RESEARCH ARTICLE

A novel concatenate feature fusion RCNN architecture for sEMG-based hand gesture recognition

Pufan Xu[1], **Fei Li**[2]*, **Haipeng Wang**[3]*

1 School of Electronic Science and Engineering, Southeast University, Nanjing, Jiangsu, China, **2** Institute of RF- & OE-ICs, Southeast University, Nanjing, Jiangsu, China, **3** School of Electronic and Information Engineering, Sanjiang University, Nanjing, Jiangsu, China

* lifei@seu.edu.cn (FL); wanghaipeng4213@126.com (HW)

Abstract

Hand gesture recognition tasks based on surface electromyography (sEMG) are vital in human-computer interaction, speech detection, robot control, and rehabilitation applications. However, existing models, whether traditional machine learnings (ML) or other state-of-the-arts, are limited in the number of movements. Targeting a large number of gesture classes, more data features such as temporal information should be persisted as much as possible. In the field of sEMG-based recognitions, the recurrent convolutional neural network (RCNN) is an advanced method due to the sequential characteristic of sEMG signals. However, the invariance of the pooling layer damages important temporal information. In the all convolutional neural network (ACNN), because of the feature-mixing convolution operation, a same output can be received from completely different inputs. This paper proposes a concatenate feature fusion (CFF) strategy and a novel concatenate feature fusion recurrent convolutional neural network (CFF-RCNN). In CFF-RCNN, a max-pooling layer and a 2-stride convolutional layer are concatenated together to replace the conventional simple dimensionality reduction layer. The featurewise pooling operation serves as a signal amplitude detector without using any parameter. The feature-mixing convolution operation calculates the contextual information. Complete evaluations are made on both the accuracy and convergence speed of the CFF-RCNN. Experiments are conducted using three sEMG benchmark databases named DB1, DB2 and DB4 from the NinaPro database. With more than 50 gestures, the classification accuracies of the CFF-RCNN are 88.87% on DB1, 99.51% on DB2, and 99.29% on DB4. These accuracies are the highest compared with reported accuracies of machine learnings and other state-of-the-art methods. To achieve accuracies of 86%, 99% and 98% for the RCNN, the training time are 2353.686 s, 816.173 s and 731.771 s, respectively. However, for the CFF-RCNN to reach the same accuracies, it needs only 1727.415 s, 542.245 s and 576.734 s, corresponding to a reduction of 26.61%, 33.56% and 21.19% in training time. We concluded that the CFF-RCNN is an improved method when classifying a large number of hand gestures. The CFF strategy significantly improved model performance with higher accuracy and faster convergence as compared to traditional RCNN.

OPEN ACCESS

Citation: Xu P, Li F, Wang H (2022) A novel concatenate feature fusion RCNN architecture for sEMG-based hand gesture recognition. PLoS ONE 17(1): e0262810. https://doi.org/10.1371/journal.pone.0262810

Editor: Robertas Damaševičius, Politechnika Slaska, POLAND

Received: November 16, 2021

Accepted: January 6, 2022

Published: January 20, 2022

Data Availability Statement: All EMG signal files are available from the Ninapro database (URL: http://ninapro.hevs.ch/node/7).

Funding: This work is supported by Natural Science Foundation of China (NSFC) (URL:http://www.nsfc.gov.cn/)(No. 61801262).

Competing interests: The authors have declared that no competing interests exist.

1 Introduction

Electromyography (EMG) measures bioelectric currents produced by motor units during muscle contraction [1]. Surface EMG (sEMG) detects the sum of the motor unit action potential (MUAP) over the skin [2]. Due to its noninvasive and low-cost characteristics [3], sEMG-based hand gesture recognition systems are widely used in human-computer interaction [4], speech detection [5], robot control [6], and rehabilitation studies, like prosthesis operation [7–9] and stroke rehabilitation [10].

