Robust and Reliable Hand Gesture Recognition with Surface Electromyography (sEMG) Signals via Deep Learning

Pouya Pourakbarian Niaz [1]

Abstract

Recognizing hand motions and gestures using surface electromyography (sEMG) signals has gained significant attention in research and development due to its applications in human-computer interaction, assistive and rehabilitation devices, gaming, and so forth. Since human musculoskeletal dynamics are immensely intricate, processing sEMG signals to extract meaningful information is done via deep learning today. However, many methods have been suggested in the literature, not only for feature engineering but also for architecture and training procedures. Most findings of previous studies apply to a narrow range of motions or scenarios, and methods that perform well under certain conditions have not been compared against one another thoroughly to understand the merits of using them in various motion/gesture recognition scenarios. This work proposes a new model architecture adapted from the existing state-of-the-art to incorporate scenarios in which high-density surface electromyography (HD-sEMG) equipment is used for data acquisition. Furthermore, we propose corrections in the data preparation and feature engineering procedure to suit such scenarios more efficiently. Our comparative results show superior accuracy compared to the latest proposed models in HD-sEMG scenarios in inter-session and inter-subject cross-validation, even without using augmentation techniques such as transfer learning or postprocessing methods.

1. Introduction

Surface Electromyography (sEMG) is a technology used for evaluating and recording the electrical activity produced by skeletal muscles without invading or puncturing the skin (Robertson et al., 2013). With sEMG sensors becoming more affordable and adequately precise in recent years, this technique has found its way into a broad range of applications, such as clinical diagnosis and therapy. Clinical studies using sEMG measurements typically concern muscle fatigue, motor neuron diseases (MND), neuropathic/myopathic conditions, and spinal cord injuries (Drost et al., 2006; Balbinot et al., 2022). Similarly, sEMG is used frequently in motor intention prediction (Bi et al., 2019) for therapeutic research purposes. Another example of an sEMG application involves the prediction of pose and/or joint angles in exoskeletons in the realm of assistive and rehabilitative robotics (Foroutannia et al., 2022). Research has also been conducted into creating more intelligent and intuitive control of prosthetic robotic devices (Jarrah et al., 2022). Due in large part to advances in both human-computer interaction (HCI), human-robot interaction (HRI), and human-machine interface (HMI), gesture and motion recognition have become popular applications of sEMG techniques as well (Ghalyan et al., 2018; Taghizadeh et al., 2021). In physical human-robot interaction (pHRI), sEMG has been used for human intention recognition, typically followed by adaptive admittance/impedance control for minimizing human effort or maximizing task efficiency (Sirintuna et al., 2020).

Surface electromyography typically involves the placement of individual or densely-packed sets of monopolar or bipolar electrodes on human skin on top of the center of a muscle. The sensor can measure the activation levels of the muscle. An example of individually placed bipolar sEMG sensors is the MyoWare®Muscle Sensor AT-04-001 from Advancer Technologies Inc.® (Fig. 1). Examples of packed bipolar dry-electrode sEMG sensors include the Myo Gesture Control Armband from Thalmic Labs® (Fig. 2). Individual sensors such as the Myoware® must be attached to a pair of conductive gel-based patches closely placed on human skin at the center of the muscle, held firmly on the skin via an adhesive layer. They also typically include a third electrode which needs to be connected to a patch placed somewhere outside the muscle where there is no activation, such as another tendon or bone. Densely packed sEMG sensors such as the Myo Armband® only contain a thick and large armband placed on the arm, forearm, legs, thighs,

[1]Robotics and Mechatronics Laboratory & KUIS AI Center, Department of Mechanical Engineering, Koc University, Sariyer 34450, Istanbul, Turkey. Correspondence to: Pouya Pourakbarian Niaz <pniaz20@ku.edu.tr>.

Fall 2022.

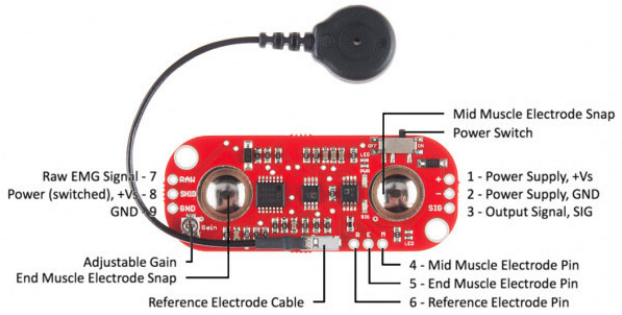

Figure 1. Myoware® AT-04-001 sEMG sensor details. Image from learn.sparkfun.com

Figure 2. Thalmic Labs® Myo Gesture Control Armband. Image from betakit.com

etc., on the circumference of which 6 or 8 bipolar sEMG sensors are placed close together. These sensors are more convenient to use than their counterparts because they do not need to be placed on patches adhering to human skin, nor do they need a third connection to a non-activated part of the limb for referencing. Most sEMG sensor outputs are voltage signals in the 0-10 Volt range.

