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, deep learning algorithms have 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 they require an unreasonable amount of effort from a single person, to generate tens of thousands of examples. This paper's hypothesis is that general, informative features can be learned from the large amounts of data generated by aggregating the signals of multiple users, thus reducing the recording burden while enhancing gesture recognition. Consequently, this paper proposes applying transfer learning on aggregated data from multiple users while leveraging the capacity of deep learning algorithms to learn discriminant features from large datasets. Two datasets comprised 19 and 17 able-bodied participants, respectively (the first one is employed for pre-training), were recorded for this work, using the Myo armband. A third Myo armband dataset was taken from the NinaPro database and is comprised ten able-bodied participants. Three different deep learning networks employing three different modalities as input (raw EMG, spectrograms, and continuous wavelet transform (CWT)) are tested on the second and third dataset. The proposed transfer learning scheme is shown to systematically and significantly enhance the performance for all three networks on the two datasets, achieving an offline accuracy of 98.31% for 7 gestures over 17 participants for the CWT-based ConvNet and 68.98% for 18 gestures over 10 participants for the raw EMG-based ConvNet. Finally, a use-case study employing eight able-bodied participants suggests that real-time feedback allows users to adapt their muscle activation strategy which reduces the degradation in accuracy normally experienced over time.**

***Index Terms*— Surface electromyography, EMG, transfer learning, domain adaptation, deep learning, convolutional networks, hand gesture recognition.**

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]–[3]. Recently, researchers have proposed deep learning approaches [4]–[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.

Manuscript received July 4, 2018; revised November 18, 2018 and December 31, 2018; accepted January 22, 2019. Date of publication January 31, 2019; date of current version April 8, 2019. This work was supported in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant 401220434, in part by the Institut de recherche Robert-Sauvé en santé et en sécurité du travail (IRSST), in part by the Fondation Famille Choquette, and in part by the Research Council of Norway through its Centres of Excellence Scheme under Project 262762. *(Corresponding author: Ulysse Côté-Allard.)*

U. Côté-Allard, C. L. Fall, and B. Gosselin are with the Department of Computer and Electrical Engineering, Université Laval, Québec City, QC G1V 0A6, Canada (e-mail: ulysse.cote-allard.1@ulaval.ca).

A. Drouin and F. Laviolette are with the Department of Computer Science and Software Engineering, Université Laval, Québec City, QC G1V 0A6, Canada.

A. Campeau-Lecours and C. Gosselin are with the Department of Mechanical Engineering, Université Laval, Québec City, QC G1V 0A6, Canada.

K. Glette is with RITMO, University of Oslo, 0315 Oslo, Norway, and also with the Department of Informatics, University of Oslo, 0315 Oslo, Norway.

This paper has supplementary downloadable material available at http://ieeexplore.ieee.org, provided by the author.

Digital Object Identifier 10.1109/TNSRE.2019.2896269

Consequently, deep learning offers a particularly attractive context from which to develop a Transfer Learning (TL) 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 TL 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 expands upon the aforementioned conference paper's work, improving the TL algorithm to reduce its computational load and improving its performance. Additionally, three new ConvNet architectures, employing three different input modalities, specifically designed for the robust and efficient classification of sEMG signals are presented. The raw signal, 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 analysis of myoelectric signals [8]). Another major 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 TL augmented version are made readily available.[1] Finally, this paper further expands the aforementioned conference paper by proposing a use-case experiment on the effect of real-time feedback on the online performance of a classifier without recalibration over a period of fourteen days. Note that, due to the stochastic nature of the algorithms presented in this paper, unless stated otherwise, all experiments are reported as an average of 20 runs.

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 Sec. II. Sec. III presents the proposed new hand gesture recognition dataset, with data acquisition and processing details alongside an overview of the NinaPro DB5 dataset. A presentation of the different state-of-the-art feature sets employed in this work is given in Sec. IV. Sec. V thoroughly describes the proposed networks' architectures, while Sec. VI presents the TL algorithm used to augment said architecture. Moreover, comparisons with the state-of-the-art in gesture recognition are given in Sec. VII. A real-time use-case experiment on the ability of users to counteract signal drift from sEMG signals is presented in Sec. VIII. Finally, results are discussed in Sec. IX.

II. Related Work

sEMG signals can vary significantly between subjects, 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] and high accuracy (85% over 6 gestures) when augmented with TL on never before seen subjects [10]. As such, 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 [11], [12]. Several works have also proposed leveraging a pre-trained model removing the need to simultaneously work with data from multiple users [13]–[15]. These non-deep learning TL approaches showed important performance gains compared to their non-augmented versions. Although, some of these gains might be due to the baseline's poorly optimized hyperparameters [16].

Short-Time Fourier Transform (STFT) have been sparsely employed in the last decades for the classification of sEMG data [17], [18]. 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 [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 [17] 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 ConvNets [4], [6].

CWT features have been employed for electrocardiogram analysis [19], electroencephalography [20] and EMG signal analysis, but mainly for lower limbs [21], [22]. Wavelet-based features have been used in the past for sEMG-based hand gesture recognition [23]. The features employed however, are based on the Discrete Wavelet Transform [24] and the Wavelet Packet Transform (WPT) [17] 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 and can now be efficiently implemented on embedded systems (see Appendix C). To the best of the authors' knowledge, this is the first time that CWT is utilized for sEMG-based hand gesture recognition.

Recently, ConvNets have started to be employed for hand gesture recognition using single array [4], [5] and matrix [25] of electrodes. Additionally, Du *et al.* [6] applied deep learning in conjunction with domain adaptation techniques but for inter-session classification as opposed to the inter-subject context of this paper. A thorough overview of deep learning techniques applied to EMG classification is given in [26]. To the best of our knowledge, this paper, which is an extension of [7], is the first time inter-user data is leveraged through TL for training deep learning algorithms on sEMG data.

III. sEMG Datasets

A. Myo Dataset

One of the major contributions of this article is to provide a new, publicly available, sEMG-based hand gesture recognition

[1] https://github.com/Giguelingueling/MyoArmbandDataset

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. 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 it will become a useful tool for the sEMG-based hand gesture classification community.

The data acquisition protocol was approved by the Comités d'Éthique de la Recherche avec des êtres humains de l'Université Laval (approbation number: 2017-026/21-02-2016) and informed consent was obtained from all participants.

1) sEMG Recording Hardware: The electromyographic activity of each subject's forearm was recorded with the Myo Armband; an 8-channel, dry-electrode, low-sampling rate (200Hz), low-cost consumer-grade sEMG armband.

