slow fusion aspect is omitted from the representation although they are present on both networks (i.e. both for the spectrogram and CWT-based ConvNet). The + boxes represent the merging through an element-wise summation of the ConvNets' corresponding layers. The *Target Network* contains 133 695 and 59 799 learnable parameters for the spectrogram and CWT-based ConvNet respectively.

In this section, as in Section V-A, the Wilcoxon signed rank test is applied to compare the methods. Table II shows that, in all cases, the CWT ConvNet significantly ( $p < 0.05$ ) outperformed its Spectrogram variant, suggesting that CWT might be better suited to represent sEMG signals. Additionally, the ConvNets augmented with the proposed transfer learning scheme significantly outperformed ( $p < 0.05$ ) their non-augmented version both for the CWT and Spectrogram-based classifiers. Unsurprisingly, reducing the amount of training data (cycles) systematically degraded the performance of all tested methods, with the non-transfer learning ConvNets being the most adversely affected. This is likely due to their overfitting stemming from the size of the dataset. However, it is worth noting that, when using a single cycle of training, augmenting the ConvNets with the proposed transfer learning scheme significantly improves their accuracies. In fact, with this addition, the accuracies of the ConvNets go from the lowest to the highest of all methods. In addition, for the Transfer Learning CWT ConvNet, a single training cycle is sufficient to outperform almost all the tested classification methods (Enhanced TD (LDA) being the exception) with double the amount of training examples. Overall, the Transfer Learning CWT ConvNet significantly ( $p < 0.05$ ) outperformed all the other 22 classification methods (i.e. four features sets tested on five different classifiers plus the two non-augmented ConvNets), regardless of the amount of training data provided, except for the Enhanced TD (LDA) with four cycles of training.

### VIII. NINAPro DB5

The *NinaPro DB5* is a dataset built to benchmark sEMG-based gesture recognition algorithms [39]. This dataset, which was recorded with the Myo Armband, contains data from 10 able-bodied participants performing a total of 53 different movements (including neutral) divided into three exercise sets. The second exercise set, which contains 18 gestures (including neutral), is of particular interest, as it includes all the gestures considered so far in this work and an additional 11 gestures.

These additional 11 gestures which are presented in [72] include wrist pronation, wrist supination and diverse finger extension amongst others. While this particular dataset was recorded with two Myo Armband, only the lower armband is considered as to allow direct comparison to the precedent dataset. Note that the increased amount of gestures in this dataset is expected to decrease the accuracy obtained across all the classification methods.

#### A. Data Acquisition and Processing

Each participant was asked to hold a gesture for five seconds following by three seconds of rest and to repeat this action five more times (total of six repetitions). This procedure was repeated for all the movements contained within the dataset. The first four repetitions serve as the training set and the last two as the test set.

In this section, all data processing (e.g. window size, window overlap, feature calculation) are exactly as described in the previous sections. The four previously presented feature set will be compared against the CWT ConvNet and the Transfer Learning CWT ConvNet.

#### B. Classification Method Comparison

The spectrograms-based ConvNets are omitted for concision sake as they appeared to perform strictly worst than their CWT variant on the *Evaluation Dataset*. Performances based on two different pre-training are investigated. The first one uses the pre-training from the previous section, while the second will pre-train on the *NinaPro DB5* dataset. In the latter case, the pre-training for each participant is done employing the training sets of the remaining nine participants.

Table III presents the average accuracy obtained for each classification methods over the ten participants. Similarly to Section VII, the proposed ConvNet architecture is competitive with the state of the art. Furthermore, the proposed transfer learning approaches significantly outperform all the other tested classification techniques regardless of the amount of training data employed (Wilcoxon signed rank test  $p < 0.05$ ). Note that this significant difference in performance between the transfer learning and non-transfer learning augmented classifiers holds true for both the *Transfer Learning on NinaPro DB5 CWT ConvNet* and the *Transfer Learning on pre-training dataset CWT ConvNet*.

#### C. Out-of-Sample Gestures

A final test involving the *NinaPro DB5* was conducted to evaluate the impact on the proposed transfer learning algorithm when the target is comprised solely of Out-of-sample gestures (i.e. never-seen-before gestures). To do so, the proposed CWT ConvNet was trained and evaluated on the train and test set of the *NinaPro DB5* as described before, but considering only the gestures that were absent from the *pre-training dataset* (11 gestures total). The CWT ConvNet was then compared to its transfer learning augmented version which was pre-trained on the *pre-training dataset*. Fig 7 presents the accuracies obtained for the two classifiers with different number of repetitions

TABLE II

CLASSIFICATION ACCURACY ON THE *Evaluation Dataset* WITH RESPECT TO THE NUMBER OF TRAINING CYCLES PERFORMED. THE *Wilcoxon signed rank test* IS APPLIED TO COMPARE THE *Transfer Learning CWT ConvNet* AGAINST THE OTHER CLASSIFICATION TECHNIQUES. NULL HYPOTHESIS IS REJECTED WHEN  $H_0 = 0$  ( $p < 0.05$ ).

