ORIGINAL ARTICLE

Gesture detection from sEMG signals based on similarity learning

Semih Celik[1] · Ulug Bayazit[1]

Received: 23 December 2024 / Accepted: 23 May 2025 / Published online: 21 June 2025

Abstract

EMG (electromyography) signals provide critical information on muscle activity and have long been used in prosthetics and health care. With the rise of wearable technologies, EMG is now widely applied in human–computer interaction, rehabilitation and emerging remote control applications such as drone operation and gaming. The primary objective of this study was to accurately identify hand gestures utilizing a limited quantity of user electromyography (EMG) data, thereby improving the applicability and extending the potential use cases of EMG-based systems in real-world scenarios. The proposed method makes use of deep learning methods, namely convolutional neural networks and transfer learning to learn the similarity between samples of EMG signals to predict hand gestures. A pretrained network, fine-tuned in multiple passes by progressive freezing of layers, is applied to extract subject-specific features. Support vector machines is used to learn the similarity of the subject's features to features in the training set to facilitate the voting for the best hand gesture by means of k-nearest neighbors method. The findings indicate viability of the proposed method even when utilizing a limited dataset collected from the user. Average accuracy obtained by the proposed method on 17 test subjects for 7 hand gestures exceeds the performances of the three approaches proposed in two related works as well as an approach based on recurrent neural network. The proposed method was also evaluated on the NinaPro DB5 Exercise B dataset, where it was compared against four existing methods. Experimental results demonstrate that the proposed approach outperforms all compared methods in terms of classification accuracy.

Keywords EMG classification · Transfer learning · CNN · Hand gesture detection

1 Introduction

EMG signals have several utilities. As a primary area of application, controlling prosthetic limbs with EMG signals [1, 2] has been of interest for several decades. Controlling robots or drones with EMG signals [3, 4] also represents a natural way of human–robot interaction. Exoskeleton control [5] is another area of EMG application. As human–robot interaction and human–computer interaction received increasing attention in recent years, EMG signals provided alternative or additional options to classical approaches like vision and sound. When EMG signals from multiple limbs could be collected, it becomes possible to classify human actions at a higher semantic level.

EMG signals are grouped into the categories of surface EMG (sEMG) and intramuscular EMG (iEMG) signals according to the location of the electrodes. In this study, we focus on sEMG signals. Surface electromyography (sEMG) signals carry intricate information about a user's intent through muscular contractions, as discerned by surface electrodes. The noninvasive acquisition of sEMG signals can be seamlessly incorporated into a control

 https://doi.org/10.1007/s00521-025-11393-1

Table 1 LCNN and Conv2D comparison based on single-pass and multi-pass fine-tuning options

Network structure	SP FT (%)	MP FT (%)	SP FT+SVM+kNN (%)	MP FT +SVM+kNN (%)
LCNN	96.31	97.36	95.72	97.39
Conv2D	98.29	98.44	98.42	98.77

framework to enable proportional control over force or speed [6]. Hand gesture classification, rehabilitation of muscle disorders, robotic control systems and silent speech recognition are some of the application areas of sEMG signals [7]. Many of these applications could be grouped into three categories: speech recognition, robotic applications and diagnostics applications [8]. These applications have various constraints on and considerations of dataset acquisition time, accuracy and robustness against noise.

Some professional and medical usage allow for the acquisition of large amounts of data. For example, a physician could collect large amounts of data in a clinical environment in order to detect muscle disorders by using an intelligent system that learns from data. On the other hand, some daily usage practices require one to work with very little data collected in an uncontrolled environment. For example, to control the operations of an unmanned aerial vehicle, the training data collection time has to be as short as calibration time. The requirements for data size exhibit significant disparities across diverse application domains. With this perspective in mind, the current study focuses on achieving high-accuracy results for small personalized dataset for the subjects.

Specifically, we have developed a performance efficient model with a small size fine-tuning data recorded for a single user in one of the above-mentioned application areas. Improved model success rates, together with reduced data size, increase the applicability of the model. The proposed method is centered around a deep learning system for successfully detecting hand gestures from EMG data in a time window. Pretraining of the system is performed on universal data collected from multiple subjects. Final training (transfer learning) of the system is performed with personal data collected from each subject to more precisely learn that subject's intents (gestures).

The proposed deep learning system is structured in three stages. The first stage extracts a feature vector using a convolutional neural network (CNN) from the sEMG signals of the user. The second stage matches this feature vector with the best feature vectors from the set of all (fine-tuned) training feature vectors for the same user. In the third stage, the hand gesture class labels associated with the best feature vectors are aggregated within the k-nearest neighbor (kNN) framework to yield the estimate for the gesture class.

While the CNN network is trained and fine-tuned to produce the optimal feature vector for the gesture by using classical gradient decent and back propagation, the similarity between two feature vectors is learned by employing SVM regression. The kNN voting at the end handles label differences near class boundaries and ensures that the system is robust against outlier training feature vectors.

An important feature of the current work is that overfitting to fine-tuning data is avoided by progressive freezing of the system before the second stage is designed rather than jointly designing the first two stages. This approach decouples the first two stages and imparts robustness to the overall system.

In summary, we pose two distinctive features of our approach as our contributions in the domain of CNN-based sEMG classification. Firstly fine-tuning of the pretrained CNN feature extraction network is very efficiently performed by using a multi-pass approach, particularly in the face of limited data. Secondly, the incorporation of similarity learning in conjunction with the KNN method enhances overall performance.

The organization of this article is as follows. In Sect. 2, we cover the related works in the area of EMG classification. In Sect. 3, we describe the dataset used in our study. In Sect. 4, we explain our proposed EMG classification model in detail. In Sect. 5, we present our experiments and analysis. In Sect. 6, we summarize the method and results and give future research directions.

2 Related works

A wide range of research studies have been conducted on sEMG (surface electromyography) signals for controlling devices, robots or prosthesis [1–3], addressing muscle disorders and rehabilitation [9, 10], employing them in human–computer interface [12], authenticating users [13, 14] and understanding sign languages [15, 16]. An abundance of studies exists that focus on a variety of datasets, methods, objectives and performance criteria. For example, muscle disorder detection is performed offline and the dataset includes normal and diseased/injured muscle data records [17]. On the other hand, remote device control has been performed by real-time classification and a small dataset is gathered just from healthy subjects [3].