Due to the immense complexity and high-frequency outputs of the activations occurring in skeletal muscles, sEMG signals that are recorded from the sensors are not trivial to process or decompose. These raw signals contain multiple high-amplitude peaks across a wide range of frequencies and visually resemble audio signals. Even after preliminary amplification and band-pass filtering, extracting meaningful features from the processed signals is difficult, especially when trying to recognize motion/gesture or estimate muscle force from sEMG signals. Therefore, modern approaches in (pre)processing sEMG signals often combine several signals processing, machine learning, and deep learning approaches (Bi et al., 2019; Foroutannia et al., 2022; Sirintuna et al., 2020; Xiong et al., 2021; Clarke et al., 2021).

Due in part to its growing popularity in HCI, HRI, and gaming, and thanks to the availability of a few proper datasets, we have selected motion/gesture recognition from sEMG signals as the application on which we would like to focus. Several studies (Du et al., 2017; Zia ur Rehman et al., 2018; Côté-Allard et al., 2019; Chen et al., 2020; Ozdemir et al., 2020; Shanmuganathan et al., 2020; Yang & Liu, 2022) have attempted to use various deep learning methods to recognize motions/gestures of human arms and hands from sEMG sensors (more frequently using the densely-packed Myo Armbands) under different conditions. Their methods vary greatly in terms of model architectures and training procedures, dataset collection methods, signal (pre)processing techniques, experimental protocols, and so forth. A handful of previous studies provide open-access datasets for other researchers or practitioners to use in their works.

This work not only aims to address the shortcomings of the current models in different scenarios (such as HD-sEMG sensors) but also aims to offer more generalized heuristics for choosing preprocessing methods, feature engineering routines, model architectures, and training procedures to account for scenarios not so well-explored in the literature, as mentioned above.

2. Related Work

A review of the existing literature about gesture recognition from sEMG tells how challenging it is to perform such a task online and potentially use it for instantaneous action in HCI/gaming, as well as adaptive control of prosthetic/rehabilitative devices or robots, in pHRI scenarios, online (Xiong et al., 2021). Most papers studying hand gesture recognition from sEMG have used deep learning, specifically various architectures of recurrent neural networks (RNN), convolutional neural networks (CNN), or a combination of both. Zia ur Rehman et al. (2018) recorded sEMG data from Myo Armbands over multiple days with several subjects, then performed hand motion recognition using CNN from raw sEMG data (end-to-end approach). They then compared their results with other cases where the inputs were preprocessed using linear discriminant analysis (LDA) as well as stacked sparse autoencoders with features (SSAE-f) and raw samples (SSAE-r), and no CNN was used. They concluded that the raw-data-based CNN outperformed other approaches in relatively long-term comparisons, such as between-day comparisons or leave-one-day-out scenarios.

Chen et al. (2020) also used CNN, even though with a much more compact architecture than is typically chosen, to detect hand gestures. They validated their findings on two main-stream hand gesture recognition datasets available online and demonstrated acceptable results obtained with a much lighter architecture. Ozdemir et al. (2020) used only

4 individual bipolar sEMG sensors, which is more sparse than common, especially compared to the 8-channel Myo Armband, which they used to train a relatively deep CNN (a 50-layer ResNet architecture) to classify among 7 different hand gestures using data collected from 30 participants. They used the spectrogram images of sEMG data as inputs to the CNN. Shanmuganathan et al. (2020) used R-CNN (a combination of RNN and CNN) using wavelet-transformed sEMG signals from individual bipolar sEMG sensors to perform hand gesture recognition. They proved that preprocessing the inputs using a wavelet transform improved accuracy over regular feature extraction methods. Yang & Liu (2022) provided a few CNN-based AI frameworks for recognizing and decoding wrist movements in 3D Cartesian space. They used raw sEMG data as the input signal to their CNN architectures and performed experiments related to prosthetic applications to validate their findings.

Since collecting sEMG data requires arduous physical experimentation with multiple subjects, it is not always feasible to collect a sufficiently large and statistically representative dataset for performing the task. To this end, a few researchers have proposed transfer learning techniques in the form of model adaptation or domain adaptation to account for such deficiencies.

Côté-Allard et al. (2019) used transfer learning to perform hand gesture recognition using CNN, using raw and preprocessed data as input. They used a large dataset collected from multiple subjects using Myo armbands. Du et al. (2017) proposed a deep-learning-based domain adaptation framework to enhance sEMG-based inter-session hand gesture recognition.

Despite an abundance of research into sEMG-based gesture/motion recognition, few studies offer an inclusive study of which model or architecture produces more robust and generalizable results, and studies rarely present reliable heuristics in choosing (pre)processing methods to employ under different circumstances. Taking the HD-sEMG scenario as an example case study, this study elucidates the shortcomings of models and methods validated in different settings, e.g., when Myo armbands are used. In addition, this work proposes a variant of the latest models suggested in recent literature, which is more suitable for HD-sEMG scenarios. We also offer corrections in the preprocessing and feature engineering methods that are more generalizable across different experimental settings.