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	Spectrogram ConvNet	CWT ConvNet	Transfer Learning Spectrogram ConvNet	Transfer Learning CWT ConvNet
4 Cycles	97.76% (LDA)	98.14% (LDA)	97.58% (LDA)	97.72% (LDA)	97.14%	97.95%	97.85%	<b>98.31%</b>
STD	2.63%	2.21%	2.74%	2.60%	2.70%	2.26%	2.14%	<b>1.86%</b>
Rank	5	2	7	6	8	3	4	<b>1</b>
H0	0	1	0	0	0	0	0	-
3 Cycles	96.26% (NN)	97.33% (LDA)	96.54% (NN)	96.51% (NN)	96.33%	97.22%	97.40%	<b>97.82%</b>
STD	6.07%	3.24%	3.62%	5.65%	3.45%	2.95%	2.78%	<b>2.50%</b>
Rank	8	3	5	6	7	4	2	<b>1</b>
H0	0	0	0	0	0	0	0	-
2 Cycles	94.12% (NN)	94.79% (LDA)	93.82% (NN)	94.64% (NN)	94.19%	95.17%	96.05%	<b>96.63%</b>
STD	8.98%	7.82%	7.42%	8.24%	5.86%	5.68%	4.70%	<b>4.67%</b>
Rank	7	4	8	5	6	3	2	<b>1</b>
H0	0	0	0	0	0	0	0	-
1 Cycle	90.62% (NN)	91.25% (LDA)	90.13% (LDA)	91.08% (NN)	88.51%	89.02%	93.73%	<b>94.69%</b>
STD	9.08%	9.44%	7.73%	8.88%	8.04%	9.75%	5.75%	<b>5.22%</b>
Rank	5	3	6	4	8	7	2	<b>1</b>
H0	0	0	0	0	0	0	0	-

\*For brevity's sake, only the best performing classifier for each feature set is reported (indicated in parenthesis).

\*\*All results report the average accuracy over all subjects of the evaluation dataset over 20 runs.

\*\*\*The STD represents the standard variation in accuracy over the 17 participants.

TABLE III

CLASSIFICATION ACCURACY ON THE *NinaPro DB5* WITH RESPECT TO THE NUMBER OF REPETITIONS USED DURING TRAINING. THE *Wilcoxon signed rank test* IS APPLIED TO COMPARE THE *Transfer Learning on NinaPro DB5 CWT ConvNet* AGAINST THE OTHER CLASSIFICATION TECHNIQUES. NULL HYPOTHESIS IS REJECTED WHEN  $H_0 = 0$  ( $p < 0.05$ ).

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	CWT ConvNet	Transfer Learning on pre-training dataset CWT ConvNet	Transfer Learning on NinaPro DB5 CWT ConvNet
4 Cycles	59.91% (RF)	59.57% (RF)	56.72%(RF)	62.30% (RF)	61.89%	64.33%	<b>65.57%</b>
STD	3.31%	4.20%	3.80%	3.70%	3.77%	3.75%	<b>3.43%</b>
Rank	5	6	7	3	4	2	<b>1</b>
H0	0	0	0	0	0	0	-
3 Cycles	55.73% (RF)	55.32% (RF)	52.33% (RF)	58.24% (RF)	58.37%	60.78%	<b>62.21%</b>
STD	3.52%	4.25%	4.40%	4.00%	3.82%	3.88%	<b>3.67%</b>
Rank	5	6	7	4	3	2	<b>1</b>
H0	0	0	0	0	0	0	-
2 Cycles	50.85% (RF)	50.08% (LDA)	46.85% (LDA)	53.00% (RF)	53.32%	56.20%	<b>57.34%</b>
STD	4.04%	4.63%	4.81%	3.65%	3.16%	3.68%	<b>3.43%</b>
Rank	5	6	7	4	3	2	<b>1</b>
H0	0	0	0	0	0	0	-
1 Cycle	40.70% (RF)	40.86% (LDA)	37.60% (LDA)	42.26% (LDA)	42.47%	47.13%	<b>48.33%</b>
STD	5.52%	6.91%	6.67%	5.78%	6.36%	4.85%	<b>4.75%</b>
Rank	6	5	7	4	3	2	<b>1</b>
H0	0	0	0	0	0	0	-

\*For brevity's sake, only the best performing classifier for each feature set is reported (indicated in parenthesis).

\*\*All results report the average accuracy over all subjects of the evaluation dataset over 20 runs.

\*\*\*The STD represents the standard variation in accuracy over the 10 participants.

employed for training. In all cases, the CWT ConvNet classification accuracy was worst than its transfer learning-augmented counterpart.

Furthermore, regardless of the number of repetitions employed for training, the difference in accuracy of the two methods was deemed significant by the Wilcoxon Signed Rank Test ( $p < 0.05$ ). This suggests that the proposed transfer learning algorithm enables the network to learn features that can generalize not only across participants but also for never-seen-before gestures. As such, the weights learned from the *pre-training dataset* can easily be re-used for other work that employs the Myo Armband with different gestures.

## IX. MEDIUM TERM PERFORMANCES (CASE STUDY)

This last experiment section proposes a use-case study of the performance of the classifier over a period of 14 days for eight able-bodied participants. In previous literature, it has been shown that, when no re-calibration occurs, the performance of a classifier degrades over time due to the non-stationary property of sEMG signals [73]. The main goal of this use-case experiment is to evaluate if giving visual feedback from the classifier to the user can influence this degradation.