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 [27]. Additionally, while the recommended frequency range of sEMG signals is 5-500Hz [28] requiring a sampling frequency greater or equal to 1000Hz, the Myo Armband is limited to 200Hz. This information loss was shown to significantly impact the ability of various classifiers to differentiate between hand gestures [29]. As such, robust and adequate classification techniques are needed to process the collected signals accurately.

2) Time-Window Length: For real-time control in a closed loop, input latency is an important factor to consider. A maximum latency of 300ms was first recommended in [30]. Even though more recent studies suggest that the latency should optimally be kept between 100-250ms [31], [32], the performance of the classifier should take priority over speed [31], [33]. As is the case in [7], a window size of 260ms was selected to achieve a reasonable number of samples between each prediction due to the low frequency of the Myo.

3) Labeled Data Acquisition Protocol: The seven hand/wrist gestures considered in this work are depicted in Fig. 1. For both sub-datasets, the labeled data was created by requiring the user to hold each gesture for five seconds. The data recording was manually started by a researcher only once the participant correctly held the requested gesture. Generally, five seconds was given to the user between each gesture. This rest period was not recorded and as a result, the final dataset is balanced for all classes. 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 subject. For the *evaluation dataset* three *round*s are available with the first *round* utilized for training (i.e. 140s per participant) and the last two for testing (i.e. 240s per participant).

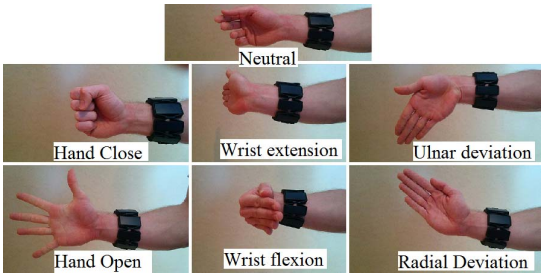

Fig. 1. The 7 hand/wrist gestures considered in the *Myo Dataset*.

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

During recording, participants were instructed to stand up and have their forearm parallel to the floor and supported by themselves. For each of them, 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). While the electrode placement was not controlled for, the orientation of the armband was always such that the blue light bar on the Myo was facing towards the hand of the subject. Note that this is the case for both left and right handed subjects. 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 is first separated by applying sliding windows of 52 samples (260ms) with an overlap of 235ms (i.e. 7×190 samples for one cycle (5s of data)). Employing windows of 260ms allows 40ms for the pre-processing and classification process, while still staying within the 300ms target [30]. Note that utilizing sliding windows is viewed as a form of data augmentation in the present context (see Appendix B). 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 Sec. IV and V.

B. NinaPro DB5

The *NinaPro DB5* is a dataset built to benchmark sEMG-based gesture recognition algorithms [34]. 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 17 gestures + neutral gesture, is of particular interest, as it includes all the gestures considered so far in this work. The 11 additional gestures which are presented in [35] 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 preceding dataset.

1) Data Acquisition and Processing: Each participant was asked to hold a gesture for five seconds followed by three seconds of neutral gesture 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 (20s per gesture) and the last two (10s per gesture) as the test set for each gesture. Note that the *rest* movement (i.e. neutral gesture) was treated identically as the other gestures (i.e. first four repetitions for training (12s) and the next two for testing (6s)).

All data processing (e.g. window size, window overlap) are exactly as described in the previous sections.

IV. Classic sEMG Classification

Traditionally, one of the most researched aspects of sEMG-based gesture recognition comes from feature engineering (i.e. manually finding a representation for sEMG signals that allows easy differentiation between gestures). Over the years, several efficient combinations of features both in the time and frequency domain have been proposed [35]–[38]. This section presents the feature sets used in this work. See Appendix D for a description of each feature.

A. Feature Sets

As this paper's main purpose is to present a deep learning-based TL approach to the problem of sEMG hand gesture recognition, contextualizing the performance of the proposed algorithms within the current state-of-the-art 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 employed for sEMG pattern recognition: Support Vector Machine (SVM) [35], Artificial Neural Networks (ANN) [39], Random Forest (RF) [35], K-Nearest Neighbors (KNN) [35] and Linear Discriminant Analysis (LDA) [38]. Hyperparameters for each classifier were selected by employing three fold cross-validation alongside random search, testing 50 different combinations of hyperparameters for each participant's dataset for each classifier. The hyperparameters considered for each classifier are presented in Appendix E.

As is often the case, dimensionality reduction is applied [1], [3], [40]. 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 [38], [41]. A comparison of the accuracy obtained with and without dimensionality reduction on the *Myo Dataset* is given in Appendix F. This comparison shows that in the vast majority of cases, the dimensionality reduction both reduced the computational load and enhanced the average performances of the feature sets.

The implementation employed for all the classifiers comes from the scikit-learn (v.1.13.1) Python package [42]. The four feature sets employed for comparison purposes are:

1) Time Domain Features (TD) [37]: This set of features, which is probably the most commonly employed in [29], often serves as the basis for bigger feature sets [1], [34], [38]. 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 [38]: 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 performances on a setup similar to the one employed in this article.

3) Nina Pro Features [34], [35]: 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 the following features: RMS, Marginal Discrete Wavelet Transform (mDWT) (wavelet = db7, S = 3), EMG Histogram (HIST) (bins = 20, threshold = 3σ) and the TD features.

4) SampEn Pipeline [36]: 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 = 0.2σ), Cepstral Coefficient (order = 4), RMS and WL.

V. Deep Learning Classifiers Overview

ConvNets 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 (see Appendix C). As previously mentioned, the inherent limitations of sEMG-based gesture recognition force the proposed ConvNets to contend with a limited amount of data from any single individual. To address the over-fitting issue, Monte Carlo Dropout (MC Dropout) [43], Batch Normalization (BN) [44], and early stopping are employed.

A. Batch Normalization

BN is a technique that accelerates training and provides some form of regularization with the aims of maintaining a standard distribution of hidden layer activation values throughout training [44]. 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 network.

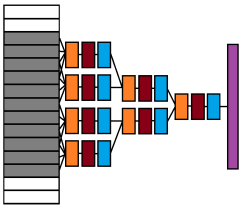

Fig. 3. Typical slow-fusion ConvNet architecture [46]. In this graph, the input (represented by gray rectangles) is a video (i.e. a sequence of images). The model separates the temporal part of the examples into disconnected parallel layers, which are then slowly fused together throughout the network.

Once training is completed, the whole dataset is fed through the network one last time to compute the final normalization parameters in a layer-wise 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.

B. Proposed Convolutional Network Architectures

Videos are a representation of how spatial information (images) change through time. Previous works have combined this representation with ConvNets to address classification tasks [45], [46]. One such successful algorithm is the slow-fusion model [46] (see Fig. 3).