Some approaches focus on generating better features from input data. In [1], Arvetti, Gini and Folgheraiter developed a model to detect five different movements with two EMG sensors, in order to control active hand prosthesis. The method consists of an amplifier, feature extraction unit (including short-time Fourier transform and wavelet) and ANN (artificial neural network). The reported accuracies were 86% and 96.77% for five movements of each of the two subjects. Kocyigit and Korurek [18] developed a method which made use of wavelet transform for feature extraction, PCA (principal component analysis) for dimension reduction and FCM (fuzzy c-means)/PCM (possibilistic c-means) and FKNN (fuzzy K-nearest neighbor) for classification. This method yielded 95.7% average classification accuracy on 4 classes. Shroffe and Manimegalai [19] proposed a method that extracts handcrafted features based on the frequency and distribution of each time window within the signal. These features have been fed to an ANN, and the network is used to classify the EMG signal. Average cross-validation accuracies between 92% and 98% has been reported for three subjects. An additional investigation [20] focused on time-domain electromyography (EMG) features, commonly employed in the context of human arm prosthetics. Narayan [21] examines features in both the frequency domain and time–frequency domain, providing KNN accuracy results of each feature as part of the analysis.

Support vector machine is another method widely used in sEMG signal classification. Crawford, Miller, Shenoy and Raos [3] proposed to employ linear support vector machines on channel amplitude features in order to solve an 8-class robotic and prosthetic hand control problem. The data has been collected with sensors placed on 7 selected muscles. The method achieves accuracies over 90% for three subjects.

One of the basic design choices of a EMG signal classification is the exploitation of the temporal information in the signal. One approach is to create 2D data by applying sliding window and use 2D convolutional neural network. Atzori, Cognolato and Muller [2] presented a method for classifying a large number of hand movements for the purpose of controlling a robotic hand which represents the test environment for the control of a prosthetic hand. The study includes an average of 50 hand movements each from 67 (dataset 1 and dataset 2) intact and 11 (dataset 3) transradial amputees and compares classical methods with the proposed simple convolutional neural network model. The accuracy of the proposed CNN model is measured as 66.59% on dataset 1, 60.27% on dataset 2 and 38.09% on dataset 3.

Allard et al. [4] conducted a study that has employed an easy to use sEMG sensor in order to guide a robotic device by detecting gestures. The study has adopted Myo Armband (Thalmic Labs) which has 8 sensors with 200 MHz rate. Recalibration is considered as an important design issue. Therefore, the data has been collected on 6 consecutive days in order to experiment without recalibration. The experiments show that the proposed convolutional neural network with frequency features achieves 97.9% accuracy for 7 hand/wrist gestures.

Another way of exploiting temporal information in EMG signal is proposed by Sun et al. [22]. They propose a method which works by applying generative flow model to get a factorized feature that is classified by a linear softmax classifier. The method is applied to a dataset of 53 different hand gestures recorded with 16 sensors (two lines of 8 sensors). The experiments show that the method achieves an average accuracy of 63.86% on the Ninapro database 5.

Vasiliev and Melnikov [23] employed a recurrent neural network, specifically a long short-term memory (LSTM) architecture, to capture temporal dependencies in sEMG signals. Their experiments demonstrated that their LSTM-based model achieved a notable accuracy of 94.59%, outperforming the 1D-CNN. A related approach was proposed by Qi et al. [24], who introduced an LSTM-RNN-based architecture for multi-sensor gesture classification. Similarly, Wu et al. [25] proposed a hybrid model combining LSTM and CNN layers to enhance classification performance.

Given the challenges of limited dataset size and discrepancies in the distribution between training and test datasets encountered by deep learning models, certain studies concentrate on the application of transfer learning. Transfer learning approaches commonly fall within five categories [26]: inter-subject [27–29], electrodes variation [31], intersession, modality variation and extensive learning .

To implement inter-subject transfer learning, Allard et al. [27] created two datasets by recording 18 and 17 able-bodied subjects. The study focuses on transfer learning in order to increase the accuracy for an individual. Baseline-CNN and proposed CNN with transfer learning have been compared. The experiments have been conducted with a low-to-high amount of fine-tuning data and has shown that the proposed method achieves 97.81% accuracy which is better than the accuracy of baseline-CNN.

Lehmler et al. [28] proposed two distinct pretrained deep learning models and employed weight initialization for the recalibration of new subjects' data. Through experimentation across diverse datasets and settings, their findings indicate that fine-tuning enhances accuracy by approximately 5%, relative to pretrained models. Moreover, their results demonstrate approximately 12% improvement over subject-specific models.

Kim, Guan and Lee [29] introduced an alternative inter-subject transfer learning method employing a leave-one-out strategy. The authors trained CNN models on datasets generated through the leave-one-out approach. Subsequently, they ranked and selected the top-n models to form a model set, denoted as supportive CNNs, for each subject. Each supportive CNN underwent fine-tuning using the data specific to the target subject. Classification was executed through majority voting among the fine-tuned supportive CNNs.

Several studies have concentrated on enhancing fine-tuning schemes, with a notable focus on parameter-efficient fine-tuning (PEFT). Lialin, Deshpande and Rumshisky [32] provide a comprehensive survey encompassing 40 papers on PEFT methods. The studies are systematically categorized into three distinct groups: additive methods, selective methods, reparametrization-based methods and hybrid methods.

Liu et al. [33] investigate few-shot in-context learning and propose a method called T-Few. Liao, Meng and Monz [34] introduced a sparse mask alongside an innovative adapter technique to facilitate task-specific fine-tuning without introducing additional latency.

Certain investigations concentrate on the refinement of selective layer tuning [35, 36], whereas others center their attention on two-stage fine-tuning strategies [37] aimed at addressing challenges such as the bias–variance trade-off [38] and issues related to class imbalance [39]. Sarhan, Lauri and Frintrop [40] introduced a fine-tuning framework named "multi-phase." In this proposed methodology, the training process involves successive train (fine-tune) phases wherein layers are unfrozen for training, commencing from the uppermost layers of the network.

In a related field of EEG classification, Calhas, Romero and Henriques [41] propose a pairwise distance learning method for Schizophrenia diagnosis. They advocate for the utilization of a Siamese neural network architecture as a means to acquire a discriminative feature space through the exploration of pairwise combinations of observations. The results indicate 15–20% accuracy increase over alternatives. An alternative investigation [42] centers on examining sample similarity through a comparative analysis of Siamese convolutional neural network (CNN) architecture and a few-shot classifier based on cosine similarity. The findings indicate superior accuracy results for the cosine similarity-based classifier.