3. Approach

In this study, we investigate how to produce improved gesture recognition performance using deep learning methods, as well as which preprocessing or training methods to select for more portable results. To this end, we use open-access datasets produced by the previous researchers (Du et al., 2017; Chen et al., 2020; Côté-Allard et al., 2019; Pizzolato et al., 2017). Since the experiments in these datasets have been performed under a wide range of circumstances, they will collectively offer a great means for comparison and contrast of models and approaches.

3.1. Datasets

Three datasets are widely used for hand gesture recognition using EMG. The most widely known is the "NinaPro DB5" dataset (Pizzolato et al., 2017), in which two dry-electrode Myo armbands (Thalmic Labs, Inc.®) are used (16 channels in total) for detecting a collective set of 52 gestures under different circumstances.

Another dataset called "Myo Armband" is reported by Côté-Allard et al. (2019). They use only one Myo armband and detect a subset of the gestures used in the "NinaPro DB5" dataset with relatively high performance. They use a convolutional neural network to achieve their results.

Another slightly different dataset is called the "Capg Myo" dataset, reported by Du et al. (2017), in which 8 high-density bands (HD-sEMG) with 16 densely-packed sensor nodes on each are placed on the arm (128 channels in total). This dataset also proposes a convolutional neural network architecture for detecting hand gestures based on HD-sEMG data. Table 1 summarizes the prominent characteristics of each dataset.

We selected the *Capg Myo* dataset (Du et al., 2017) for our study. This dataset involves 23 human subjects of age 23 to 26 years who conduct experiments with 22 hand gestures and basic hand motions. The dataset contains 3 sub-datasets, out of which the first one (Dubbed "DB-a") is used in this study for simplicity. This dataset contains 18 subjects performing 8 isometric and isotonic hand gestures, with 10 repetitions per gesture. Each gesture lasts about 3-10 seconds, and a 7-second rest is included between the trials. The data is collected using 8 HD-sEMG (High-Density Surface Electromyography) bands, each containing 16 densely packed channels. In other words, 128 channels of sEMG data are collected from the right arm of all subjects. After collecting the data at 1000 Hz, it is passed through a notch filter to remove the interference of the powerline at 50 Hz and its harmonics. Then, the data is band-pass filtered between 20-380 Hz to extract the useful envelope of the signal.

3.2. The State of the Art

The current state of the art in hand gesture recognition from EMG is reported by Chen et al. (2020). They proposed a relatively compact convolutional neural network architecture called "EMGNet," and evaluated it on the *NinaPro*

Table 1. Prominent datasets used for hand gesture recognition.

	Myo Armband	**NinaPro DB5**	**Capg Myo**
Subjects	19	10	23
Channels	8	16	128
Sampling (Hz)	200	200	1000
Electrode	Dry	Dry	Wet (Gel)
Measurement	1 × Myo band	2 × Myo band	8 × HD-sEMG
Gestures	7	53	22

DB5 (Pizzolato et al., 2017) and the *Myo Armband* (Côté-Allard et al., 2019) datasets. They reported state-of-the-art performance in both cases of using and not using transfer learning for inter-subject testing. Both datasets above are highly similar (see Table 1) in that they both use the same type and brand of sensors, and they both use similar gestures, procedures, and sampling frequencies.

There are significant differences between the *Capg Myo* (Du et al., 2017) dataset and the other two, the most obvious ones being the type of sensory equipment and acquisition setup and the sampling frequency. The *Capg Myo* dataset uses high-density EMG sensors (HD-sEMG), meaning the EMG data covers a wide range of spatial dimensions on the body and a total of 128 channels of data are sampled at 1000 Hz. It stands to reason that some of the state-of-the-art methods and architectures may not work as well on this dataset as on the previous two. Therefore, in this study, we focus on evaluating the efficacy of the architecture and procedure proposed by Chen et al. (2020) on HD-sEMG scenarios like the *Capg Myo* dataset, meanwhile proposing a modified architecture and procedure for such scenarios.

3.3. Proposed Model: 3D EMGNet

Since there are 8 HD-sEMG bands with 16 channels in the *Capg Myo* scenario, Du et al. (2017) interpret the data at each time step as an 8×16 image, not considering the time dimension. They propose a CNN architecture called "ConvNet" with typical 2D convolution layers, which only processes the instantaneous data, disregarding the temporal dependency completely. Fig. 3 shows the architecture they use. We believe this approach may not apply to many scenarios since temporal dependencies in time-series classification and prediction are usually of great importance, especially in biomedical signals. On the other hand, considering the performance achieved by Chen et al. (2020) and others before them in data that use Myo armbands, higher frequencies than 200 Hz may seem a bit redundant for acceptable performance in hand gesture recognition.