When calculating the spectrogram of a signal, the information is structured in a Time $\times$ Frequency fashion (Time $\times$ Scale for CWT). When the signal comes from an array of electrodes, these examples can naturally be structured as Time $\times$ Spatial $\times$ Frequency (Time $\times$ Spatial $\times$ Scale for CWT). As such, the motivation for using a slow-fusion architecture based ConvNet 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 $\times$ Spatial $\times$ 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) 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 4×15. 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 $4 \times 8 \times 14$ (Time $\times$ Channel $\times$ Frequency).

The implementation of the spectrogram ConvNet architecture (Appendix A, Fig. 1) was created with Theano [47] and Lasagne [48]. As usual in deep learning, the architecture was created in a trial and error process taking inspiration from previous architectures (primarily [4], [6], [7], [46]). The non-linear activation functions employed are the parametric exponential linear unit (PELU) [49] and PReLU [50]. ADAM [51] is utilized for the optimization of the ConvNet (learning rate $=$ 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 randomly removing 10% of the data from the training and using it as a validation set at the beginning of the optimization process. 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 occurs with no network performance amelioration on the validation set. All hyperparameter values were found by a random search on the *pre-training dataset*.

2) ConvNet for Continuous Wavelet Transforms: The architecture for the CWT ConvNet, (Appendix A, Fig. 2), 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 [52], [53]. In the end, the Mexican Hat wavelet was selected, as it was the best performing during cross-validation on the *pre-training dataset*. The CWTs were calculated with 32 scales yielding a 32×52 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, similarly to the spectrogram, the last row of the calculated CWT was removed as to reduce baseline drift and motion artifact. Additionally, the last column of the calculated CWT was also removed as to provide an even number of time-columns from which to perform the slow-fusion process. The final matrix shape is thus $12 \times 8 \times 7$ (i.e. Time $\times$ Channel $\times$ Scale). The MC Dropout deactivation rate, batch size, optimization algorithm, and activation functions remained unchanged. The learning rate was set at 0.0879923 (found by cross-validation).

3) ConvNet for Raw EMG: A third ConvNet architecture taking the raw EMG signal as input is also considered. This network will help assess if employing time-frequency features lead to sufficient gains in accuracy performance to justify the increase in computational cost. As the raw EMG represents a completely different modality, a new type of architecture must be employed. To reduce bias from the authors as much as possible, the architecture considered is the one presented in [54]. The *raw ConvNet* architecture can be seen in Appendix A, Fig. 3. This architecture was selected as it was also designed to classify a hand gesture dataset employing the Myo Armband. The architecture implementation (in PyTorch v.0.4.1) is exactly as described in [54] except for the learning rate ($= 1.1288378916846883e-5$) which was found by cross-validation (tested 20 uniformly distributed values between $1e-6$ to $1e-1$ on a logarithm scale) and extending the length of the window size as to match with the rest of this manuscript.

The *raw ConvNet* is further enhanced by introducing a second convolutional and pooling layer as well as adding dropout, BN, replacing RELU activation function with PReLU and using ADAM (learning rate $=$ 0.002335721469090121) as the optimizer. The *enhanced raw ConvNet*'s architecture, which is shown in Appendix A, Fig. 4, achieves an average accuracy of 97.88% compared to 94.85% for the *raw ConvNet*. Consequently, all experiments using raw emg as input will employ the *raw enhanced ConvNet*.

VI. Transfer Learning

One of the main advantages of deep learning comes from its ability to leverage large amounts of data for learning. As it would be too time-consuming for a single individual to record tens of thousands of examples, this work proposes to aggregate the data of multiple individuals. 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. TL 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 data recording 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 TL, automatic *alignment* is a necessary first step. The alignment for each subject was made by identifying the most active channel (calculated using the IEMG feature) 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 TL technique in deep learning [55], [56]. 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* [57], 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) [57] 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 (See [57] for details).

B. Adaptive Batch Normalization

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 TL 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. As such, when training the source network, the data from all subjects are aggregated and fed to the network together. It is important to note that each training batch is comprised solely of examples that belong to a single participant. This allows the update of the participant's corresponding BN statistic.

C. Proposed Transfer Learning Architecture

The main tenet behind TL is that similar tasks can be completed in similar ways. The difficulty in this paper's context is then to learn a mapping 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 training a *Source Network* (presented in Sec. V) shared across all participants of the *pre-training dataset* with the multi-stream AdaBatch and adding only a second network for the target task using the PNN architecture, the scaling problem in the current context vanishes. 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, whereas the *Second Network* only employed PELU. This architecture choice was made through trial and error and cross-validation on the *pre-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. 4). The idea behind the merging of information through element-wise summation is two-fold. First, compared to concatenating the features maps (as in [7]) or employing non-linear lateral connections (like in [57]), element-wise summation minimizes the computational impact of connecting the *Source Network* and the *Second Network* together. Second, this provides a mechanism that fosters residual learning as inspired by Residual Networks [58]. 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

TABLE I
CLASSIFICATION ACCURACY OF THE CONVNETS ON THE *Evaluation Dataset* WITH RESPECT TO THE NUMBER OF TRAINING CYCLES PERFORMED

	Raw	Raw + TL	Spectrogram	Spectrogram + TL	CWT	CWT + TL
4 Cycles	97.08%	**97.39%**	97.14%	**97.85%**	97.95%	**98.31%**
STD	4.94%	**4.07%**	2.85%	**2.45%**	2.49%	**2.16%**
H0 (p-value)	0 (0.02187)	-	0 (0.00030)	-	0 (0.00647)	-
3 Cycles	96.22%	**96.95%**	96.33%	**97.40%**	97.22%	**97.82%**
STD	6.49%	**4.88%**	3.49%	**2.91%**	3.46%	**2.41%**
H0 (p-value)	0 (0.00155)	-	0 (0.00018)	-	0 (0.00113)	-
2 Cycles	94.53%	**95.49%**	94.19%	**96.05%**	95.17%	**96.63%**
STD	9.63%	**7.26%**	5.95%	**6.00%**	5.77%	**4.54%**
H0 (p-value)	0 (0.00430)	-	0 (0.00015)	-	0 (0.00030)	-
1 Cycle	89.04%	**92.46%**	88.51%	**93.93%**	89.02%	**94.69%**
STD	10.63%	**7.79%**	8.37%	**6.56%**	10.24%	**5.58%**
H0 (p-value)	0 (0.00018)	-	0 (0.00015)	-	0 (0.00015)	-

* The *one-tail Wilcoxon signed rank test* is applied to compare the ConvNet enhanced with the proposed TL algorithm to their non-augmented counterpart. Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$).
**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

from the *Source Network* actually hinders learning. 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.