To achieve this, on the first day, each participant recorded a training set as described in Section III. Then, over the next fourteen days, a daily *session* was recorded based on the participant's availability. A *session* consisted in holding a set

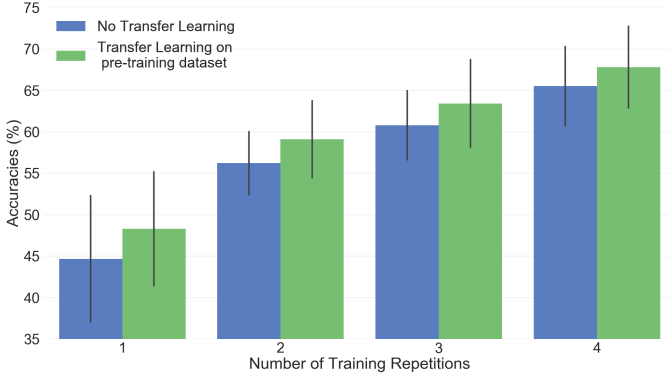


Fig. 7. Classification accuracy on the *NinaPro DB5* with respect to the number of repetitions employed during training. The pre-training was done using the *pre-training dataset*. Training and testing only considered the eleven gestures from the *NinaPro DB5* not included in the *pre-training dataset*.

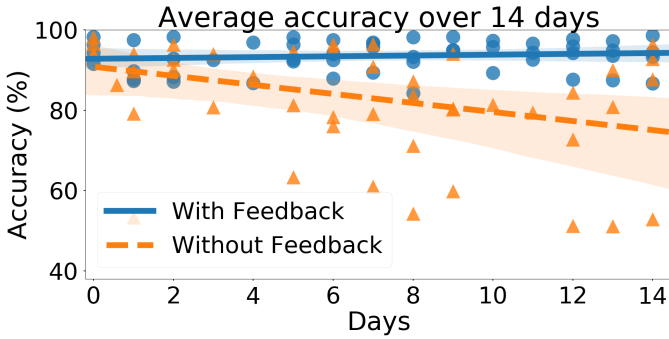


Fig. 8. Average accuracy over 14 days without recalibration of the Transfer Learning CWT ConvNet for participants belonging to the *Feedback* and *Without Feedback* groups. The blue circles represent data from the *Feedback* group whereas the orange triangles represent data from the *Without Feedback* group. The number of data points generated by a single participant varies between 10 and 16 depending on the participant's availability during the experiment period. The translucent bands around the linear regressions represent the confidence interval (95%) estimated by bootstrap.

of 30 randomly selected gestures (among the seven shown in Fig. 1) for ten seconds each, resulting in five minutes of continuous sEMG data. During the experiment, the eight participants were randomly separated into two equal groups. The first group, referred to as the *Without Feedback* group, did not receive any classifier feedback during the experiment. The second group referred to as the *Feedback* group, received real-time feedback of the gesture prediction of the classifier in the form of text displayed on a screen. The classifier employed in this experiment is the *Transfer Learning CWT ConvNet*, as it was the best performing classifier tested in this paper.

As it can be observed in Fig. 8, while the *Without Feedback* group did experience accuracy degradation over the 14 days, the *Feedback* group was seemingly able to counteract this degradation. Note that the difference in accuracy compared to the one in Section VII is likely due to the small sample size and, more importantly, the inclusion of computer-specified transition that necessarily reduce the accuracy (i.e. the latency between the computer asking for (and expecting) a new gesture from the human and the participant reading, assessing and finally performing the gesture).

Many participants reported experiencing muscular fatigue

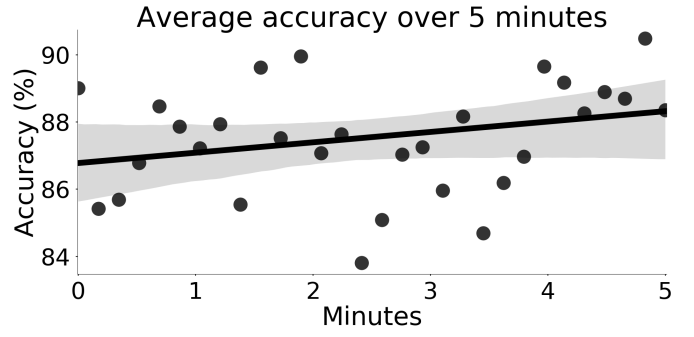


Fig. 9. The average accuracy of the eight participants over all the five minute sessions recorded to evaluate the effect of muscle fatigue on the classifier performance. During each session of the experiment, participants were asked to hold a total of 30 random gestures for ten seconds each. As such, a dot represents the average accuracy across all participants over one of the ten second periods. The translucent bands around the linear regression represent the confidence intervals (95%) estimated by bootstrap.

during the recording of both this experiment and the *evaluation dataset*. As such, in an effort to quantify the impact of muscle fatigue on the classifier's performance, the average accuracy of the eight participants over the five minute session is computed as a function of time. As it can be observed from the positive slope of the linear regression presented in Fig. 9, muscle fatigue, at least during a five minute period of intensive activity, does not seem to negatively affect the proposed ConvNet's accuracy.

## X. CONCLUSION

This paper presents two novels ConvNet architectures that were shown to be competitive with current sEMG-based classifiers. Moreover, this work presents a new transfer learning scheme that, when coupled with the CWT-based ConvNet, outperforms the state of the art. On the newly proposed *evaluation dataset*, the transfer learning augmented ConvNet achieves an average accuracy of 98.31%, for the recognition of 7 hand/wrist gestures in 17 participants, despite a limited amount of target data. Furthermore, on the *NinaPro DB5* dataset, the proposed classifier achieved an average accuracy of 65.57% for the recognition of 18 hand/wrist gestures in 10 participants on a single Myo Armband. This dataset showed that the proposed transfer learning algorithm learns sufficiently general features to significantly enhance the performance of the ConvNet on out-of-sample gestures. Thus, showing that deep learning algorithms can be efficiently trained within the inherent constraints of sEMG-based hand gesture recognition offers exciting new research avenues for this field.