Regarding feature engineering, Chen et al. (2020) propose continuous wavelet transform (CWT), which can be done on 8 or 16 channels of time series data as seen in the *NinaPro DB5* and the *Myo Armband* datasets but will prove forbiddingly computational cost-prohibitive for the 128 channels of data acquired in the *Capg Myo* dataset. It is impractical to use that method, especially in online (real-time) deployment.

Without using any time-domain or frequency-domain features that might be too expensive in HD-sEMG settings, the simplest way of including temporal data is to add a third "time" dimension to the 2D images, effectively producing videos (sequences of 2D grayscale images) and then using 3D convolutional layers on these 3D images, while retaining the *EMGNet* architecture and training procedure. The justification here is that any useful temporal dependency can be captured using convolution (as is often attempted when using 1D convolutions on time series data), in addition to spatial dependencies within the 2D images representing data coming from 8 HD-sEMG bands placed around the arm.

In summary, to perform hand gesture recognition in HD-SMG scenarios such as the *Capg Myo* dataset, we propose the following, consequently naming it the "3D EMGNet" model.

- Filter and preprocess the data as suggested by Du et al. (2017).
- Down-sampling data at 200 Hz as with previous datasets.
- Using sliding windows of L_{in} time steps to extract sequences of 8×16 2D images, thereby making $L_{\text{in}} \times 8 \times 16$ 3D images.
- Using the *EMGNet* architecture, but using 3D convolution, pooling and batch-normalization layers in place of their 2D counterparts, thereby making the *3D EMGNet* model.
- Using EMG data directly without performing feature engineering procedures in the time or frequency domain.
- Using the training procedure reported by Chen et al. (2020) for the EMGNet dataset, only with higher mini-batch sizes to accelerate training.

3.4. Model Architecture

The architecture we suggest for HD-sEMG applications is a variant of the EMGNet architecture suggested by Chen et al. (2020). Fig. 4 shows our architecture for 3D images, with time being the first dimension and image height and width being the following two dimensions. Our modification to the EMGNet architecture turns all 2D layers (convolution, batch-normalization, and pooling) to their 3D counterparts, constituting "3D EMGNet". This architecture has batch-normalization and ReLU layers at the very beginning to extract only CWT coefficients corresponding

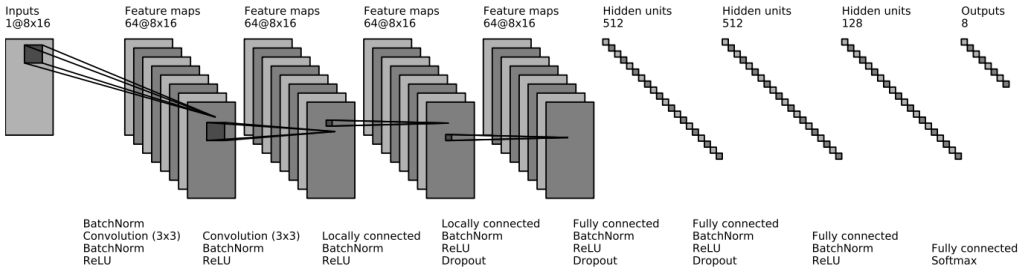

Figure 3. The *ConvNet* architecture, as suggested by Du et al. (2017). The image is adapted from their paper directly.

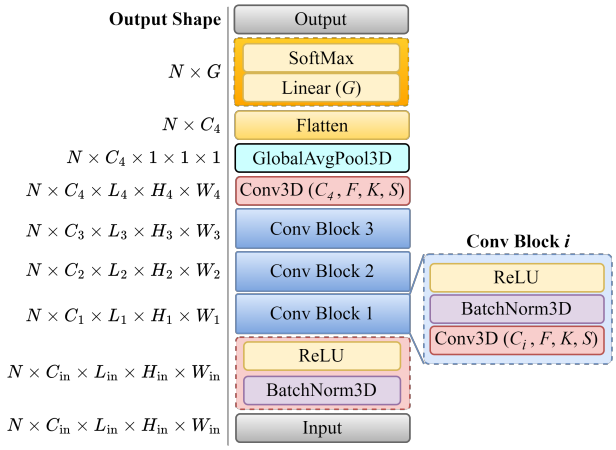

Figure 4. The *3D EMGNet* architecture, a variant of the *EMGNet* architecture suggested by Chen et al. (2020)

Table 2. Nomenclature used in Fig. 4 regarding the details of the *3D EMGNet* architecture.