In summary, prior research on sEMG-based gesture recognition spans a wide range of applications and methods, including traditional machine learning with handcrafted features and deep learning models such as CNNs, LSTMs and Siamese networks. Key challenges addressed in the literature include inter-subject variability, data scarcity and real-time performance, with solutions often involving transfer learning and fine-tuning strategies. These studies form a strong foundation for our work and offer relevant benchmarks for comparison.

3 Datasets in the literature and current study

Electromyography (EMG) entails the capture of electric signals produced during muscle contractions that constitute a rich source of information. The EMG signal represents the collective motor unit action potential (MUAP) trains originating from distinct active motor units within the electrode's range. The motor unit is the most fundamental entity within the neuromuscular system [43].

The power spectral density function of the sEMG signals has negligible contributions outside the range 5 to 500 Hz [44]. The segmentation of EMG signals into a sequence of envelopes is typically accomplished through the adoption of the sliding window method.

Allard et al. [27] created a dataset with a simple sensor layout (8 sensor Myo Armband) for transfer learning. The dataset, having 7 gestures, is composed of two parts: training dataset and evaluation dataset. The training dataset was formed by data from 18 able-bodied subjects while the evaluation dataset was formed by data from 17 able-bodied subjects.

In our experiments, we have used the dataset constructed by Allard et al. [27]. The training dataset has been used for pretraining, and the evaluation dataset has been used for fine-tuning and testing. In the data acquisition phase, the continuous recording of each gesture for 5 s is termed a cycle. Four cycles form a round. With this nomenclature, the training dataset consists of one round while the evaluation dataset consists of three rounds (one round for fine-tuning and two rounds for testing).

Another dataset creation work was performed in [45]. The 9 classes defined in this dataset correspond to the opening and grasping of fingers, flexion and extension of the wrist, pronation and supination of the wrist, radial and ulnar flexion of the wrist and relaxation. The dataset has been gathered from 10 normal subjects by using Myo Armband.

Hartwell, Kadirkamanathan and Anderson [46] gathered data from 10 able-bodied subjects by Myo Armband (8 surface electrodes) and 5 Delsys Trigno [47] wireless surface electrodes. In the dataset acquisition phase, each subject has performed 14 gestures and one rest position. Each stationary gesture was recorded for 10 s.

The Ninapro DB5 dataset, as described in [48], is one of the most comprehensive publicly available datasets for hand gesture recognition using sEMG signals. It comprises recordings from 10 subjects performing three different exercise protocols. Exercise A includes 12 basic finger movements, Exercise B comprises 8 hand postures and 9 basic wrist movements, and Exercise C consists of 23 functional and grasping movements. In total, the dataset encompasses 53 distinct classes (52 movement/posture classes plus one neutral class). The placement of sEMG sensors is determined based on the anatomical locations of relevant muscle groups to ensure accurate signal acquisition. By following the experimental setup outlined in [30], Ninapro DB5 Exercise B is also employed in our study as one of the benchmark datasets to assess the performance and generalizability of the proposed method.

4 Method

The central theme of our approach is the learning of good feature vectors from the sEMG data for gesture classification by feature matching. The feature vectors are obtained by means of a CNN network of Conv2D and fully connected layers with output classification layer removed following network design. The CNN network, pretrained on a universal training set of a large number of subjects, is optimized to reduce overfitting and adapt to the data of the specific subject using the system by means of successive freezing of layers during a multiple-pass fine-tuning process. Feature vector matching is facilitated by learning the similarity of feature vectors through SVM design on training data. An SVM-based similarity predictor has been preferred since SVM is known to be effective on small size data of the subject. At inference time, feature vectors of the training data that are closest to the feature vector of the subject's gesture are determined by predicting similarities by SVM. Finally, the effect of

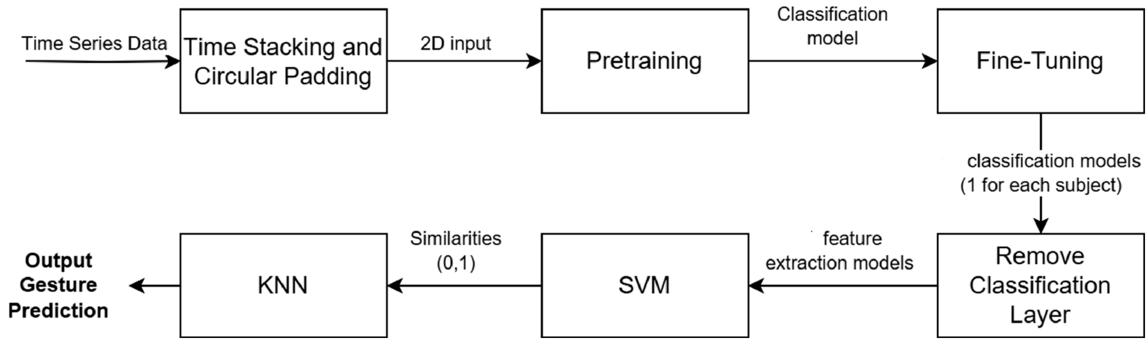


Fig. 1 Block diagram of the proposed gesture classification system.

outlier matches, which typically occur around class boundaries of the training feature vectors, is largely reduced by kNN voting.

A block diagram of the proposed gesture classification system is shown in Fig. 1. In the following, we first outline the steps of the proposed approach for designing and applying the proposed classification system and then elaborate on each step:

1. Data formatting: Circular padding and time stacking
   Output: 2d data
2. Pretraining: Training the CNN classifier with pretraining dataset
   Output: A trained CNN classifier
3. Fine-Tuning: Multi-pass fine-tuning of the CNN classifier with personalized fine-tuning dataset
   Output: One fine-tuned CNN classifier for each subject
4. Removing the top layer: Removing the classification layer of each CNN classifier
   Output: One CNN feature extractor for each subject
5. Learning similarity with SVM: Training SVM regressor to learn similarity (label 0 is used to indicate two samples of different classes, and label 1 is used to indicate two samples of the same class)
   Output: One SVM regression model for each subject which predicts the similarity value between any two samples of that subject
6. KNN-based classification: Determining the training samples closest to the test sample (distance=1-similarity) and aggregating their labels in a KNN setting to predict the class of the test sample
   Output: The predicted class label for the test sample

Step 1 is common to both the design of the proposed system and its application to unlabeled gesture signal (test sample). The CNN model designed in steps 2, 3 and 4 is applied to the formatted labeled and unlabeled gesture signals to yield their feature vectors. The similarities of the feature vector of the unlabeled gesture signal to the labeled gesture signals are measured to yield the class label in a majority voting kNN setting.