Furthermore, as shown in Section IX, important degradation of the proposed system occurs over time when no recalibration is performed. As such, future works will focus on leveraging unsupervised domain adaptation methods [74] to alleviate the need for periodic re-calibration. These improvements will be pursued with the goal of applying the system to the control of a robotic arm/myoelectric prosthetic. Additionally, important efforts will be deployed on adapting and testing the proposed transfer learning algorithm on upper-extremity amputees. This will provides additional challenges

due to the greater muscle variability across amputees and the decrease in classification accuracy compared to able-bodied participants [72].

## XI. ACKNOWLEDGEMENTS

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), [401220434] and the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST).

Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG), [401220434] et l'Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST).

## REFERENCES

- [1] M. A. Oskoei and H. Hu, "Myoelectric control systems a survey," *Biomedical Signal Processing and Control*, vol. 2, no. 4, pp. 275–294, 2007.
- [2] A. Phinyomark, S. Hirunviriya, C. Limsakul, and P. Phukpattaranont, "Evaluation of emg feature extraction for hand movement recognition based on euclidean distance and standard deviation," in *Electrical Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010 International Conference on*. IEEE, 2010, pp. 856–860.
- [3] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for emg signal classification," *Expert Systems with Applications*, vol. 39, no. 8, pp. 7420–7431, 2012.
- [4] U. C. Allard, F. Nougrou, C. L. Fall, P. Giguère, C. Gosselin, F. Laviolette, and B. Gosselin, "A convolutional neural network for robotic arm guidance using semg based frequency-features," in *Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 2464–2470.
- [5] 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*, vol. 10, 2016.
- [6] Y. Du, W. Jin, W. Wei, Y. Hu, and W. Geng, "Surface emg-based inter-session gesture recognition enhanced by deep domain adaptation," *Sensors*, vol. 17, no. 3, p. 458, 2017.
- [7] U. Côté-Allard, C. L. Fall, A. Campeau-Lecours, C. Gosselin, F. Laviolette, and B. Gosselin, "Transfer learning for semg hand gestures recognition using convolutional neural networks," in *Systems, Man, and Cybernetics, 2017 IEEE International Conference on (in press)*. IEEE, 2017.
- [8] S. Karlsson, J. Yu, and M. Akay, "Time-frequency analysis of myoelectric signals during dynamic contractions: a comparative study," *IEEE transactions on Biomedical Engineering*, vol. 47, no. 2, pp. 228–238, 2000.
- [9] C. Castellini, A. E. Fiorilla, and G. Sandini, "Multi-subject/daily-life activity emg-based control of mechanical hands," *Journal of neuroengineering and rehabilitation*, vol. 6, no. 1, p. 41, 2009.
- [10] R. N. Khushaba, "Correlation analysis of electromyogram signals for multiuser myoelectric interfaces," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 22, no. 4, pp. 745–755, 2014.
- [11] R. Chattopadhyay, N. C. Krishnan, and S. Panchanathan, "Topology preserving domain adaptation for addressing subject based variability in semg signal," in *AAAI Spring Symposium: Computational Physiology*, 2011, pp. 4–9.
- [12] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo, "Improving control of dexterous hand prostheses using adaptive learning," *IEEE Transactions on Robotics*, vol. 29, no. 1, pp. 207–219, 2013.
- [13] N. Patricia, T. Tommasi, and B. Caputo, "Multi-source adaptive learning for fast control of prosthetics hand," in *Pattern Recognition (ICPR), 2014 22nd International Conference on*. IEEE, 2014, pp. 2769–2774.
- [14] F. Orabona, C. Castellini, B. Caputo, A. E. Fiorilla, and G. Sandini, "Model adaptation with least-squares svm for adaptive hand prosthetics," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*. IEEE, 2009, pp. 2897–2903.
- [15] V. Gregori, A. Gijsberts, and B. Caputo, "Adaptive learning to speed-up control of prosthetic hands: A few things everybody should know," in *Rehabilitation Robotics (ICORR), 2017 International Conference on*. IEEE, 2017, pp. 1130–1135.