Parameter	Description	Value
N	Mini-batch size	512
C_{in}	Number of input channels	1
L_{in}	Input sequence length	64
H_{in}	Input image height	8
W_{in}	Input image width	16
C_i	Filters in the i'th Conv3D layer	16, 32, 32, 64
L_i	Output length of i'th Conv block	Varies
H_i	Output height of i'th Conv block	Varies
W_i	Output width of i'th Conv block	Varies
G	Number of gestures (output size)	8
F	Conv3D filter size	3
K	Conv3D padding size	0
S	Conv3D stride	1

to frequencies with higher-than-average amplitudes. Then, 3 convolution blocks follow, each containing convolution, batch-normalization, and ReLU layers. After the final convolution layer, a global average pooling layer is included to generate a condensed image before flattening, thereby decreasing model size and the number of parameters. Fig. 4 also demonstrates the dimension of the tensors coming out of every module. The nomenclature used in this figure is explained in Table 2 along with our chosen values (as suggested by Chen et al. (2020) for *EMGNet*) for these values.

Chen et al. (2020) show that this architecture is much lighter than the existing state-of-the-art proposed by the literature. They also achieve better performance metrics and computational advantage than the previous CNN models, which is why we select their model for our study, albeit with some modifications. Since the convolution, batch-normalization, and pooling layers are 3D in our model, time is another image dimension. There is no need for feature engineering such as CWT, etc.

The filter kernels in convolution layers are shared across the input images, meaning the number of learnable parameters of convolution layers is only a function of the kernel size and the number of filters, not the input image size. The number of learnable weights and biases of Batch-Normalization layers only depend on the number of channels they are meant to normalize (number of input features). Pooling and ReLU layers have no learnable parameters. In CNN models, often the largest number of learnable parameters comes from the Flatten layer at the end of the CNN encoding part, just before the Dense (fully-connected) section. Every channel of every pixel across the time, height and width of the output image of the final layer in the encoder section is mapped to a separate feature, processed by the Dense section. This means that the input of the Dense section has a very large number of features, thereby increasing the size of the learnable weight/bias tensors. One effective way of reducing this size (apart from reducing input image size) is introducing global pooling layers such global average pooling (GlobalAvgPool3D) and global max pooling (GlobalMaxPool3D) layers at the end of the encoder part. These layers summarize the entire image into a single pixel, making the output much smaller, with much fewer features to be mapped to the Dense network that follows. This is a feature that the *EMGNet* and the *3D EMGNet* both take advantage of.

Table 3. Hyperparameters for the training procedure. The Adam optimizer is used.

Parameter	Description	Value
α	Learning rate for Adam	0.01
β_1	First momentum of Adam	0.9
β_2	Second momentum of Adam	0.999
ϵ	Epsilon value of Adam	1.0 e-8
γ	Learning rate exponential decay	0.9
λ	L_2 regularization parameter	1.0 e-4
M_{tot}	(Maximum) number of epochs	60
M_{pat}	Early stopping patience epochs	N/A

3.5. Training Procedure

The training procedure we follow is similar to that suggested by (Chen et al., 2020) and previous studies before them, with little modifications to the minibatch size, learning rate schedule, and epoch count. The training procedure for performing hand gesture recognition using the *DB-a* subset of the *Capg Myo* dataset is similar to any supervised multi-class classification problem. The loss function is categorical cross-entropy, the equivalent of a negative log-likelihood loss function following a SoftMax activation function at the output layer. We are using the Adam optimizer due to its popularity and widespread success in deep learning.

Table 3 explains the nomenclature of the hyperparameters used for the training procedure. The evaluation of the model is done using the model's accuracy on the validation set at every epoch. The validation data may come from the training set or the test set. This way, by observing how the training and validation errors and/or accuracies change throughout training, we can understand whether or not there is high bias (underfitting), high variance (overfitting), or both.

Fig. 6 shows an example of a training history for this architecture, trained on the *Capg Myo DB-a* dataset. Training is done using PyTorch 1.12.1 package in Python, and PyTorch's default parameters are kept for all layers, modules and training procedures, if not mentioned in Table 3.

3.6. Cross-Validation Methods

Since several subjects are experimenting with various conditions with some repetition, cross-validation for any training scenario can help account for the existing stochasticity in the data, thereby providing more reliable statistics for evaluating the training procedure and the model architecture. Using k-fold cross-validation is common for machine-learning tasks that use data from a wide range of subjects and conditions. As such, we have also performed k-fold cross-validation on our training scenario. The different scenarios used for the cross-validation are described hereunder. In general, it

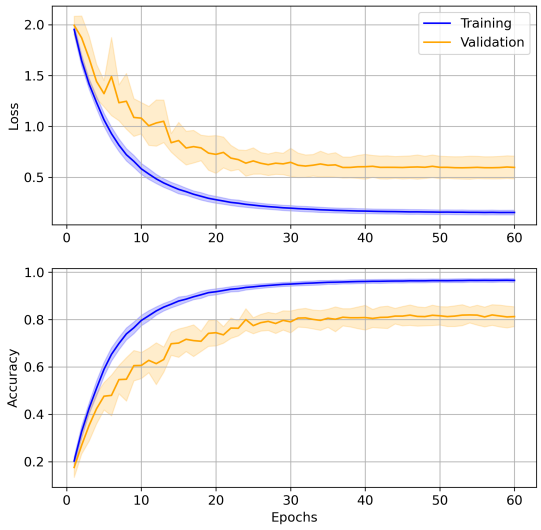

Figure 5. Example training history, containing loss (epoch loss for the entire training set) and accuracy values for both training and validation data. Solid curves are means across the k-fold cross-validation trials, and the shaded areas are standard deviations.

is assumed that n_S subjects have experimented under n_C different conditions ($n_C = G = 8$ for the *Capg Myo DB-a* dataset), with n_R repetitions (trials) under every condition.