Since the data is in the form of a consecutive time series, exploiting the temporal dependencies has great importance. The data coming from the sensors at N consecutive sampling instants can be arranged as a 2D array

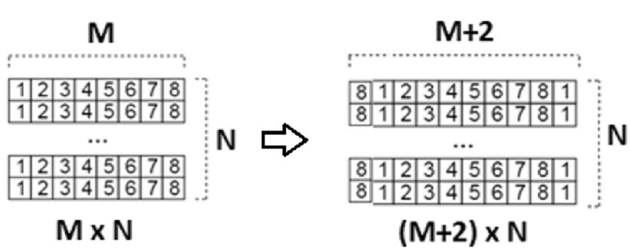


Fig. 2 Left: Stack of sensor array data in a time window of size N (size MxN), Right: Time stack after padding with circular extension (size (M+2)xN). M sensors horizontally arranged and N consecutive time instants vertically arranged.

of size M x N. Padding is applied to the left of the leftmost sensor in the array and right of the rightmost sensor in the array by performing the circular extension depicted in Fig. 2. Padding is necessary for the support of the CNN kernels. Circular extension is valid since the rightmost and leftmost sensors are neighbors of each other on the human arm. After windowing and padding, we obtain a dataset that consists of 2D arrays of data samples of (M+2) x N size. We adopt M=8 and N=52 from [30].

In our method, the initial step involves training a CNN model using the training dataset. It comprises two convolution layers, each containing 48 nodes with a 3x3 kernel size, and four fully connected layers consisting of 256, 128, 16 and 7 nodes for the Myo Dataset. The final layer, with 7 nodes, serves as the classification layer in the CNN model. For the NinaPro DB5 Exercise B dataset, the last two fully connected layers are configured with 32 and 18 nodes, respectively, to accommodate its 18-class classification task. In each layer, a dropout probability of 0.5 is applied, except for the classification layer. This initial CNN model possesses universal applicability across all subjects, as it is trained on the training data from multiple subjects.

A small part of the evaluation dataset has been reserved as fine-tuning data for each subject. Since the learning of a large number of parameters typically requires a large size of training data, we needed to formulate a fine-tuning scheme. We developed a fine-tuning framework called "multi-pass," where we define a "pass" as the complete training of the trainable parameters within the model with all of the fine-tuning data. The first fine-tuning pass trains the top-n (entire network) layers, the second pass trains the top (n-1) layers (with the lowest layer frozen), the third pass trains the top (n-2) layers (with the lowest two layers frozen), etc. until the final pass where only the top layer is trained and all other layers are frozen. These steps are presented in Fig. 3.

The above-described procedure generates a distinct CNN classification model tailored to each individual subject.

In order to learn the similarity metric between samples, the fine-tuned model, described thus far, is used as a feature extractor. Removing the top (classification) layer from the model results in a feature extractor which yields as many features as there are nodes in the last layer.

The feature vectors of training samples extracted by the CNN feature extractor are used to train a SVM regressor which learns the similarities between samples as depicted in Fig. 4. SVM regression model inputs k-dimensional feature vectors of sample 1 and sample 2 and predicts the similarity value between these two samples. A value of 1 indicates that the two samples are from the same class with a high likelihood, and a value of 0 indicates that the two samples are from distinct classes with a high likelihood. During the training of the regressor, pairs of random samples from the dataset are used.

In SVM implementation, we employ ϵ-SVR ([49]). The regression formula is

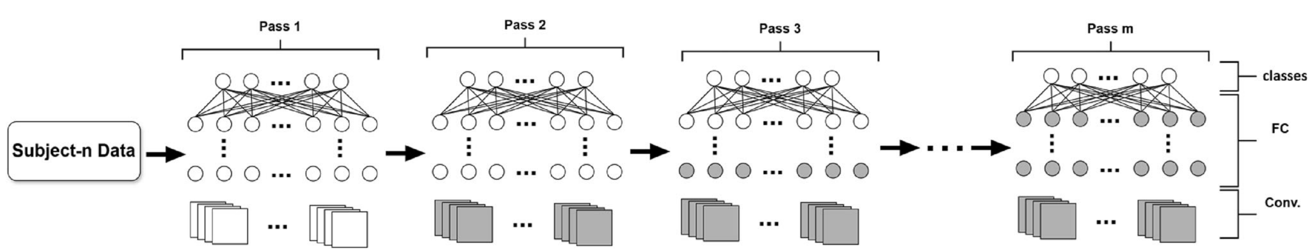
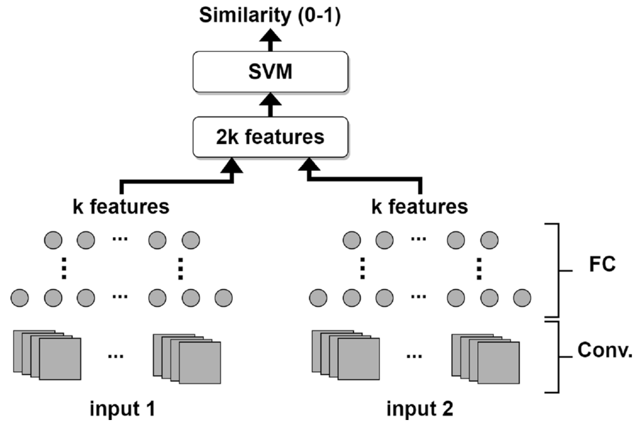


Fig. 3 Multi-pass fine-tuning scheme is presented for a subject. Each stage of the figure depicts the state of the model after successive fine-tuning iterations. Model components are represented using geometric symbols: Squares denote convolutional layers, while circles indicate fully connected layers. Node colors conveys parameter update status, white nodes correspond to trainable layers with unfrozen weights, and gray nodes indicate frozen layers with fixed weights. Progressing from left to right, each fine-tuning iteration freezes an additional layer, thereby transferring previously learned parameters and reducing the number of trainable weights.

Fig. 4 Similarity learning.



$$y = \sum_{i=1}^{l} (-\alpha_i + \alpha_i^*) K(x_i, x) + b, \tag{1}$$

where y is the output and $x_i \in R^n$ is a feature vector. We employ Radial basis function (RBF) $exp(-\gamma \|x - x'\|^2)$ as kernel where γ is the parameter that determines the support of the kernel. In the SVM model, we set regularization parameter as C=1.0, errorless distance to hyperplane as ϵ=0.1, and $\gamma = 1/((\#features)(\sigma^2))$.