- [16] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Medical engineering & physics*, vol. 21, no. 6, pp. 431–438, 1999.
- [17] G. Tsenov, A. Zeghib, F. Palis, N. Shoylev, and V. Mladenov, "Neural networks for online classification of hand and finger movements using surface emg signals," in *Neural Network Applications in Electrical Engineering, 2006. NEUREL 2006. 8th Seminar on*. IEEE, 2006, pp. 167–171.
- [18] S. Karlsson and B. Gerdle, "Mean frequency and signal amplitude of the surface emg of the quadriceps muscles increase with increasing torque study using the continuous wavelet transform," *Journal of electromyography and kinesiology*, vol. 11, no. 2, pp. 131–140, 2001.
- [19] A. R. Ismail and S. S. Asfour, "Continuous wavelet transform application to emg signals during human gait," in *Signals, Systems & Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, vol. 1. IEEE, 1998, pp. 325–329.
- [20] P. S. Addison, "Wavelet transforms and the ecg: a review," *Physiological measurement*, vol. 26, no. 5, p. R155, 2005.
- [21] O. Faust, U. R. Acharya, H. Adeli, and A. Adeli, "Wavelet-based eeg processing for computer-aided seizure detection and epilepsy diagnosis," *Seizure*, vol. 26, pp. 56–64, 2015.
- [22] K. Englehart, B. Hudgin, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 48, no. 3, pp. 302–311, 2001.
- [23] C. Toledo, R. Muñoz, and L. Leija, "semg signal detector using discrete wavelet transform," in *Health Care Exchanges (PAHCE), 2012 Pan American*. IEEE, 2012, pp. 62–65.
- [24] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface emg images," *Scientific reports*, vol. 6, p. 36571, 2016.
- [25] D. Stegeman and B. L. B.U. Kleine, "High-density surface emg: Techniques and applications at a motor unit level," *Biocybernetics and Biomedical Engineering*, vol. 32, no. 3, 2012.
- [26] R. Merletti and P. Di Torino, "Standards for reporting emg data," *J Electromyogr Kinesiol*, vol. 9, no. 1, pp. 3–4, 1999.
- [27] A. Phinyomark, R. N. Khushaba, and E. Scheme, "Feature extraction and selection for myoelectric control based on wearable emg sensors," *Sensors*, vol. 18, no. 5, p. 1615, 2018.
- [28] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Transactions on Biomedical Engineering*, vol. 40, no. 1, pp. 82–94, 1993.
- [29] T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Transactions on neural systems and rehabilitation engineering*, vol. 15, no. 1, pp. 111–118, 2007.
- [30] B. Peerdeman, D. Boere, H. Witteveen, R. H. Veld, H. Hermens, S. Stramigioli *et al.*, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *Journal of Rehabilitation Research & Development*, vol. 48, no. 6, p. 719, 2011.
- [31] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "Emg feature evaluation for improving myoelectric pattern recognition robustness," *Expert Systems with Applications*, vol. 40, no. 12, pp. 4832–4840, 2013.
- [32] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE transactions on biomedical engineering*, vol. 50, no. 7, pp. 848–854, 2003.
- [33] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, p. 140053, 2014.
- [34] R. N. Khushaba and S. Kodagoda, "Electromyogram (emg) feature reduction using mutual components analysis for multifunction prosthetic fingers control," in *Control Automation Robotics & Vision (ICARCV), 2012 12th International Conference on*. IEEE, 2012, pp. 1534–1539.
- [35] M. R. Ahsan, M. I. Ibrahimy, O. O. Khalifa *et al.*, "Emg signal classification for human computer interaction: a review," *European Journal of Scientific Research*, vol. 33, no. 3, pp. 480–501, 2009.
- [36] R. N. Khushaba, S. Kodagoda, M. Takuri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (emg) signals," *Expert Systems with Applications*, vol. 39, no. 12, pp. 10731–10738, 2012.
- [37] D. Zhang, X. Zhao, J. Han, and Y. Zhao, "A comparative study on pca and lda based emg pattern recognition for anthropomorphic robotic hand," in *Robotics and Automation (ICRA), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4850–4855.