3.6.1. Inter-Session Testing

This cross-validation method assumes that all the subjects have performed several trials and the model is trained on the existing data. Then, the model performance is evaluated on new trials of the same subjects. During training, out of the available n_R trials for every condition of every subject, $n_R - 1$ are used for training, and 1 is used for testing. Since there are n_R ways of choosing one trial out of n_R for testing and keeping the rest for training, this is n_R-fold cross-validation.

3.6.2. Inter-Subject Testing

In this scenario, it is assumed that the model has access to the training data of many subjects that have experimented repetitively under all possible conditions. A model has been trained on that data. There is now a new subject whose data is unseen by previous training. We seek to evaluate the trained model's performance on the new subject's data or figure out a way to retrain the model on the new subject to make it more adaptable. Therefore, one subject out of n_S is kept for testing, and the others are used for training. Since we can switch the subject used for testing, this is an n_S-fold

cross-validation setting.

4. Results

4.1. Accuracies

Results of training the "3D EMG" architecture on the Capg Myo DB-a dataset while performing k-fold cross-validation as described previously, are summarized in Table 4. As seen in this table, our model superseded the performance of the CNN model suggested by the *Capg Myo* dataset's creators (Du et al., 2017), called "ConvNet". As mentioned before, their architecture comprised many 2D CNN layers, processing instantaneous 8×16 images without any temporal information. Our model shows better performance in both inter-session and inter-subject scenarios than the ConvNet architecture suggested by Du et al. (2017) for HD-sEMG scenarios.

The low performance and catastrophic overfitting seen in the inter-subject testing are normal in EMG-related tasks. This problem is commonly tackled via transfer learning. Specifically, domain adaptation is made using a method called "AdaBN" (Adaptive Batch-Normalization) (Du et al., 2017; Côté-Allard et al., 2019; Chen et al., 2020). In this study, we have not made any domain adaptation. The idea is that whichever model/method achieves the best results without transfer learning is bound to achieve equally superior results when using transfer learning. In other words, the effect of transfer learning is similar in almost all methods and model architectures. Selection of a model/method can be made without transfer learning to hasten the decision-making, after which, when deployed in real-life inter-subject or inter-session scenarios, transfer learning can be used to improve the results.

Similar to transfer learning, a common method for increasing performance is using post-processing, such as a majority voting scheme, so that the raw outputs of the classifier are averaged over a short period of time. We also have not performed any such postprocessing, with the same justification that any model/method achieving superior performance without such postprocessing will ensure better improvements when using such methods.

4.2. Training Histories

The average training history of 3D EMGNet on Capg Myo DB-a is shown in Fig. 6. Similarly, the average training history for this model in the inter-subject scenario is shown in Fig. **??**. In both these figures, the validation data is chosen randomly from 5% of the test set data. The figures show acceptable performance in inter-session testing but catastrophic overfitting in the inter-subject testing scenario. As mentioned, this is typical in EMG studies, commonly remedied by transfer learning methods.

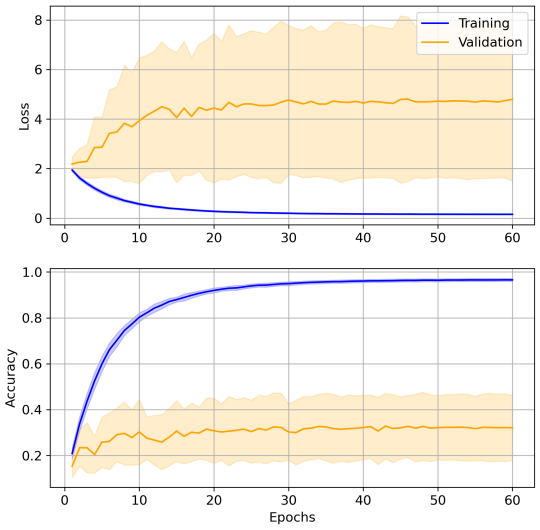

Figure 6. Training history for *3D EMGNet* on the *Capg Myo DB-a* dataset under the inter-subject scenario, containing loss (epoch loss for the entire training set) and accuracy values for both training and validation data. Solid curves are means across the k-fold cross-validation trials, and the shaded areas are standard deviations.