A traditional method for sEMG-based recognition is machine learning, which in general is not inherently efficient or scalable enough to handle massive datasets [11]. For simple pattern recognition (PR) based on sEMG signals [12–15], methods including linear discriminate analysis (LDA), k-nearest neighbor (KNN), principal component analysis (PCA), and artificial neural network (ANN) are usually chosen. Because of the stochastic nature of biological signals [16], signal preprocessing and feature extraction are necessary steps when applying these algorithms [17]. Data preprocessing, such as filtering, may result in the loss of valid information [18]. Feature extraction in machine learning is time-consuming and error-prone as it requires specialization, which significantly increases chances of reduced classification accuracy [19].

Deep learning methods are capable of handling large data calculation and perform auto-feature extraction in PR tasks. A model based on You Only Look Once (YOLO) v3 and DarkNet-53 convolutional neural networks (CNN) was built for image-based gesture recognition without additional data processing [20]. In CNN models, feature extraction is replaced by interconnected convolutions in the hidden layers, improving classification accuracies [21–23].

To further consider temporal information and process sequential data, state-of-the-art deep learning structures in time-series classification are welcomed in biosignal processing. A 50-layer CNN based on residual networks (ResNet) was built to classify 7 different gestures based on EMG [3]. Three architectures were designed for electrocardiography (ECG) classification, which were InceptionTime, ResNet and XResNet [24]. A novel spatial–temporal transformer network was proposed in reference [25] to solve skeleton-based human activity recognition tasks. RNNs also achieved high accuracies in speech recognition, signal detection, and video classification [26–28]. Combining the advantages of CNNs and RNNs, RCNNs have shown good performances in object detection, video classification, and emotion recognition [29–31]. Their merits when dealing with sEMG signals have also been proven both in discrete-motion classification and continuous-motion estimation [32, 33].

However, existing hand gesture recognizers mostly recognize only a few movements

Classifying an average of more than 50 gesture classes led to a lower than 2% average chance level [34, 35]. By simply reducing the number of gestures, the classification accuracy exceeded 90% [36]. An LSTM-CNN (LCNN) achieved 98.14% of accuracy when classifying 5 hand gestures using a proposed MyoDataset, but only 71.66% using NinaPro DB5 Exercise A (12 movements) and 61.4% using NinaPro DB5 Exercise B (17 movements) [37]. This is likely due to the conventional RCNN structure, where the CNN part includes a 1-stride convolutional layer and a pooling layer, with the pooling layer playing a role in dimensionality reduction [38]. Max-pooling tosses information about the precise position of the entity within the region [39], resulting in significant degradation of sequential features [40]. Springenberg describes ACNN structure to replace the pooling layer with a normal convolution having a stride larger than 1 [38], expecting to gain more learnings of contextual information from the receptive field of the kernel. However, except the ability of information extraction, ACNN also has an information confusion characteristic. Because with the feature-mixing convolution operation, a same output can be received from completely different inputs. More details are explained in the Methods section. Therefore, it is important to develop an innovative method of dimensionality reduction to allow more classes in hand-gesture recognition tasks.

The main contributions of this work are twofold:

1. This paper introduces a new dimensionality reduction strategy, which is named the concatenate feature fusion (CFF) strategy. Under the CFF strategy, a max-pooling layer and a convolutional layer with a stride of 2 are concatenated together to replace the single max-pooling in CNN or the 2-stride convolution with in ACNN. The featurewise nature of the pooling operation maintains signal intensity, while the feature-mixing of the convolution operation obtains temporal information from the context.

2. This paper proposes a concatenate feature fusion recurrent convolutional neural network (CFF-RCNN) structure based on a traditional RCNN network, which consists of a 4-layer CNN and a long short-term memory network (LSTM). Classifications of more than 50 gestures are achieved with high accuracies and fast convergence using the CFF-RCNN structure. The classification accuracies of the CFF-RCNN on DB1, DB2 and DB4 three databases in the NinaPro are 88.87%, 99.51% and 99.29%, which are higher than the machine learnings, RCNN, and other state-of-the-art methods. When analyzing the convergence rate of the RCNN and CFF-RCNN, training times to reach the same classification accuracy are compared. To achieve accuracies of 86%, 99% and 98%, the training times are 2353.686 s, 816.173 s and 731.771 s for RCNN, while 1727.415 s, 542.245 s and 576.734 s for CFF-RCNN, corresponding to a reduction of 26.61%, 33.56% and 21.19%.