- [38] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
- [39] S. Pizzolato, L. Tagliapietra, M. Cognolato, M. Reggiani, H. Müller, and M. Atzori, "Comparison of six electromyography acquisition setups on hand movement classification tasks," *PloS one*, vol. 12, no. 10, p. e0186132, 2017.
- [40] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *Journal of machine learning research*, vol. 15, no. 1, pp. 1929–1958, 2014.
- [41] Y. Gal and Z. Ghahramani, "Dropout as a bayesian approximation: Representing model uncertainty in deep learning," in *international conference on machine learning*, 2016, pp. 1050–1059.
- [42] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International Conference on Machine Learning*, 2015, pp. 448–456.
- [43] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.
- [44] S. Dieleman, A. van den Oord, I. Korshunova, J. Burms, J. Degraeve, L. Pigou, and P. Buteneers, "Classifying plankton with deep neural networks," *UR L* <http://benanne.github.io/2015/03/17/plankton.html>, 2015.
- [45] J. H. Hollman, J. M. Hohl, J. L. Kraft, J. D. Strauss, and K. J. Traver, "Does the fast fourier transformation window length affect the slope of an electromyogram's median frequency plot during a fatiguing isometric contraction?" *Gait & posture*, vol. 38, no. 1, pp. 161–164, 2013.
- [46] F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics bulletin*, vol. 1, no. 6, pp. 80–83, 1945.
- [47] Y. Gal and Z. Ghahramani, "Bayesian convolutional neural networks with bernoulli approximate variational inference," *arXiv preprint arXiv:1506.02158*, 2015.
- [48] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *International Workshop on Human Behavior Understanding*. Springer, 2011, pp. 29–39.
- [49] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2014, pp. 1725–1732.
- [50] H. Guo, G. Wang, and X. Chen, "Two-stream convolutional neural network for accurate rgb-d fingertip detection using depth and edge information," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 2608–2612.
- [51] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.
- [52] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau *et al.*, "Theano: A python framework for fast computation of mathematical expressions," *arXiv preprint arXiv:1605.02688*, 2016.
- [53] S. Dieleman, J. Schlter, C. Raffel, E. Olson, S. K. Snderby *et al.*, "Lasagne: First release," Aug. 2015. [Online]. Available: <http://dx.doi.org/10.5281/zenodo.27878>
- [54] L. Trottier, P. Giguère, and B. Chaib-draa, "Parametric exponential linear unit for deep convolutional neural networks," *arXiv preprint arXiv:1605.09332*, 2016.
- [55] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1026–1034.
- [56] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.
- [57] M. B. Reaz, M. Hussain, and F. Mohd-Yasin, "Techniques of emg signal analysis: detection, processing, classification and applications," *Biological procedures online*, vol. 8, no. 1, pp. 11–35, 2006.
- [58] R. Reynolds and M. Lakie, "Postmovement changes in the frequency and amplitude of physiological tremor despite unchanged neural output," *Journal of neurophysiology*, vol. 104, no. 4, pp. 2020–2023, 2010.
- [59] Y. Chen, H. H. Li, C. Wu, C. Song, S. Li, C. Min, H.-P. Cheng, W. Wen, and X. Liu, "Neuromorphic computing's yesterday, today, and tomorrow—an evolutionary view," *Integration, the VLSI Journal*, 2017.
- [60] E. Nurvitadhi, G. Venkatesh, J. Sim, D. Marr, R. Huang, J. Ong Gee Hock, Y. T. Liew, K. Srivatsan, D. Moss, S. Subhaschandra *et al.*, "Can fpgas beat gpus in accelerating next-generation deep neural networks?" in *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*. ACM, 2017, pp. 5–14.
- [61] L. Cavigelli, M. Magno, and L. Benini, "Accelerating real-time embedded scene labeling with convolutional networks," in *Design Automation Conference (DAC), 2015 52nd ACM/EDAC/IEEE*. IEEE, 2015, pp. 1–6.
- [62] Y.-H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE Journal of Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, 2017.
- [63] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: efficient inference engine on compressed deep neural network," in *Proceedings of the 43rd International Symposium on Computer Architecture*. IEEE Press, 2016, pp. 243–254.
- [64] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 4820–4828.
- [65] Y. T. Qassim, T. R. Cutmore, and D. D. Rowlands, "Optimized fpga based continuous wavelet transform," *Computers & Electrical Engineering*, vol. 49, pp. 84–94, 2016.
- [66] N. Žarić, S. Stanković, and Z. Uskoković, "Hardware realization of the robust time–frequency distributions," *annals of telecommunications-Annales des télécommunications*, vol. 69, no. 5-6, pp. 309–320, 2014.
- [67] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *ICML Unsupervised and Transfer Learning*, vol. 27, pp. 17–36, 2012.
- [68] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems*, 2014, pp. 3320–3328.
- [69] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," *arXiv preprint arXiv:1606.04671*, 2016.
- [70] Y. Li, N. Wang, J. Shi, J. Liu, and X. Hou, "Revisiting batch normalization for practical domain adaptation," *arXiv preprint arXiv:1603.04779*, 2016.
- [71] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.
- [72] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G. M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific data*, vol. 1, p. 140053, 2014.
- [73] J. Liu, X. Sheng, D. Zhang, J. He, and X. Zhu, "Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation," *IEEE journal of biomedical and health informatics*, vol. 20, no. 1, pp. 166–176, 2016.
- [74] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *Journal of Machine Learning Research*, vol. 17, no. 59, pp. 1–35, 2016.
- [75] B. Hjorth, "Eeg analysis based on time domain properties," *Electroencephalography and clinical neurophysiology*, vol. 29, no. 3, pp. 306–310, 1970.
- [76] M. Mouzé-Amady and F. Horwat, "Evaluation of hjorth parameters in forearm surface emg analysis during an occupational repetitive task," *Electroencephalography and Clinical Neurophysiology/Electromyography and Motor Control*, vol. 101, no. 2, pp. 181–183, 1996.
- [77] X. Zhang and P. Zhou, "Sample entropy analysis of surface emg for improved muscle activity onset detection against spurious background spikes," *Journal of Electromyography and Kinesiology*, vol. 22, no. 6, pp. 901–907, 2012.
- [78] M. Zardoshti-Kermani, B. C. Wheeler, K. Badie, and R. M. Hashemi, "Emg feature evaluation for movement control of upper extremity prostheses," *IEEE Transactions on Rehabilitation Engineering*, vol. 3, no. 4, pp. 324–333, 1995.
- [79] W.-J. Kang, J.-R. Shiu, C.-K. Cheng, J.-S. Lai, H.-W. Tsao, and T.-S. Kuo, "The application of cepstral coefficients and maximum likelihood method in emg pattern recognition [movements classification]," *IEEE Transactions on Biomedical Engineering*, vol. 42, no. 8, pp. 777–785, 1995.
- [80] M.-F. Lucas, A. Gaufriaux, S. Pascual, C. Doncarli, and D. Farina, "Multi-channel surface emg classification using support vector machines and signal-based wavelet optimization," *Biomedical Signal Processing and Control*, vol. 3, no. 2, pp. 169–174, 2008.