4.3. Training Times

Since training would take too long when using all the data, before separating training, validation, and testing data, only 10% of all the data is extracted randomly. While training the *3D EMGNet* architecture on the *Capg Myo DB-a* dataset, we observed an average training time of 68 ± 12 seconds per epoch in the inter-subject scenario and approximately 65 ± 7 seconds per epoch for the inter-session scenario. With a minibatch size of 512 as chosen (and considering that only 10% of all the data is chosen), there are 51 mini-batches in the inter-subject scenario and ?? mini-batches in the inter-session scenario. In both scenarios, training was done using PyTorch 1.12.1 on a high-performance computing cluster with 4 CPUs, 1 GPU, and 32 GB of RAM. The CPUs were Intel®Xeon®Gold 6248, and the GPUs were NVidia®Tesla®T4.

5. Conclusion

The present work investigates hand gesture recognition using deep learning under various circumstances. Three datasets, the *NinaPro DB5* dataset (Pizzolato et al., 2017), the *Myo Armband* dataset (Côté-Allard et al., 2019), and the *Capg Myo* dataset (Du et al., 2017) are reviewed and studied. In light of state of the art proposed for settings in which Myo armbands are used (Chen et al., 2020), a

Table 4. Summary of accuracies for training, validation, and test sets in inter-session and inter-subject cross-validation scenarios when using the *ConvNet* architecture proposed by Du et al. (2017), versus our proposed model, *3D EMGNet*.

Approach	Method	Statistic	Training Set	Validation Set	Test Set
Inter-Session	ConvNet	mean	0.7272	0.7497	0.7543
		std	*0.0105*	*0.0244*	*0.0222*
	EMGNet	mean	0.9652	0.8125	**0.8167**
		std	*0.0082*	*0.0439*	***0.0325***
Inter-Subject	ConvNet	mean	0.7373	0.2401	0.2432
		std	*0.0110*	*0.0848*	*0.0833*
	EMGNet	mean	0.9656	0.3208	**0.3017**
		std	*0.0064*	*0.1447*	***0.1432***

3D CNN architecture called "3D EMGNet" is suggested here for HD-sEMG settings where there are too many channels for feature extraction techniques such as CWT to be feasible.

Using the *3D EMGNet* architecture on 3D images, time is the first dimension, followed by the number of HD-sEMG bands and the number of channels in every band as the height and width of every image, respectively. This way, better performance is achieved in both inter-session and inter-subject scenarios compared with the *ConvNet* architecture suggested in the literature for HD-sEMG settings. It is therefore suggested that no matter the type of sensor used, the data can be downsampled at 200 Hz after filtering.

In the HD-sEMG case, feature engineering, as in CWT, is unnecessary and cost-prohibitive. It can therefore be concluded that the EMGNet architecture is suitable for both low-density and high-density EMG scenarios, with the prominent difference being that in low-density scenarios, CWT is the preferred feature engineering approach to generate multi-channel 2D images from time series signals, whereas in high-density scenarios, time should simply be added to 2D images of instantaneous data, to generate single-channel (grayscale) 3D images.

Without touching the architecture of the EMGNet, convolution, pooling and batch-normalization layers should be 2D variants in the low-density case, and 3D in the high-density case. In inter-subject scenarios, the results achieved by 3D EMGNet also suggest that methods such as transfer learning and majority voting schemes will yield better outcomes when tried on 3D EMGNet in the HD-sEMG case.

References

Balbinot, G., Joner Wiest, M., Li, G., Pakosh, M., Cesar Furlan, J., Kalsi-Ryan, S., and Zariffa, J. The use of surface emg in neurorehabilitation following traumatic spinal cord injury: A scoping review. *Clinical Neurophysiology*, 138:61–73, 2022. ISSN 1388-2457. doi: https://doi.org/10.1016/j.clinph.2022.02.028. URL https://www.sciencedirect.com/science/article/pii/S1388245722002103.

Bi, L., Feleke, A. G., and Guan, C. A review on emg-based motor intention prediction of continuous human upper limb motion for human-robot collaboration. *Biomedical Signal Processing and Control*, 51:113–127, 2019. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2019.02.011. URL https://www.sciencedirect.com/science/article/pii/S1746809419300473.

Chen, L., Fu, J., Wu, Y., Li, H., and Zheng, B. Hand gesture recognition using compact cnn via surface electromyography signals. *Sensors*, 20(3), 2020. ISSN 1424-8220. doi: 10.3390/s20030672. URL https://www.mdpi.com/1424-8220/20/3/672.

Clarke, A. K., Atashzar, S. F., Vecchio, A. D., Barsakcioglu, D., Muceli, S., Bentley, P., Urh, F., Holobar, A., and Farina, D. Deep learning for robust decomposition of high-density surface emg signals. *IEEE Transactions on Biomedical Engineering*, 68(2):526–534, Feb 2021. ISSN 1558-2531. doi: 10.1109/TBME.2020.3006508.