The organization of this paper is as follows. Section 2 discusses the novel CFF technique and the basic construction of CFF-RCNN. In Section 3, we explain the experiment methods to test the performance, accuracy and efficiency, and tolerance of CFF-RCNN. Results are presented and analyzed in Section 4. Finally, limitations are discussed and conclusions are drawn in Section 5.

2 Methods

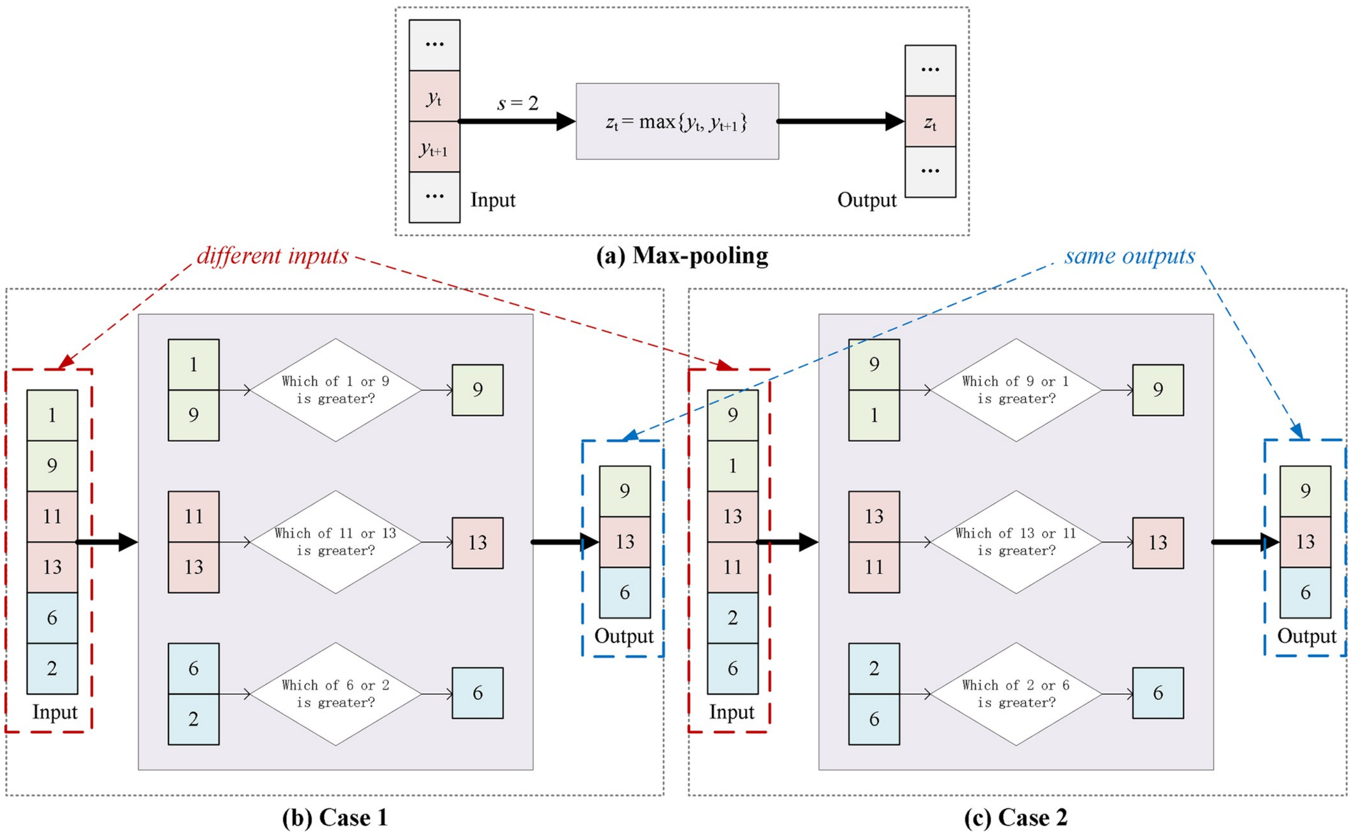
2.1 The theory of CFF strategy

The pooling operation can be viewed as a featurewise convolutional layer with a p-norm activation function [38]. It is a significant component in CNNs for three main reasons: 1) neighboring pooling units take input from patches that are shifted by more than one row or column, making the representation of movements and distortions more invariant; 2) the spatial dimensionality reduction reduces the computational cost for the subsequent conventional layers; and 3) the featurewise nature could make optimization easier than the feature-mixing nature of the convolution [38, 41].

The first reason plays an important role in image recognition tasks where the locations of objects have less important features [42]. Therefore, the invariance of the pooling allows representations to change very little when elements in the previous layer vary with position and appearance [41]. However, the invariance characteristic has been recently questioned. By extracting some entity from the context, the clutter and noise are removed, but the information of the background is also damaged [43]. Additionally, contextual information, corresponding to the temporal information when inputting a sequence signal, is extremely important during the recognition process. Therefore, the pooling layer may result in heavy losses of crucial data features.

To obtain a more quantized understanding of the information loss, the featurewise operation of 1D max-pooling can be described as follows:

$$z_t = \max\{y_t, y_{t+1}, \cdots, y_{t+s-1}\} \tag{1}$$



(a) Max-pooling
y_t
y_{t+1}
$s = 2$
$z_t = \max\{y_t, y_{t+1}\}$
z_t
Input
Output
different inputs
same outputs
Which of 1 or 9 is greater?
Which of 11 or 13 is greater?
Which of 6 or 2 is greater?
Which of 9 or 1 is greater?