- [81] D. Gabor, "Theory of communication. part 1: The analysis of information," *Journal of the Institution of Electrical Engineers-Part III: Radio and Communication Engineering*, vol. 93, no. 26, pp. 429–441, 1946.
- [82] M. Teplan *et al.*, "Fundamentals of eeg measurement," *Measurement science review*, vol. 2, no. 2, pp. 1–11, 2002.
- [83] R. C. Gonzalez, *Digital image processing*, 1977.
- [84] A. Graps, "An introduction to wavelets," *IEEE computational science and engineering*, vol. 2, no. 2, pp. 50–61, 1995.

## APPENDIX

This section presents the features employed in this work. Unless specified otherwise, features are calculated by dividing the signal  $x$  into overlapping windows of length  $L$ . The  $k$ th element of the  $i$ th window then corresponds to  $x_{i,k}$ .

### A. Time Domain Features

1) *Mean Absolute Value (MAV)* [32]: A feature returning the mean of a fully-rectified signal.

$$\text{MAV}(x_i) = \frac{1}{L} \sum_{k=1}^L |x_{i,k}| \quad (1)$$

2) *Slope Sign Changes (SSC)* [32]: A feature that measures the frequency at which the sign of the signal slope changes. Given three consecutive samples  $x_{i,k-1}$ ,  $x_{i,k}$ ,  $x_{i,k+1}$ , the value of SSC is incremented by one if:

$$(x_{i,k} - x_{i,k-1}) * (x_{i,k} - x_{i,k+1}) \geq \epsilon \quad (2)$$

Where  $\epsilon \geq 0$ , is employed as a threshold to reduce the impact of noise on this feature.

3) *Zero Crossing (ZC)* [32]: A feature that counts the frequency at which the signal passes through zero. A threshold  $\epsilon \geq 0$  is utilized to lessen the impact of noise. The value of this feature is incremented by one whenever the following condition is satisfied:

$$(|x_{i,k} - x_{i,k+1}| \geq \epsilon) \wedge (\text{sgn}(x_{i,k}, x_{i,k+1}) \Leftrightarrow \text{False}) \quad (3)$$

Where  $\text{sgn}(a, b)$  return true if  $a$  and  $b$  (two real numbers) have the same sign and false otherwise. Note that depending on the slope of the signal and the selected  $\epsilon$ , the zero crossing point might not be detected.

4) *Waveform Length (WL)* [32]: A feature that offers a simple characterization of the signal's waveform. It is calculated as follows:

$$\text{WL}(x_i) = \sum_{k=1}^L |x_{i,k} - x_{i,k-1}| \quad (4)$$

5) *Skewness*: The Skewness is the third central moment of a distribution which measures the overall asymmetry of a distribution. It is calculated as follows:

$$\text{Skewness}(x_i) = \frac{1}{L} \sum_{k=1}^L \left( \frac{x_{i,k} - \bar{x}_i}{\sigma} \right)^3 \quad (5)$$

Where  $\sigma$  is the standard deviation:

6) *Root Mean Square (RMS)* [2]: This feature, also known as the quadratic mean, is closely related to the standard deviation as both are equal when the mean of the signal is zero. RMS is calculated as follows:

$$\text{RMS}(x_i) = \sqrt{\frac{1}{L} \sum_{k=1}^L x_{i,k}^2} \quad (6)$$

7) *Hjorth Parameters* [75]: Hjorth parameters are a set of three features originally developed for characterizing electroencephalography signals and then successfully applied to sEMG signal recognition [76], [34]. *Hjorth Activity Parameter* can be thought of as the surface of the power spectrum in the frequency domain and corresponds to the variance of the signal calculated as follows:

$$\text{Activity}(x_i) = \frac{1}{L} \sum_{k=1}^L (x_{i,k} - \bar{x}_i)^2 \quad (7)$$

Where  $\bar{x}_i$  is the mean of the signal for the  $i$ th window. *Hjorth Mobility Parameter* is a representation of the mean frequency of the signal and is calculated as follows:

$$\text{Mobility}(x_i) = \sqrt{\frac{\text{Activity}(x'_i)}{\text{Activity}(x_i)}} \quad (8)$$

Where  $x'_i$  is the first derivative in respect to time of the signal for the  $i$ th window. Similarly, the *Hjorth Complexity Parameter*, which represents the change in frequency, is calculated as follows:

$$\text{Complexity}(x_i) = \frac{\text{Mobility}(x'_i)}{\text{Mobility}(x_i)} \quad (9)$$

8) *Integrated EMG (IEMG)*: [2]: A feature returning the sum of the fully-rectified signal.

$$\text{IEMG}(x_i) = \sum_{k=1}^L |x_{i,k}| \quad (10)$$

9) *Autoregression Coefficient (AR)*: [3] An autoregressive model tries to predict future data, based on a weighted average of the previous data. This model characterizes each sample of the signal as a linear combination of the previous sample with an added white noise. The number of coefficients calculated is a trade-off between computational complexity and predictive power. The model is defined as follows:

$$x_{i,k} = \sum_{j=1}^P \rho_j x_{i,k-j} + \epsilon_t \quad (11)$$

Where  $P$  is the model order,  $\rho_j$  is the  $j$ th coefficient of the model and  $\epsilon_t$  is the residual white noise.

10) *Sample Entropy (SampEn)*: [77] Entropy measures the complexity and randomness of a system. Sample Entropy is a method which allows entropy estimation.

$$\text{SampEn}(x_i, m, r) = -\ln \left( \frac{A^m(r)}{B^m(r)} \right) \quad (12)$$



11) *EMG Histogram (HIST)* [78]: When a muscle is in contraction, the EMG signal deviates from its baseline. The idea behind HIST is to quantify the frequency at which this deviation occurs for different amplitude levels. HIST is calculated by determining a symmetric amplitude range centered around the baseline. This range is then separated into  $n$  bins of equal length ( $n$  is a hyperparameter). The HIST is obtained by counting how often the amplitude of the signal falls within each bin boundaries.