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. 4. 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.

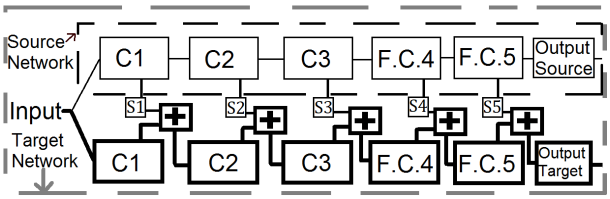

Fig. 4. The PNN-inspired architecture. This figure represents the case with the spectrogram ConvNet. Note that the TL behavior is the same for the Raw-based or CWT-based ConvNet. C1,2,3 and F.C.4,5 correspond to the three stages of convolutions and two stages of fully connected layers respectively. The *Si* (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 for both the spectrogram and CWT-based ConvNet). The + boxes represent the merging through an element-wise summation of the ConvNets' corresponding layers.

VII. CLASSIFIER COMPARISON

A. Myo Dataset

All pre-trainings in this section were done on the *pre-training dataset* and all training (including for the traditional machine learning algorithms) were done on the first *round* of the *evaluation dataset*.

1) Comparison With Transfer Learning: Considering each participant as a separate dataset allows for the application of the one-tail Wilcoxon signed-rank test [59] ($n = 17$). Table I shows a comparison of each ConvNet with their TL augmented version. Accuracies are given for one, two, three and four cycles of training.

2) Comparison With State of the Art: A comparison between the proposed CWT-based ConvNet and a variety of classifiers trained on the features sets presented in Sec. IV-A is given in Table II.

As suggested in [60], a two-step procedure is employed to compare the deep learning algorithms with the current state-of-the-art. First, Friedman's test ranks the algorithms amongst each other. Then, Holm's post-hoc test is applied ($n = 17$) using the best ranked method as a comparison basis.

B. NinaPro Dataset

1) Comparison With Transfer Learning: Performance of the proposed ConvNet architecture alongside their TL augmented versions are investigated on the *NinaPro DB5*. As no specific pre-training dataset is available for the *NinaPro DB5*, the pre-training for each participant is done employing the training sets of the remaining nine participants. Table III shows the average accuracy over the 10 participants of the *NinaPro DB5* for one to four cycles. Similarly to Sec. VII-A.1, the one-tail Wilcoxon Signed rank test is performed for each cycle between each ConvNet and their TL augmented version.

2) Comparison With State of the Art: Similarly to Sec. VII-A.2, a comparison between the TL-augmented ConvNet and the traditional classifier trained on the state-of-the-art feature set is given in Table IV. The accuracies are given

TABLE II
CLASSIFIERS COMPARISON ON THE *Evaluation Dataset* WITH RESPECT TO THE NUMBER OF TRAINING CYCLES PERFORMED

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	CWT	CWT + TL
4 Cycles	97.61% (LDA)	98.14% (LDA)	97.59% (LDA)	97.72% (LDA)	97.95%	**98.31%**
STD	2.63%	2.21%	2.74%	1.98%	2.49%	**2.16%**
Friedman Rank	3.94	2.71	4.29	3.47	3.94	**2.65**
H0	1	1	1	1	1	-
3 Cycles	96.33% (KNN)	97.33% (LDA)	96.76% (KNN)	96.87% (KNN)	97.22%	**97.82%**
STD	6.11%	3.24%	3.85%	5.06%	3.46%	**2.41%**
Friedman Rank	4.41	2.77	4.05	3.53	3.94	**2.29**
H0	0 (0.00483)	1	0 (0.02383)	1	0 (0.03080)	-
2 Cycles	94.12% (KNN)	94.79% (LDA)	94.23% (KNN)	94.68% (KNN)	95.17%	**96.63%**
STD	9.08%	7.82%	7.49%	8.31%	5.77%	**4.54%**
Friedman Rank	4.41	3.24	4.41	3.29	3.65	**2.00**
H0 (adjusted p-value)	0 (0.00085)	1	0 (0.00085)	1	0 (0.03080)	-
1 Cycle	90,77% (KNN)	91.25% (LDA)	90.21% (LDA)	91.66% (KNN)	89.02%	**94.69%**
STD	9.04%	9.44%	7.73%	8.74%	10.24%	**5.58%**
Friedman Rank	3.71	3.41	4.41	3.05	4.88	**1.53**
H0 (adjusted p-value)	0 (0.00208)	0 (0.00670)	0 (0.00003)	0 (0.01715)	0 (<0.00001)	-

*For brevity's sake, only the best performing classifier for each feature set in each cycle is reported (indicated in parenthesis).
**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.
***The Friedman Ranking Test followed by the Holm's post-hoc test is performed.

TABLE III
CLASSIFICATION ACCURACY OF THE CONVNETS ON THE *NinaPro DB5* WITH RESPECT TO THE NUMBER OF TRAINING CYCLES PERFORMED

	Raw	Raw + TL	Spectrogram	Spectrogram + TL	CWT	CWT + TL
4 Repetitions	66.32%	**68.98%**	63.60%	**65.10%**	61.89%	**65.57%**
STD	3.94%	**4.46%**	3.94%	**3.99%**	4.12%	**3.68%**
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
3 Repetitions	61.91%	**65.16%**	60.09%	**61.70%**	58.37%	**62.21%**
STD	3.94%	**4.46%**	4.03%	**4.29%**	4.19%	**3.93%**
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
2 Repetitions	55.67%	**60.12%**	55.35%	**57.19%**	53.32%	**57.53%**
STD	4.38%	**4.79%**	4.50%	**4.71%**	3.72%	**3.69%**
H0 (p-value)	0 (0.00253)	-	0 (0.00253)	-	0 (0.00253)	-
1 Repetitions	46.06%	**49.41%**	45.59%	**47.39%**	42.47%	**48.33%**
STD	6.09%	**5.82%**	5.58%	**5.30%**	7.04%	**5.07%**
H0 (p-value)	0 (0.00467)	-	0 (0.00467)	-	0 (0.00253)	-

* The *Wilcoxon signed rank test* is applied to compare the ConvNet enhanced with the proposed TL algorithm to their non-augmented counterpart. Null hypothesis is rejected when $H_0 = 0$ ($p < 0.05$).
**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.

for one, two, three and four cycles of training. A two-step statistical test with the Friedman test as the first step and Holm post-hoc as the second step is again employed.