Which of 13 or 11 is greater?
Which of 2 or 6 is greater?
(b) Case 1
(c) Case 2

Fig 1. The procedure of the max-pooling layer. (a) The featurewise operation will result in information losses, where y_t is the t-th element of the input, z_t is the output, s_p is the pooling size and s_s is the stride ($s_p = s_s = s = 2$). (b, c) Examples of the invariance characteristic. Case 1 and 2 are different in the order. The output of the max-pooling layer cannot reflect small input changes, losing important sequential information.

https://doi.org/10.1371/journal.pone.0262810.g001

where y_t is the t-th element of a row in the input map, z_t is the output map, s_p is the pooling size and s_s is the stride. Usually, the pooling size and stride are equal ($s_p = s_s = s$). Clearly, the max-pooling will throw away s-1 elements for each step, which is also shown in Fig 1A. Additionally, in Fig 1B (Case 1) and Fig 1C (Case 2), when only changing the order of the inputs, the output z_t remains unchanged. This is the so-called invariance characteristic, which will result in a serious loss in sequential information.

We symbolize the input data map of the i-th 1D convolutional layer as $X_{i\text{-}1}$ and the output map as X_i. The final output of the n-th convolutional layer is $Y = X_n$, and there are n layers of 1-stride convolution in total. The kernel sizes are unified to k and there is no padding. Considering a single convolution step as well, the output element $x_{i,t}$ can be calculated to k input elements by a convolutional function f_i:

$$x_{i,t} = f_i(x_{i-1,t}, x_{i-1,t+1}, \cdots, x_{i-1,t+k-1}) \tag{2}$$

$$\begin{aligned} y_t &= f_n(x_{n-1,t}, x_{n-1,t+1}, \cdots, x_{n-1,t+k-1}) \\ &= f_n\left(f_{n-1}(x_{n-2,t}, x_{n-2,t+1}, \cdots, x_{n-2,t+k-1}), \cdots, f_{n-1}(x_{n-2,t+k-1}, x_{n-2,t+k}, \cdots, x_{n-2,t+2\cdot(k-1)})\right) \\ &= F(x_{0,t}, x_{0,t+1}, \cdots, x_{0,t+n(k-1)}) \end{aligned} \tag{3}$$

Then, y_t is related to $n(k\text{-}1)\text{+}1$ input elements. Therefore, after a max-pooling layer with a size of s, there will be a total loss of s-1 pieces of sequential information between $n(k\text{-}1)\text{+}1$ input data, which violently affects the performances of the following networks.