The preceding steps provide the feature extraction method based on CNN and a SVM model for predicting the similarities between samples. The learned similarity metric is used in KNN classification in order to predict the class of an input sample. The KNN model uses a distance metric $d(x_i, x_j)$ for any two inputs x_i and x_j. The SVM model, produces the similarity measure $s(x_i, x_j)$ for x_i and x_j such that $d(x_i, x_j) = 1 - s(x_i, x_j)$. For the KNN, we have set k=15 neighbors and assign uniform weights to samples within the neighborhood of the test sample for voting its class.

5 Experiments

In our experiments, we have first fine-tuned the model with different sized datasets and have examined the inference performance on the test dataset [27]. We repeat the experiments 10 times and get the average accuracy for each number of cycles used for fine-tuning.

In order to inspect the effects of dataset size on the compared models, comparisons were done with datasets of different sizes. Since the fine-tuning dataset has 4 cycles (each cycle includes one continuous recording of 35 s for each gesture), we compare accuracies of the models fine-tuned with 1 cycle, 2 cycles, 3 cycles and 4 cycles of the fine-tuning dataset. In these experiments, fine-tuning with 1 cycle represents the smallest data available for fine-tuning.

5.1 Component-level ablation analysis

The proposed method comprises the steps of training CNN with general dataset, multi-pass fine-tuning with personalized dataset [27], learning similarities between samples with SVM and applying KNN as a classifier. To assess the individual contributions of each step, we record the accuracies achieved in 7-class hand gesture classification after the addition of that step. In the following, we describe each such classifier.

We train the proposed CNN with the training dataset of multiple subjects. This CNN model is then fine-tuned with the fine-tuning dataset. In our approach, we set the maximum number of epochs and early stopping criteria based on rate of change in the validation set error. As a result, 7-class CNN classification models are readily

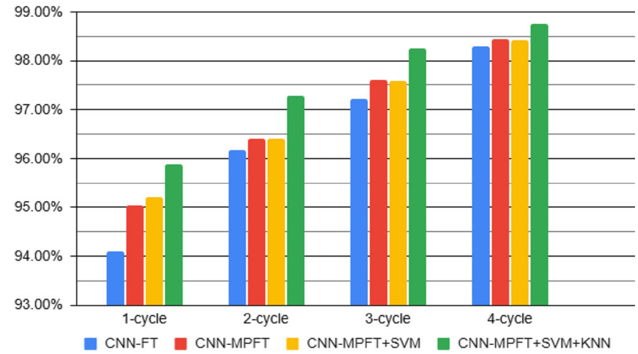


Fig. 5 Accuracies with varying dataset sizes for tested system configurations (increasing complexity from left to right).

generated for each subject that appear in the evaluation dataset. The accuracies of these models are presented as "CNN FT" in Fig. 5. This approach attains classification accuracies ranging from 94.10% to 98.29% across 1–4 cycles.

The above-described CNN model is then fine-tuned with multi-pass fine-tuning and the results are presented as "CNN MPFT" in Fig. 5. This method achieves classification accuracies in the range of 95.05% to 98.44% for 1–4 cycles. Employing the multi-pass approach exhibits a consistent improvement in accuracy across all cycle counts, compared to the "CNN FT" method.

We have used the truncated "CNN MPFT" as a feature extractor. The extracted features are classified with SVM and the results are presented as "CNN MPFT + SVM" in Fig. 5. In SVM implementation, we employ ϵ-SVR ([49]) implemented in Python scikit-learn library. Our proposed method applies SVM regression in the 0–1 interval, whereas we analyzed the 7-class SVM classification performance without employing similarity learning in this intermediate model. Experiments indicate slight improvement, 0.31% on average, compared to the "CNN MPFT" method.

For KNN, we adopt k=15 neighbors and uniform weights which means nearest k points (samples) are weighted equally. The final (proposed) model accuracies are shown as "CNN MPFT + SVM similarity + KNN" in Fig. 5. Our experiments show that learning similarity further improves the model accuracies 0.67% on average.

Our proposed method, "CNN MPFT + SVM similarity + KNN," achieve 95.87% and 98.77% accuracies for 1-cycle and 4-cycle fine-tuning dataset. In summary, as presented in Fig. 5, each of the additional components in the proposed model improves performances for all fine-tuning dataset sizes, albeit the gains are the largest for the smallest dataset sizes

5.2 Evaluation of fine-tuning strategies

We evaluated three fine-tuning strategies: the single-pass fine-tuned model (without progressive layer freezing), the multi-pass fine-tuned model (with progressive layer freezing) and a selectively fine-tuned model based on a greedy subtuning approach [36]. The evaluation was conducted using Myo Dataset [27] to assess the performance of the proposed method. Among these, the multi-pass fine-tuned model yielded the highest average accuracy of 98.77%, surpassing both the single-pass (98.62%) and the selectively fine-tuned subtuning model (98.46%).

5.3 Benchmarking on the Myo Dataset

We compared the classification accuracy of our method against slow fusion CNN [27], progressive NN [27] and transfer learning with a CWT (continuous wavelet transform) on the same dataset [30]. Comparison results are given in Fig. 6.

CWT+TL has come second to our proposed method in accuracy. For 1-cycle experiments, CWT+TL method achieves 94,69% accuracy and our method achieves 95.87% accuracy. This improvement means that the errors are

 https://doi.org/10.1007/s00521-025-11393-1

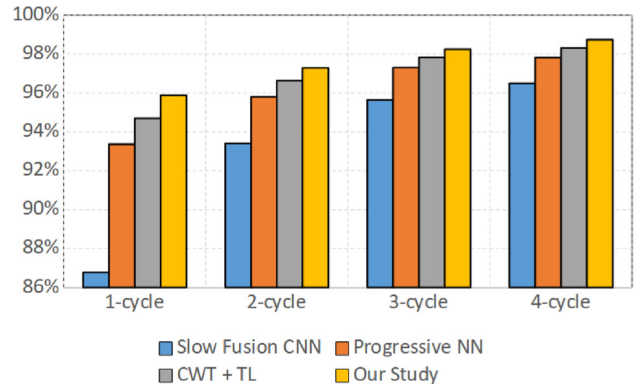


Fig. 6 Accuracy comparisons of leading previous methods ([27, 30]) and the proposed method parameterized by the number of cycles used for fine-tuning.