3) Out-of-Sample Gestures: A final test involving the *NinaPro DB5* was conducted to evaluate the impact on the proposed TL 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 training and test set of the *NinaPro DB5* as described before, but considering only the gestures that were absent from the *pre-training dataset* (11 total). The CWT ConvNet was then compared to its TL augmented version which was pre-trained on the *pre-training dataset*. Fig. 5 presents the accuracies obtained for the classifiers with different number of repetitions employed for training. The difference in accuracy is considered statistically significant by the one-tail Wilcoxon Signed rank test for all cycles of training. Note that, similar, statistically significant results were obtained for the raw-based and spectrogram-based ConvNets.

VIII. REAL-TIME CLASSIFICATION AND MEDIUM TERM PERFORMANCES (CASE STUDY)

This last experiment section proposes a use-case study of the online (i.e. real-time) 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 occur, the performance of a classifier degrades over time due to the non-stationary property of sEMG signals [61]. The main goal of this use-case experiment is to evaluate if users are able to self-adapt and improve the way they perform gestures based on visual feedback from complex classifiers (e.g. *CWT+TL*), thus reducing the expected classification degradation.

To achieve this, each participant recorded a training set as described in Sec. III. Then, over the next fourteen days, a daily

TABLE IV
CLASSIFIERS COMPARISON ON THE *NinaPro DB5* WITH RESPECT TO THE NUMBER OF REPETITIONS USED DURING TRAINING.

	TD	Enhanced TD	Nina Pro	SampEn Pipeline	Raw	Raw + TL
4 Repetitions	59.91% (RF)	59.57% (RF)	56.72% (RF)	62.30% (RF)	66.32%	**68.98%**
STD	3.50%	4.43%	4.01%	3.94%	3.77%	**4.09%**
Friedman Rank	4.30	4.60	6.00	3.00	2.10	**1.00**
H0 (Adjusted p-value)	0 (0.00024)	0 (0.00007)	0 (<0.00001)	0 (0.03365)	1	-
3 Repetitions	55.73% (RF)	55.32% (RF)	52.33% (RF)	58.24% (RF)	61.91%	**65.16%**
STD	3.75%	4.48%	4.63%	4.22%	3.94%	**4.46%**
Friedman Rank	4.40	4.60	6.00	3.00	2.00	**1.00**
H0 (Adjusted p-value)	0 (0.00014)	0 (0.00007)	0 (<0.00001)	0 (0.03365)	1	-
2 Repetitions	50.85% (RF)	50.08% (LDA)	46.85% (LDA)	53.00% (RF)	55.65%	**60.12%**
STD	4.29%	4.63%	4.81%	3.85%	4.38%	**4.79%**
Friedman Rank	4.20	4.60	6.00	3.10	2.10	**1.00**
H0 (Adjusted p-value)	0 (0.00039)	0 (0.00007)	0 (<0.00001)	0 (0.02415)	1	-
1 Repetitions	40.70% (RF)	40.86% (LDA)	37.60% (LDA)	42.26% (LDA)	46.06%	**49.41%**
STD	5.84%	6.91%	6.67%	5.78%	6.09%	**5.82%**
Friedman Rank	4.30	4.30	5.80	3.50	2.00	**1.10**
H0 (Adjusted p-value)	0 (0.00052)	0 (0.00052)	0 (<0.00001)	0 (0.00825)	1	-

*For brevity's sake, only the best performing classifier for each feature set is reported (indicated in parenthesis).
**The STD represents the pooled standard variation in accuracy for the 20 runs over the 17 participants.
***The Friedman Ranking Test followed by the Holm's post-hoc test is performed.

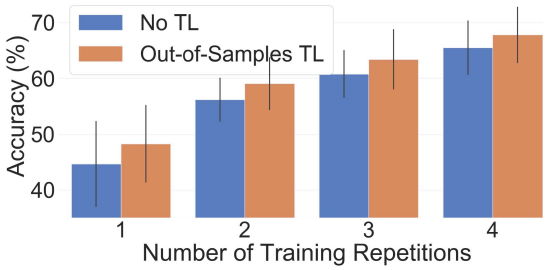

Fig. 5. Classification accuracy of the CWT-based ConvNets on the *NinaPro DB5* with respect to the number of repetitions employed during training. The pre-training was done using the *pre-training dataset*. Training and testing only considered the 11 gestures from the *NinaPro DB5* not included in the pre-training. The error bars correspond to the STD across all ten participants.

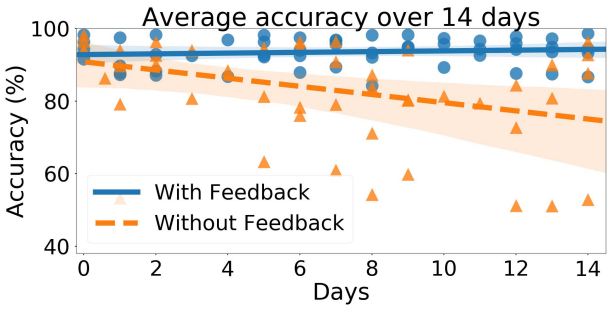

Fig. 6. Average accuracy over 14 days without recalibration of the CWT+TL ConvNet The blue circles represent data from the *Feedback* group whereas the orange triangles represent data from the *Without Feedback* group. The translucent bands around the linear regressions represent the confidence interval (95%) estimated by bootstrap.

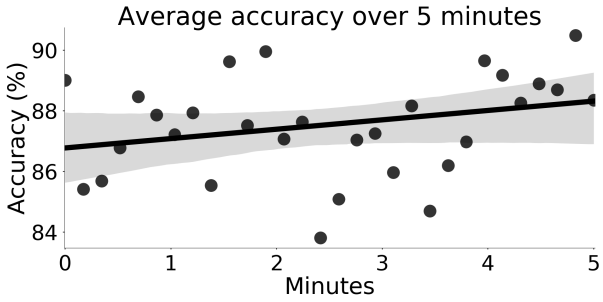

Fig. 7. The average accuracy of the eight participants over all the five minute sessions recorded to evaluate the effect of muscle fatigue on the classifier performance. During each session of the experiment, participants were asked to hold a total of 30 random gestures for ten seconds each. As such, 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.

session was recorded based on the participant's availability. A *session* consisted of holding a set of 30 randomly selected gestures (among the seven shown in Fig. 1) for ten seconds each, resulting in five minutes of continuous sEMG data. Note that to be more realistic, the participants began by placing the armband themselves, leading to slight armband position variations between sessions.

The eight participants were randomly separated into two equal groups. The first group, referred to as the *Feedback* group, received real-time feedback on the gesture predicted by the classifier in the form of text displayed on a computer screen. The second group, referred to as the *Without Feedback* group, did not receive classifier feedback. The classifier employed in this experiment is the *CWT+TL*, as it was the best performing classifier tested on the *Evaluation Dataset*. Because the transitions are computer-specified, there is a latency between a new requested gesture and the participant's reaction. To reduce the impact of this phenomenon, the data from the first second after a new requested gesture is ignored from this section results. 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.