Côté-Allard, U., Fall, C. L., Drouin, A., Campeau-Lecours, A., Gosselin, C., Glette, K., Laviolette, F., and Gosselin, B. Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4):760–771, April 2019. ISSN 1558-0210. doi: 10.1109/TNSRE.2019.2896269.

Drost, G., Stegeman, D. F., van Engelen, B. G., and Zwarts, M. J. Clinical applications of high-density surface emg: A systematic review. *Journal of Electromyography and Kinesiology*, 16(6):586–602, 2006. ISSN 1050-6411. doi: https://doi.org/10.1016/j.jelekin.2006.09.005. URL https://www.sciencedirect.com/science/article/pii/S1050641106001209. Special Section (pp. 541–610): 2006 ISEK Congress.

Du, Y., Jin, W., Wei, W., Hu, Y., and Geng, W. Surface emg-based inter-session gesture recognition enhanced by

deep domain adaptation. *Sensors*, 17(3), 2017. ISSN 1424-8220. doi: 10.3390/s17030458. URL https://www.mdpi.com/1424-8220/17/3/458.

Foroutannia, A., Akbarzadeh-T, M.-R., and Akbarzadeh, A. A deep learning strategy for emg-based joint position prediction in hip exoskeleton assistive robots. *Biomedical Signal Processing and Control*, 75:103557, 2022. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2022.103557. URL https://www.sciencedirect.com/science/article/pii/S1746809422000799.

Ghalyan, I. F., Abouelenin, Z. M., and Kapila, V. Gaussian filtering of emg signals for improved hand gesture classification. In *2018 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pp. 1–6, 2018. doi: 10.1109/SPMB.2018.8615596.

Jarrah, Y. A., Asogbon, M. G., Samuel, O. W., Wang, X., Zhu, M., Nsugbe, E., Chen, S., and Li, G. High-density surface emg signal quality enhancement via optimized filtering technique for amputees' motion intent characterization towards intuitive prostheses control. *Biomedical Signal Processing and Control*, 74:103497, 2022. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2022.103497. URL https://www.sciencedirect.com/science/article/pii/S1746809422000192.

Ozdemir, M. A., Kisa, D. H., Guren, O., Onan, A., and Akan, A. Emg based hand gesture recognition using deep learning. In *2020 Medical Technologies Congress (TIPTEKNO)*, pp. 1–4, Nov 2020. doi: 10.1109/TIPTEKNO50054.2020.9299264.

Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H., and Atzori, M. Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLOS ONE*, 12(10):1–17, 10 2017. doi: 10.1371/journal.pone.0186132. URL https://doi.org/10.1371/journal.pone.0186132.

Robertson, D., Caldwell, G., Hamill, J., Kamen, G., and Whittlesey, S. *Research Methods in Biomechanics*. Human Kinetics, 2013. ISBN 9781492581857. URL https://books.google.com.tr/books?id=_u56DwAAQBAJ.

Shanmuganathan, V., Yesudhas, H. R., Khan, M. S., Khari, M., and Gandomi, A. H. R-cnn and wavelet feature extraction for hand gesture recognition with emg signals. *Neural Computing and Applications*, 32:16723–16736, 11 2020. ISSN 14333058. doi: 10.1007/S00521-020-05349-W/FIGURES/13. URL https://link.springer.com/article/10.1007/s00521-020-05349-w.

Sirintuna, D., Ozdamar, I., Aydin, Y., and Basdogan, C. Detecting human motion intention during phri using artificial neural networks trained by emg signals. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pp. 1280–1287, 2020. doi: 10.1109/RO-MAN47096.2020.9223438.

Taghizadeh, Z., Rashidi, S., and Shalbaf, A. Finger movements classification based on fractional fourier transform coefficients extracted from surface emg signals. *Biomedical Signal Processing and Control*, 68:102573, 2021. ISSN 1746-8094. doi: https://doi.org/10.1016/j.bspc.2021.102573. URL https://www.sciencedirect.com/science/article/pii/S1746809421001701.

Xiong, D., Zhang, D., Zhao, X., and Zhao, Y. Deep learning for emg-based human-machine interaction: A review. *IEEE/CAA Journal of Automatica Sinica*, 8(3):512–533, March 2021. ISSN 2329-9274. doi: 10.1109/JAS.2021.1003865.

Yang, D. and Liu, H. An emg-based deep learning approach for multi-dof wrist movement decoding. *IEEE Transactions on Industrial Electronics*, 69(7):7099–7108, July 2022. ISSN 1557-9948. doi: 10.1109/TIE.2021.3097666.

Zia ur Rehman, M., Waris, A., Gilani, S. O., Jochumsen, M., Niazi, I. K., Jamil, M., Farina, D., and Kamavuako, E. N. Multiday emg-based classification of hand motions with deep learning techniques. *Sensors*, 18(8), 2018. ISSN 1424-8220. doi: 10.3390/s18082497. URL https://www.mdpi.com/1424-8220/18/8/2497.