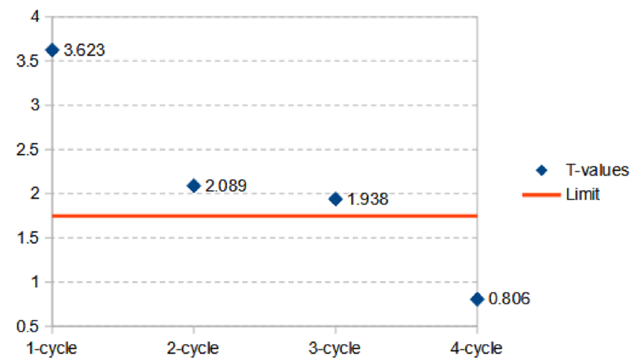


Fig. 7 Calculated t-values and threshold t-value for one tail t-rest ($p < 0.05$).

reduced by 22.2%. Our proposed approach also reduces errors by 19.9%, 20.2% and 27.2% compared to CWT+TL ([30]) for the 2-cycle, 3-cycle and 4-cycle fine-tuning cases, respectively.

A one-tailed t-test was conducted to statistically compare the performance of the CWT+TL approach with the proposed algorithm. The use of a one-tailed test is justified by our objective to demonstrate that the proposed method achieves a higher mean accuracy than the reference method with the highest mean accuracy. Accordingly, the null hypothesis assumes that the proposed method yields equal or lower mean accuracy. The results indicate that the improvements in the 1-, 2- and 3-cycle experiments are statistically significant ($p < 0.05$). The resulting values are illustrated in Fig. 7. However, the improvement observed in the 4-cycle experiments is below the threshold for statistical significance.

In addition to the detailed experiments, we also provide supplementary results for alternative network architectures. A Siamese neural network with the same architecture as our model was evaluated, corresponding to the CNN+SVM configuration in the proposed method, and achieved an accuracy of 89.46%, compared to 98.29% obtained with the proposed CNN+SVM approach.

Additionally, the model was compared with an LSTM-based LCNN variant [25], which achieved an accuracy of 97.39%. This architecture comprises two LSTM layers, each followed by a one-dimensional convolutional layer, which are followed by the same fully connected layers as used in the proposed method. LCNN performance remained below the proposed system, which achieved a higher accuracy of 98.77%.

5.4 Benchmarking on NinaPro DB5

We adopted the experimental scenario for Ninapro DB5 Exercise B [48] defined in [30]. Specifically, we utilized data from a single Myo Armband comprising 8 sEMG channels, but did not use the IMU (inertial measurement unit) and second Myo Armband data. For evaluation, a leave-one-subject-out strategy was employed, in which each of the 10 subjects, in turn, was designated as the test subject, while the remaining nine subjects were used for

Table 2 Model accuracy comparisons on the Ninapro DB5 exercise B

Method	Accuracy (%)
Raw+TL [30]	68.98
CWT + EMGNet [50]	67.42
LSTM-CNN [25]	61.4
RCviT [51]	73.23
CNN-MP+SVM+KNN	**82.05**

The proposed method is highlighted in **bold**

pretraining. Under this setting, our method achieved an accuracy of 82.05% on the 18-class dataset, outperforming the other approaches that utilized similar experimental configuration. A comparison of our results with the existing literature is provided in Table 2.

5.5 Runtime evaluation and real-time compatibility

We evaluate the real-time feasibility of our method. According to the study by Allard et al. [30], a latency of 300 ms is considered acceptable for real-time applications. Consequently, sliding windows of 52 samples (260 ms) allocate 40 ms for signal processing. The experiments were conducted on an Intel Core i7-12650 H processor within a Windows Subsystem for Linux (WSL) environment, with system resources restricted to a single CPU core and 2 GB of RAM to simulate low-resource conditions typical in mobile applications like prosthetic control. The proposed method achieves a total processing time of 20.14 ms when the steps of padding, feature extraction via CNN and classification using kNN with SVM-based similarity are executed. This runtime is well within the 40-millisecond threshold for signal processing and classification in real-time applications.

As a step toward a real-time implementation of the proposed system, we have carried out a real-time implementation of the proposed CNN network (consisting of Conv2D and Dense layers) in our study using the TensorFlow–TensorRT integration. Specifically, we have converted our CNN model to 32-bit floating precision (FP32), 16-bit floating precision (FP16) and 8-bit integer precision (INT8) models by using the TF-TRT API. TF-TRT succeeded in converting 93% of the operations in the CNN model. During inference, both the 32-bit floating precision model and the 16-bit floating precision model yielded the same accuracy as the original model whereas the 8-bit integer precision model yielded less than 0.01% loss in average accuracy. On the other hand, we have observed 45.9%, 50.4% and 52.6% reduction in runtimes with the 32-bit floating precision, 16-bit floating precision and 8-bit integer precision models, respectively.

6 Conclusion

We proposed a transfer learning-based model and achieved high accuracies with small person-specific data. In the proposed model, pretraining increases generalization ability while fine-tuning increases data fitting ability to personal data. In other words, the proposed model helps one to detect gestures from a subject by expending low effort for recording sample data.

Our proposed method includes learning a feature extraction model from general (pretraining) and personal (evaluation) dataset, learning similarity metric from personalized (fine-tuning) dataset and classification using the KNN method. The method is based on the hypothesis that acquiring knowledge about the similarity between samples yields more favorable outcomes compared to transfer learning of a classification model. An ablation study was conducted to evaluate the contribution of each component of the proposed method, demonstrating a progressive improvement in classification accuracy.

We have showed the performance efficacy of the proposed method over the methods such as CWT+TL, progressive NN, slow fusion CNN and LCNN as well as the approach based on the full fine-tuning, and greedy selective layer fine-tuning of the proposed CNN model.

For benchmarking purposes, two datasets were utilized, and in both cases, the proposed method outperformed existing approaches reported in the literature. The proposed model achieves an average accuracy ranging from 95.87 to 98.77% when evaluated on a 7-class dataset of varying sizes across 17 test subjects. Notably, our method outperforms state-of-the-art approaches documented in the literature, exhibiting superior accuracy across all tested fine-tuning dataset sizes (cycles). Furthermore, statistical analysis using t-tests confirmed that the improvements in accuracy for 1-, 2- and 3-cycle evaluations are statistically significant. The NinaPro DB5 experiment demonstrates that our method outperforms existing studies in the literature with a similar experimental setup, achieving an accuracy of 82.05% on a 18-class dataset involving 10 subjects.