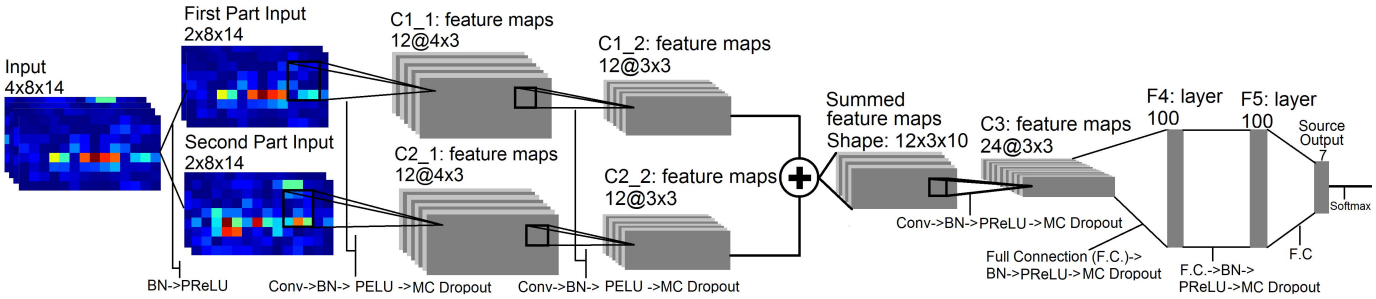

Fig. 8. The proposed spectrogram ConvNet architecture to leverage spectrogram examples employing 67 179 learnable parameters. To allow the slow fusion process, the input is first separated equally into two parts with respect to the time axis. The two branches are then fused together by element-wise summing the feature maps together. In this figure, *Conv* refer to *Convolution* and *F.C.* to *Fully Connected* layers.

As it can be observed in Fig. 6, 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 average accuracy across all participants for the first recording session was 95.42%.

Many participants reported experiencing muscular fatigue 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 can be observed from the positive slope of the linear regression presented in Fig. 7, muscle fatigue, does not seem to negatively affect the proposed ConvNet's accuracy.

IX. Discussion

Table I and Table III show that, in all cases the TL augmented ConvNets significantly outperformed their non-augmented versions, regardless of the number of training cycles. As expected, reducing the amount of training cycles systematically degraded the performances of all tested methods (see Table I, II, III, IV and Fig. 5), with the non-TL ConvNets being the most affected on the *Myo Dataset*. This is likely due to overfitting that stems from the small size of the dataset. However, it is worth noting that, when using a single cycle of training, augmenting the ConvNets with the proposed TL scheme significantly improves their accuracies. In fact, with this addition, the accuracies of the ConvNets become the highest of all methods on both tested datasets. Overall, the proposed TL-augmented ConvNets were competitive with the current state-of-the-art, with the *TL augmented CWT-based ConvNet* achieving a higher average accuracy than the traditional sEMG classification technique on both datasets for all training cycles. It is also noteworthy that while the *raw+TL* ConvNet was the worst amongst the TL augmented ConvNet on the *Myo Dataset*, it achieved the highest accuracy on the *NinaPro DB5*. Furthermore, the TL method outperformed the non-augmented ConvNets on the out-of-sample experiment. The difference in accuracy of the two methods was deemed significant by the Wilcoxon Signed Rank Test ($p < 0.05$) for all training repetitions. This suggests that the proposed TL 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.

While in this paper, the proposed source and *second network* were almost identical they are performing different tasks (see Sec. VI-C). As such further differentiation of both networks might lead to increased performance. At first glance, the element-wise summation between the *source* and *second network* might seem to impose a strong constraint on the architecture of the two networks. However, one could replace the learned scalar layers in the *target network* by convolutions or fully connected layers to bridge the dimensionality gap between potentially vastly different *source* and *second* networks.

Additionally, a difference in the average accuracy between the real-time experiment (Sec. VIII) and the *Evaluation Dataset* (Sec. VII-A.2) was observed (95.42% vs 98.31% respectively). This is likely due to the reaction delay of the participants, but more importantly to the transition between gestures. These transitions are not part of the training dataset, because they are too time consuming to record as the number of possible transitions equals $n^2 - n$ where n is the number of gestures. Consequently, it is expected that the classifiers predictive power on transition data is poor in these circumstances. As such, being able to accurately detect such transitions in an unsupervised way might have a greater impact on the system's responsiveness than simply reducing the window size. This and the aforementioned point will be investigated in future works.

The main limitation of this study is the absence of tests with amputees. Additionally, the issue of electrode shifts has not been explicitly studied and the variability introduced by various limb positions was not considered when recording the dataset. A limitation of the proposed TL scheme is its difficulty to adapt when the new user cannot wear the same amount of electrodes as the group used for pre-training. This is because changing the number of channels changes the representation of the phenomena (i.e. muscle contraction) being fed to the algorithm. The most straightforward way of addressing this would be to numerically remove the relevant channels from the dataset used for pre-training. Then re-running the proposed TL algorithm on an architecture adapted to the new representation fed as input. Another solution is to consider the EMG channels

in a similar way as color channels in image. This type of architecture seems, however, to perform worse than the ones presented in this paper (see Appendix G).

X. Conclusion

This paper presents three novel ConvNet architectures that were shown to be competitive with current sEMG-based classifiers. Moreover, this work presents a new TL scheme that systematically and significantly enhances the performances of the tested ConvNets. On the newly proposed *evaluation* dataset, the TL augmented ConvNet achieves an average accuracy of 98.31% over 17 participants. Furthermore, on the *NinaPro DB5* dataset (18 hand/wrist gestures), the proposed classifier achieved an average accuracy of 68.98% over 10 participants on a single Myo Armband. This dataset showed that the proposed TL algorithm learns sufficiently general features to significantly enhance the performance of ConvNets on out-of-sample gestures. 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.

Future works will focus on adapting and testing the proposed TL algorithm on upper-extremity amputees. This will provide additional challenges due to the greater muscle variability across amputees and the decrease in classification accuracy compared to able-bodied participants [35]. Additionally, tests for the application of the proposed TL algorithm for inter-session classification will be conducted as to be able to leverage labeled information for long-term classification.

Acknowledgements

F. Laviolette and B. Gosselin share senior authorship.

References

[1] M. A. Oskoei and H. Hu, "Myoelectric control systems—A survey," *Biomed. Signal Process. 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 *Proc. Int. Conf. Elect. Eng./Electron. Comput. Telecommun. Inf. Technol. (ECTI-CON)*, May 2010, pp. 856–860.