To achieve dimensionality reduction without the use of pooling, an ACNN structure was introduced by Springenberg, with the pooling layer replaced by a normal convolution with a stride larger than 1 [38]. It is worth noting that this improvement was based on an important assumption that only the second reason mentioned above, which is dimensionality reduction, was crucial for achieving good CNN performance. However, in this paper, the existence of max-pooling is questioned mainly because it abandons crucial sequential information. By simply removing the max-pooling and adding the convolution with stride larger than 1, the receptive field of the kernel may be helpful for learning some contextual information, but the results are probably not that good, even causing misleading guidance. To explain this, let us consider a convolution with a size of k and stride of s. The single-step operation (without bias and before activating) can be described as follows:

$$y_t = \sum_{j=1}^{k} (c_j \times x_{s(t-1)+j}) \tag{4}$$

where x is the input feature, y is the output feature, and c is the element in the convolutional kernel. Therefore, y is a summation of k input features, and the weights are learned during the training procedure. With different inputs and weights, chances are that the output feature y may be the same:

$$y_t = \sum_{j=1}^{k} (c_j \times x_{s(t-1)+j}) = \sum_{j=1}^{k} (c'_j \times x'_{s(t-1)+j}) \tag{5}$$

An example is shown in Fig 2. Fig 2A shows that when the kernel size k is 3, each output element $x_{i,t}$ in the i-th layer is related to 3 inputs, $x_{i\text{-}1,t}$, $x_{i\text{-}1,t+1}$, $x_{i\text{-}1,t+2}$, in the (i-1)-th layer. In Fig 2B (Case 1) and Fig 2C (Case 3), both the convolutional kernels are set at (-1, 1, 2). Although the input data are completely different, same outputs are received. To take a more extreme example, assuming that all the input data x' are unified to a constant a and $c'_j = c_j \times x_{s\times(t-1)+j}/a$, the same output feature y is obtained. Therefore, y represents a time-alternating signal and a constant signal at the same time, resulting in confusion. The ability to extract the sequential information of the s-stride convolution mainly depends on the training effect, which is sensitive to the input data and increases computing loads.

The dual characteristics, which are information extraction and information confusion, are related to the feature mixing operation of the convolution. Although max-pooling fails to determine contextual features, it is an excellent featurewise amplitude detector without any training parameter. Knowing the most representative intensity of a sequence, the context information can be retained as much as possible. Therefore, a new dimensionality reduction strategy, which is called the CFF strategy, is proposed in this paper. The concatenate of a max-pooling layer and a convolutional layer with the stride of 2 is used to replace the traditional pooling layer in CNN, or the single 2-stride convolution in ACNN. As illustrated in Fig 3, both the max-pooling and the 2-stride convolution are applied to the inputs. Different outputs are received from Case 1 and 2 (which have the same result when only using max-pooling), and Case 1 and 3 (which have the same result when only using 2-stride convolution). After doing so, distinguishable features are obtained to represent different sequential data. By concatenating a max-pooling layer with a 2-stride convolutional layer, both featurewise and feature-mixing calculations are performed on the same data segment, extracting sequential information efficiently.