The real-time feasibility of the proposed method was also examined, and the measured runtimes were found to be within acceptable limits for real-time applications.

7 Future works

In the future, the primary focus will be real-time implementation and increased robustness. The current study primarily focuses on offline analysis of EMG data. Future research could extend the proposed method by allowing for the accurate identification of hand gestures in real-world scenarios. This could involve addressing challenges such as low-latency processing and real-time classification of EMG signals.

Hand gestures that are most often confused with each other due to the similarity of their sEMG signals could be better distinguished from each other by the additional use of IR cameras or inertial sensors that enable optical hand tracking and motion sensing [52].

The experiments were run on a Windows (WSL—Windows Subsystem for Linux)-based PC platform. TensorRT integration to TensorFlow showed ignorable performance loss for reduced complexity (near real time) implementation. However, the actual platform controlling a device like a prosthesis might have to be embedded if the inference module needs to be completely portable and reside near the arm (when close proximity to a PC environment with fast wireless connectivity limits applicability of the method). Future work needs to address such a realistic implementation scenario.

Author Contributions Semih Celik was responsible for methodology, software, experiments, writing original draft, reviewing and editing. Ulug Bayazit was involved in methodology, experiments, supervision, writing original draft, reviewing and editing.

Funding Open access funding provided by the Scientific and Technological Research Council of Türkiye (TÜBİTAK). Not applicable

Data availability Data sharing is not applicable to this article, as no datasets were generated in the current study. For details on the experimental dataset, please refer to Sect. 3

Code availability The code can be provided by the corresponding author upon request.

Declarations

Conflict of interest The authors have no conflict of interest to declare that are relevant to the content of this article.

Ethics approval and consent to participate Not applicable

Consent for publication Not applicable

Materials availability Not applicable

References

1. Arvetti M, Gini G, Folgheraiter M (2007) Classification of EMG signals through wavelet analysis and neural networks for controlling an active hand prosthesis. In: 2007 IEEE 10th international conference on rehabilitation robotics, pp 531–536
2. Atzori M, Cognolato M, Muller H (2016) Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands. Front Neurorobot 10:9
3. Crawford B, Miller K, Shenoy P, Rao R (2005) Real-time classification of electromyographic signals for robotic control. 2:523–528
4. Côté Allard U, Nougarou F, Fall CL, Giguère P, Gosselin C, Laviolette F, Gosselin B (2016) A convolutional neural network for robotic arm guidance using sEMG based frequency-features, p 2464
5. Fu J, Choudhury R, Hosseini SM, Simpson R, Park J-H (2022) Myoelectric control systems for upper limb wearable robotic exoskeletons and exosuits - a systematic review. Sensors 22(21):8134
6. Asghari Oskoei M, Hu H (2007) Myoelectric control systems - a survey. Biomed Signal Process Control 2(4):275–294
7. Xiong D, Zhang D, Zhao X, Zhao Y (2021) Deep learning for EMG-based human-machine interaction: a review. IEEE/CAA J Autom Sin 8(3):512–533
8. Gohel V, Mehendale N (2020) Review on electromyography signal acquisition and processing. Biophys Rev 12(6):361–1367
9. Dubey R, Kumar M, Upadhyay A, Pachori RB (2022) Automated diagnosis of muscle diseases from EMG signals using empirical mode decomposition based method. Biomed Signal Process Control 71(A):103098
10. Li Q, Liu Y, Zhu J, Chen Z, Liu L, Yang S, Li J, Jin R, Tao J, Chen L (2023) A motion recognition model for upper-limb rehabilitation exercises. J Ambient Intell Humaniz Comput 13:16101
11. Chen P, Wang H, Yan H, Du J, Ning Y, Wei J (2023) sEMG-based upper limb motion recognition using improved sparrow search algorithm. Appl Intell 53(7):7677–7696
12. Barreto AB, Scargle SD, Adjouadi M (2000) A practical EMG-based human-computer interface for users with motor disabilities. J Rehabil Res Dev 37(1):53–63
13. Kim J-S, Kim M-G, Pan S-B (2021) Two-step biometrics using electromyogram signal based on convolutional neural network-long short-term memory networks. Appl Sci 11(15):6824
14. Fan B, Liu X, Su X, Hui P, Niu J (2020) Emgauth: an emg-based smartphone unlocking system using siamese network. In: IEEE International Conference on Pervasive Computing and Communications (PerCom), pp 1–10
15. Mahmoud AO, Ziedan I, Zamel AA (2023) An efficient convolutional neural network classification model for several sign language alphabets. Int J Adv Comput Sci Appl (IJACSA) **14**(9)
16. Ben Haj Amor A, El Ghoul O, Jemni M (2023) Sign language recognition using the electromyographic signal: a systematic literature review. Sensors 23(19):8343
17. Allami Sanjani M, Tahami E, Veisi G (2023) A review of surface electromyography applications for the jaw muscles characterization in rehabilitation and disorders diagnosis. Med Nov Technol Devices 20:100261. https://doi.org/10.1016/j.medntd.2023.100261
18. Kocyigit Y, Korurek M (1999) EMG signal classification using wavelet transform and fuzzy clustering algorithms. IEEE Electron Device Lett 20:569–571
19. Shroffe D, Manimegalai EH (2013) Hand gesture recognition based on EMG signals using ANN. IEEE Electron Device Lett 2(3):31–39
20. Karabulut D, Ortes F, Arslan YZ, Adli MA (2017) Comparative evaluation of emg signal features for myoelectric controlled human arm prosthetics. Biocybern Biomed Eng 37(2):326–335
21. Narayan Y (2021) Semg signal classification using knn classifier with fd and tfd features. Materials Today: Proceedings 37, 3219–3225. International conference on newer trends and innovation in mechanical engineering: materials science
22. Sun W, Liu H, Tang R, Lang Y, He J, Huang Q (2019) sEMG-based hand-gesture classification using a generative flow model. Sensors. 19(8):1952