[3] A. Phinyomark, P. Phukpattaranont, and C. Limsakul, "Feature reduction and selection for EMG signal classification," *Expert Syst. Appl.*, vol. 39, pp. 7420–7431, Jun. 2012.

[4] U. C. Allard *et al.*, "A convolutional neural network for robotic arm guidance using semg based frequency-features," in *Proc. Intell. Robots Syst. (IROS)*, Oct. 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 Neurorobot.*, vol. 10, p. 9, Sep. 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 *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2017, pp. 1663–1668.

[8] S. Karlsson, J. Yu, and M. Akay, "Time-frequency analysis of myoelectric signals during dynamic contractions: A comparative study," *IEEE Trans. Biomed. Eng.*, vol. 47, no. 2, pp. 228–238, Feb. 2000.

[9] C. Castellini, A. E. Fiorilla, and G. Sandini, "Multi-subject/daily-life activity EMG-based control of mechanical hands," *J. Neuroeng. Rehabil.*, vol. 6, no. 1, p. 41, 2009.

[10] A.-A. Samadani and D. Kulic, "Hand gesture recognition based on surface electromyography," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2014, pp. 4196–4199.

[11] R. N. Khushaba, "Correlation analysis of electromyogram signals for multiuser myoelectric interfaces," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 22, no. 4, pp. 745–755, Jul. 2014.

[12] R. Chattopadhyay, N. C. Krishnan, and S. Panchanathan, "Topology preserving domain adaptation for addressing subject based variability in SEMG signal," in *Proc. AAAI Spring Symp., Comput. Physiol.*, 2011, pp. 4–9.

[13] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo, "Improving control of dexterous hand prostheses using adaptive learning," *IEEE Trans. Robot.*, vol. 29, no. 1, pp. 207–219, Feb. 2012.

[14] N. Patricia, T. Tommasit, and B. Caputo, "Multi-source adaptive learning for fast control of prosthetics hand," in *Proc. 22nd Int. Conf. Pattern Recognit. (ICPR)*, Aug. 2014, pp. 2769–2774.

[15] F. Orabona, C. Castellini, B. Caputo, A. E. Fiorilla, and G. Sandini, "Model adaptation with least-squares SVM for adaptive hand prosthetics," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2009, pp. 2897–2903.

[16] V. Gregori, A. Gijsberts, and B. Caputo, "Adaptive learning to speed-up control of prosthetic hands: A few things everybody should know," in *Proc. Int. Conf. Rehabil. Robot. (ICORR)*, Feb. 2017, pp. 1130–1135.

[17] K. Englehart, B. Hudgins, P. A. Parker, and M. Stevenson, "Classification of the myoelectric signal using time-frequency based representations," *Med. Eng. Phys.*, vol. 21, nos. 6–7, pp. 431–438, 1999.

[18] G. Tsenov, A. Zeghbib, F. Palis, N. Shoylev, and V. Mladenov, "Neural networks for online classification of hand and finger movements using surface EMG signals," in *Proc. 8th Seminar Neural Netw. Appl. Elect. Eng. (NEUREL)*, Sep. 2006, pp. 167–171.

[19] P. S. Addison, "Wavelet transforms and the ECG: A review," *Physiol. Meas.*, vol. 26, no. 5, p. R155, 2005.

[20] O. Faust, U. R. Acharya, H. Adeli, and A. Adeli, "Wavelet-based EEG processing for computer-aided seizure detection and epilepsy diagnosis," *Seizure-Eur. J. Epilepsy*, vol. 26, pp. 56–64, Mar. 2015.

[21] S. Karlsson and B. Gerdle, "Mean frequency and signal amplitude of the surface EMG of the quadriceps muscles increase with increasing torque—A study using the continuous wavelet transform," *J. Electromyogr. Kinesiol.*, vol. 11, no. 2, pp. 131–140, 2001.

[22] A. R. Ismail and S. S. Asfour, "Continuous wavelet transform application to EMG signals during human gait," in *Proc. Conf. Rec. 32nd Asilomar Conf. Signals, Syst. Comput.*, vol. 1, Nov. 1998, pp. 325–329.

[23] K. Englehart, B. Hudgins, and P. A. Parker, "A wavelet-based continuous classification scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 48, no. 3, pp. 302–311, Mar. 2001.

[24] C. Toledo, R. Muñoz, and L. Leija, "SEMG signal detector using discrete wavelet transform," in *Proc. Pan Amer. Health Care Exchanges (PAHCE)*, Mar. 2012, pp. 62–65.

[25] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Sci. Rep.*, vol. 6, Nov. 2016, Art. no. 36571.

[26] A. Phinyomark and E. Scheme, "EMG pattern recognition in the era of big data and deep learning," *Big Data Cogn. Comput.*, vol. 2, no. 3, p. 21, 2018.

[27] D. F. Stegeman, B. U. Kleine, B. G. Lapatki, and J. P. Van Dijk, "High-density surface EMG: Techniques and applications at a motor unit level," *Biocybern. Biomed. Eng.*, vol. 32, no. 3, pp. 3–27, 2012.

[28] R. Merletti and P. Di Torino, "Standards for reporting EMG data," *J. Electromyogr. Kinesiol.*, vol. 9, no. 1, pp. 3–4, 1999.

[29] 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.

[30] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, Jan. 1993.

[31] T. R. Farrell and R. F. Weir, "The optimal controller delay for myoelectric prostheses," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 15, no. 1, pp. 111–118, Mar. 2007.

[32] L. H. Smith, L. J. Hargrove, B. A. Lock, and T. A. Kuiken, "Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 19, no. 2, pp. 186–192, Apr. 2011.

[33] B. Peerdeman *et al.*, "Myoelectric forearm prostheses: State of the art from a user-centered perspective," *J. Rehabil. Res. Develop.*, vol. 48, no. 6, pp. 719–738, 2011.

[34] 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, 2017, Art. no. e0186132.

[35] M. Atzori *et al.*, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Sci. Data*, vol. 1, p. 140053, Dec. 2014.

[36] A. Phinyomark, F. Quaine, S. Charbonnier, C. Serviere, F. Tarpin-Bernard, and Y. Laurillau, "EMG feature evaluation for improving myoelectric pattern recognition robustness," *Expert Syst. Appl.*, vol. 40, no. 12, pp. 4832–4840, 2013.

[37] K. Englehart and B. Hudgins, "A robust, real-time control scheme for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 50, no. 7, pp. 848–854, Jul. 2003.

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

[39] M. Ahsan, M. I. Ibrahimy, and O. O. Khalifa, "EMG signal classification for human computer interaction: A review," *Eur. J. Sci. Res.*, vol. 33, no. 3, pp. 480–501, 2009.