### B. Frequency Domain Features

1) *Cepstral Coefficient* [79], [3]: The cepstrum of a signal is the inverse Fourier transform of the log power spectrum magnitude of the signal. Like the AR, the coefficients of the cepstral coefficients are employed as features. They can be directly derived from AR as follows:

$$c_1 = -a_1 \quad (13)$$

$$c_i = -a_i - \sum_{n=1}^{i-1} \left(1 - \frac{n}{i}\right) a_n c_{i-n}, \text{ with } 1 < i \leq P \quad (14)$$

2) *Marginal Discrete Wavelet Transform (mDWT)* [80]: The mDWT is a feature that removes the time-information from the discrete wavelet transform to be insensitive to wavelet time instants. The feature instead calculates the cumulative energy of each level of the decomposition. The computation of the mDWT for each channel is implemented as follow in [39] (personal communication with the authors):

---

#### Algorithm 1 mDWT pseudo-code

---

```

1: procedure MDWT
2:    $wav \leftarrow db7$ 
3:    $level \leftarrow 3$ 
4:    $coefficients \leftarrow wavDec(x, level, wav)$ 
5:    $N \leftarrow length(coefficients)$ 
6:    $SMax \leftarrow \log_2(N)$ 
7:    $Mxk \leftarrow []$ 
8:   for  $s=1, \dots, SMax$  do
9:      $CMax \leftarrow \frac{N}{2^s} - 1$ 
10:     $val \leftarrow \sum_{u=0}^{CMax} |coefficients[u]|$ 
11:     $Mxk.append(val)$ 
return  $Mxk$ 

```

---

Where  $x$  is the 1-d signal from which to calculate the mDWT and  $wavDec$  is a function that calculates the wavelet decomposition of a vector at level  $n$  using the wavelet  $wav$ . The coefficients are returned in a 1-d vector with the *Approximation Coefficients(AC)* placed first followed by the *Detail Coefficients(DC)* (i.e.  $coefficients = [CA, CD3, CD2, CD1]$ , where 3, 2, 1 are the level of decomposition of the  $DC$ ).

Note that due to the choice of the level (3) of the wavelet decomposition in conjunction with the length of  $x$  (52) in this paper, the mDWT will be affected by boundaries effects. This choice was made to be as close as possible to the mDWT features calculated in [39] which employed the same wavelet and level on a smaller  $x$  length (40).

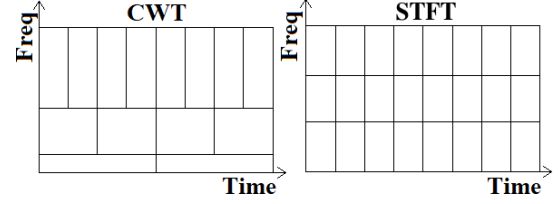


Fig. 10. A visual comparison between the CWT and the STFT. Note that due to its nature, the *frequency* of the CWT is, in fact, a pseudo-frequency.

### C. Time-Frequency Domain Features

1) *Short Term Fourier Transform based Spectrogram (Spectrogram)*: The Fourier transform allows for a frequency-based analysis of the signal as opposed to a time-based analysis. However, by its nature, this technique cannot detect if a signal is non-stationary. As sEMG are non-stationary [36], an analysis of these signals employing the Fourier transform is of limited use. An intuitive technique to address this problem is the STFT, which consists of separating the signal into smaller segments by applying a sliding window where the Fourier transform is computed for each segment. In this context, a window is a function utilized to reduce frequency leakage and delimits the segment's width (i.e. zero-valued outside of the specified segment). The spectrogram is calculated by computing the squared magnitude of the STFT of the signal. In other words, given a signal  $s(t)$  and a window of width  $w$ , the spectrogram is then:

$$spectrogram(s(t), w) = |STFT(s(t), w)|^2 \quad (15)$$

2) *Continuous Wavelet Transform (CWT)*: The Gabor limit states that a high resolution both in the frequency and time-domain cannot be achieved [81]. Thus, for the STFT, choosing a wider window yields better frequency resolution to the detriment of time resolution for all frequencies and vice versa.

Depending on the frequency, the relevance of the different signal's attributes change. Low-frequency signals have to be well resolved in the frequency band, as signals a few  $Hz$  apart can have dramatically different origins (e.g. Theta brain waves (4 to 8  $Hz$ ) and Alpha brain waves (8 to 13  $Hz$ ) [82]). On the other hand, for high-frequency signals, the relative difference between a few or hundreds  $Hz$  is often irrelevant compared to its resolution in time for the characterization of a phenomenon.

As illustrated in Fig. 10, this behavior can be obtained by employing *wavelets*. A wavelet is a signal with a limited duration, varying frequency and a mean of zero [83]. The *mother wavelet* is an arbitrarily defined wavelet that is utilized to generate different wavelets. The idea behind the wavelets transform is to analyze a signal at different scales of the mother wavelet [84]. For this, a set of wavelet functions are generated from the mother wavelet (by applying different scaling and shifting on the time-axis). The CWT is then computed by calculating the convolution between the input signal and the generated wavelets.