23. Vasiliev AV, Melnikov AO (2022) Application lstm neural networks for biological signal classification. In: 2022 IEEE International Multi-Conference on Engineering, Computer and Information Sciences (SIBIRCON), pp 750–755
24. Qi W, Ovur SE, Li Z, Marzullo A, Song R (2021) Multi-sensor guided hand gesture recognition for a teleoperated robot using a recurrent neural network. IEEE Robot Autom Lett 6:6039–6045. https://doi.org/10.1109/LRA.2021.3089999
25. Wu Y, Zheng B, Zhao Y (2018) Dynamic gesture recognition based on lstm-cnn. In: 2018 Chinese Automation Congress (CAC), pp 2446–2450. https://doi.org/10.1109/CAC.2018.8623035
26. Wu D, Yang J, Sawan M (2022) Transfer learning on electromyography (emg) tasks: approaches and beyond. IEEE Trans Neural Syst Rehabil Eng 31:3015–3034
27. Allard UC, Fall CL, Campeau-Lecours A, Gosselin C, Laviolette F, Gosselin B (2017) Transfer learning for sEMG hand gestures recognition using convolutional neural networks. IEEE International Conference on Systems, Man, and Cybernetics (SMC), 1663–1668
28. Lehmler SJ, Saif-ur-Rehman M, Glasmachers T, Iossifidis I (2021) Deep transfer-learning for patient specific model re-calibration: application to semg-classification. ArXiv arXiv:2112.15019
29. Kim K-T, Guan C, Lee S-W (2020) A subject-transfer framework based on single-trial emg analysis using convolutional neural networks. IEEE Trans Neural Syst Rehabil Eng 28(1):94–103
30. Côté-Allard U, Fall CL, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, Laviolette F, Gosselin B (2019) Deep learning for electromyographic hand gesture signal classification using transfer learning. IEEE Trans Neural Syst Rehabil Eng 27(4):760–771
31. Ameri A, Akhaee MA, Scheme E, Englehart K (2020) A deep transfer learning approach to reducing the effect of electrode shift in emg pattern recognition-based control. IEEE Trans Neural Syst Rehabil Eng 28(2):370–379
32. Lialin V, Deshpande V, Rumshisky A (2023) Scaling down to scale up: a guide to parameter-efficient fine-tuning. ArXiv arXiv:2303.15647
33. Liu H, Tam D, Muqeeth M, Mohta J, Huang T, Bansal M, Raffel C (2022) Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. ArXiv arXiv:2205.05638
34. Liao B, Meng Y, Monz C (2023) Parameter-efficient fine-tuning without introducing new latency. In: Proceedings of the 61st Annual meeting of the association for computational linguistics (Volume 1: Long Papers), pp 4242–4260
35. Jeong J-C, Yu G-H, Song M-G, Vu D, Anh L, Jung Y-A, Choi Y-A, Um T-W, Kim J-Y (2022) Selective layer tuning and performance study of pre-trained models using genetic algorithm. Electronics 11:2985
36. Kaplun G, Gurevich A, Swisa T, David M, Shalev-Shwartz S, Malach E (2023) Less is more: selective layer finetuning with subtuning (2023) https://api.semanticscholar.org/CorpusID:259313781
37. Ma X, Wang L, Yang N, Wei F, Lin J (2023) Fine-tuning llama for multi-stage text retrieval. ArXiv arXiv:2310.08319
38. Wang L, Li Y, Miller T, Bethard S, Savova G (2023) Two-stage fine-tuning for improved bias and variance for large pretrained language models. In: Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp 15746–15761. https://doi.org/10.18653/v1/2023.acl-long.877 . https://aclanthology.org/2023.acl-long.877
39. ValizadehAslani T, Shi Y, Wang J, Ren P, Zhang Y, Hu M, Zhao L, Liang H (2022) Two-stage fine-tuning: a novel strategy for learning class-imbalanced data. ArXiv arXiv:2207.10858
40. Sarhan N, Lauri M, Frintrop S (2022) Multi-phase fine-tuning: a new fine-tuning approach for sign language recognition. KI - Künstliche Intelligenz 36(1):91–98. https://doi.org/10.1007/s13218-021-00746-2
41. Calhas D, Romero E, Henriques R (2019) On the use of pairwise distance learning for brain signal classification with limited observations. NeurIPS 105:101852
42. Tam S, Boukadoum M, Campeau-Lecours A, Gosselin B (2022) Siamese convolutional neural network and few-shot learning for embedded gesture recognition. In: 2022 20th IEEE Interregional NEWCAS Conference (NEWCAS), pp 114–118
43. Chen M, Zhou P (2016) A novel framework based on FastICA for high density surface emg decomposition. IEEE Trans Neural Syst Rehabil Eng 24(1):117–127
44. Marletti R (1999) Standards for reporting EMG data. J Elektromyoraphy Kinesiol (JEK) 9(1):3–4
45. Raurale S, McAllister J, Rincon JM (2019) EMG wrist-hand motion recognition system for real-time embedded platform. In: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp 1523–1527
46. Hartwell A, Kadirkamanathan V, Anderson SR (2018) Compact deep neural networks for computationally efficient gesture classification from electromyography signals. In: 2018 7th IEEE International Conference on Biomedical Robotics and Biomechatronics (Biorob), pp 891–896
47. Delsys: Delsys trigno wireless system. Available (2018): http://www.delsys.com/products/wireless-emg/trigno-lab/
48. Atzori M, Gijsberts A, Heynen S, Hager A-GM, Deriaz O, Smagt P, Castellini C, Caputo B, Müller H (2012) Building the ninapro database: a resource for the biorobotics community. In: 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob), pp 1258–1265
49. Chang C-C, Lin C-J (2011) LIBSVM: a library for support vector machines. ACM Trans Intell Syst Technol 2(3):1–27
50. Chen L, Fu J, Wu Y, Li H, Zheng B (2020) Hand gesture recognition using compact cnn via surface electromyography signals. Sensors 20(3):672. https://doi.org/10.1109/TIM.2021.3063752

51. Shen S, Wang X, Mao F, Sun L, Gu M (2022) Movements classification through semg with convolutional vision transformer and stacking ensemble learning. IEEE Sens J 22(13):13318–13325. https://doi.org/10.1109/JSEN.2022.3179535
52. Ovur SE, Su H, Qi W, De Momi E, Ferrigno G (2021) Novel adaptive sensor fusion methodology for hand pose estimation with multileap motion. IEEE Trans Instrum Meas 70:1–8. https://doi.org/10.1109/TIM.2021.3063752

Authors and Affiliations

Semih Celik[1] · **Ulug Bayazit**[1]

✉ Semih Celik
semcelik@itu.edu.tr

[1] Istanbul Technical University: Istanbul Teknik Universitesi, Istanbul, Türkiye