[40] R. N. Khushaba, S. Kodagoda, M. Takruri, and G. Dissanayake, "Toward improved control of prosthetic fingers using surface electromyogram (EMG) signals," *Expert Syst. Appl.*, vol. 39, no. 12, pp. 10731–10738, 2012.

[41] 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 *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 4850–4855.

[42] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825–2830, Oct. 2011.

[43] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[44] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. Mach. Learn.*, 2015, pp. 448–456.

[45] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Proc. Int. Workshop Hum. Behav. Understand.* Berlin, Germany: Springer, 2011, pp. 29–39.

[46] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1725–1732.

[47] R. Al-Rfou *et al.* (2016). "Theano: A Python framework for fast computation of mathematical expressions." [Online]. Available: https://arxiv.org/abs/1605.02688

[48] S. Dieleman *et al.* (Aug. 2015). *Lasagne: First Release*. [Online]. Available: http://dx.doi.org/10.5281/zenodo.27878

[49] L. Trottier, P. Giguère, and B. Chaib-Draa. (2016). "Parametric exponential linear unit for deep convolutional neural networks." [Online]. Available: https://arxiv.org/abs/1605.09332

[50] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2015, pp. 1026–1034.

[51] D. Kingma and J. Ba. (2014). "Adam: A method for stochastic optimization." [Online]. Available: https://arxiv.org/abs/1412.6980

[52] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: Detection, processing, classification and applications," *Biol. Procedures*, vol. 8, no. 1, pp. 11–35, Dec. 2006.

[53] R. Reynolds and M. Lakie, "Postmovement changes in the frequency and amplitude of physiological tremor despite unchanged neural output," *J. Neurophysiol.*, vol. 104, no. 4, pp. 2020–2023, 2010.

[54] M. Zia ur Rehman *et al.*, "Multiday EMG-based classification of hand motions with deep learning techniques," *Sensors*, vol. 18, no. 8, p. 2497, 2018.

[55] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proc. Int. Conf. Mach. Learn. Workshop Unsupervised Transf. Learn.*, vol. 27, 2012, pp. 17–36.

[56] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Proc. Adv. Neural Inf. Process. Syst.*, 2014, pp. 3320–3328.

[57] A. A. Rusu *et al.* (2016). "Progressive neural networks." [Online]. Available: https://arxiv.org/abs/1606.04671

[58] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 770–778.

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

[60] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1–30, Jan. 2006.

[61] J. Liu, X. Sheng, D. Zhang, J. He, and X. Zhu, "Reduced daily recalibration of myoelectric prosthesis classifiers based on domain adaptation," *IEEE J. Biomed. Health Informat.*, vol. 20, no. 1, pp. 166–176, Jan. 2016.

[62] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.

[63] S. Dieleman *et al.* (2015). *Classifying Plankton With Deep Neural Networks*. Online]. Available: http://benanne.github.io/2015/03/17/plankton.html

[64] 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.

[65] Y. Chen *et al.*, "Neuromorphic computing's yesterday, today, and tomorrow—An evolutional view," *Integr., VLSI J.*, vol. 61, pp. 49–61, Mar. 2018.

[66] E. Nurvitadhi *et al.*, "Can FPGAs beat GPUs in accelerating next-generation deep neural networks?" in *Proc. ACM/SIGDA Int. Symp. Field-Program. Gate Arrays*, 2017, pp. 5–14.

[67] L. Cavigelli, M. Magno, and L. Benini, "Accelerating real-time embedded scene labeling with convolutional networks," in *Proc. 52nd ACM/EDAC/IEEE Design Autom. Conf. (DAC)*, Jun. 2015, pp. 1–6.

[68] Y. H. Chen, T. Krishna, J. S. Emer, and V. Sze, "Eyeriss: An energy-efficient reconfigurable accelerator for deep convolutional neural networks," *IEEE J. Solid-State Circuits*, vol. 52, no. 1, pp. 127–138, Jan. 2017.

[69] S. Han *et al.*, "EIE: Efficient inference engine on compressed deep neural network," in *Proc. 43rd Int. Symp. Comput. Archit.*, Jun. 2016, pp. 243–254.

[70] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized convolutional neural networks for mobile devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4820–4828.

[71] Y. T. Qassim, T. R. Cutmore, and D. D. Rowlands, "Optimized FPGA based continuous wavelet transform," *Comput. Elect. Eng.*, vol. 49, pp. 84–94, Jan. 2016.

[72] N. Žarić, S. Stanković, and Z. Uskoković, "Hardware realization of the robust time–frequency distributions," *Ann. Telecommun.*, vol. 69, nos. 5–6, pp. 309–320, 2014.

[73] B. Hjorth, "EEG analysis based on time domain properties," *Electroencephalogr. Clin. Neurophysiol.*, vol. 29, no. 3, pp. 306–310, 1970.

[74] M. Mouzé-Amady and F. Horwat, "Evaluation of hjorth parameters in forearm surface emg analysis during an occupational repetitive task," *Electroencephalogr. Clin. Neurophysiol.*, vol. 101, no. 2, pp. 181–183, 1996.

[75] X. Zhang and P. Zhou, "Sample entropy analysis of surface EMG for improved muscle activity onset detection against spurious background spikes," *J. Electromyogr. Kinesiol.*, vol. 22, no. 6, pp. 901–907, Dec. 2012.

[76] M. Zardoshti-Kermani, B. C. Wheeler, K. Badie, and R. M. Hashemi, "EMG feature evaluation for movement control of upper extremity prostheses," *IEEE Trans. Rehabil. Eng.*, vol. 3, no. 4, pp. 324–333, Dec. 1995.

[77] 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 Trans. Biomed. Eng.*, vol. 42, no. 8, pp. 777–785, Aug. 1995.

[78] M.-F. Lucas, A. Gaufriau, S. Pascual, C. Doncarli, and D. Farina, "Multi-channel surface EMG classification using support vector machines and signal-based wavelet optimization," *Biomed. Signal Process. Control*, vol. 3, no. 2, pp. 169–174, 2008.

[79] D. Gabor, "Theory of communication. Part 1: The analysis of information," *J. Inst. Elect. Eng.*, vol. 93, no. 26, pp. 429–441, Jul. 1946.

[80] M. Teplan, "Fundamentals of EEG measurement," *Meas. Sci. Technol.*, vol. 2, no. 2, pp. 1–11, 2002.

[81] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, 3rd ed. London, U.K.: Pearson, 2009.

[82] A. Graps, "An introduction to wavelets," *IEEE Comput. Sci. Eng.*, vol. 2, no. 2, pp. 50–61, Jun. 1995.