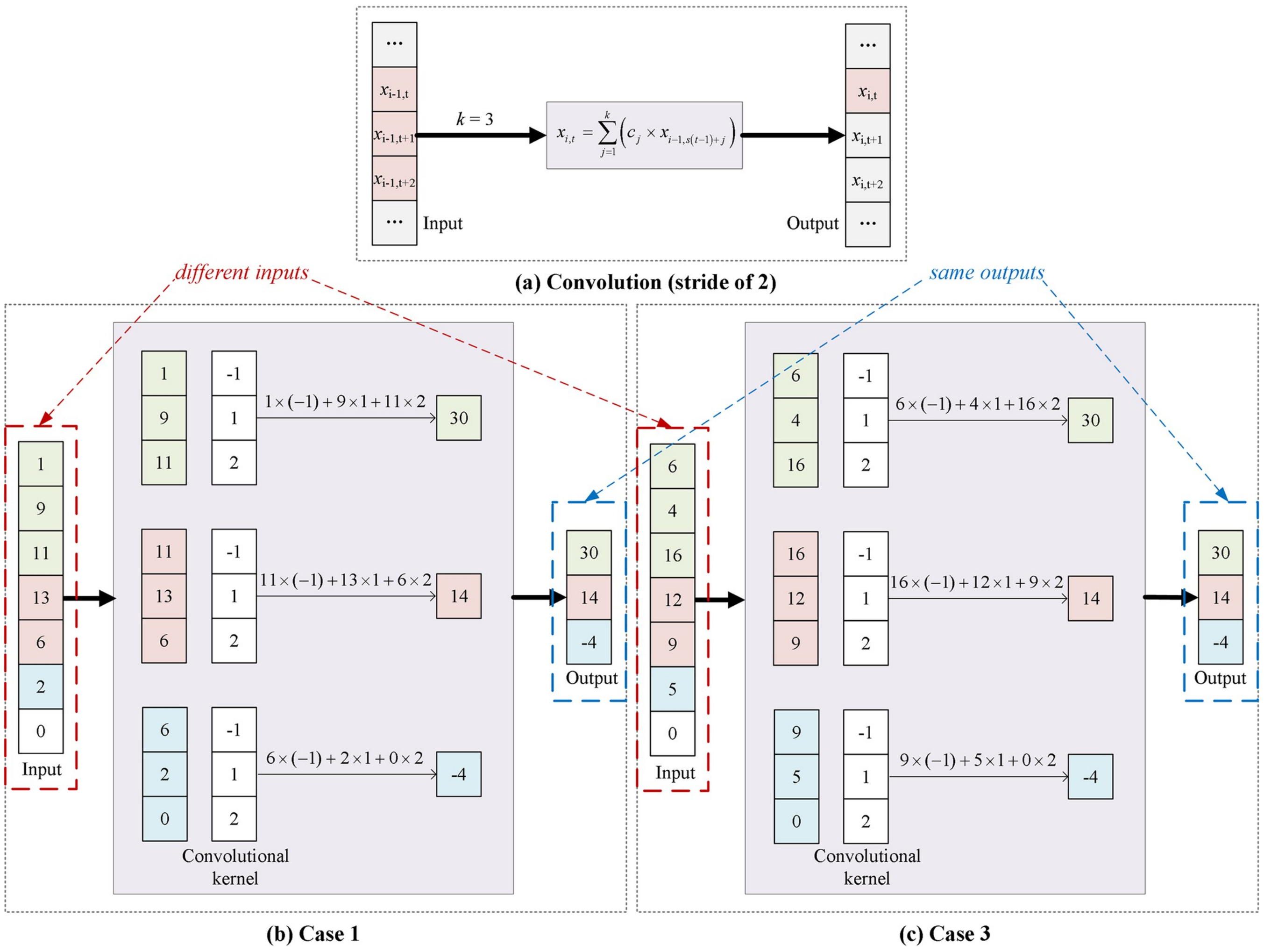


Fig 2. The procedure of the convolutional layer. (a) The feature-mixing operation of the convolution. *x* is the input feature, *y* is the output feature, and *c* is the element in the convolutional kernel. The kernel size is *k* (*k* = 3) and stride is *s*. (b, c) Examples of the information confusion caused by the feature mixing operation of the convolution. With the same convolutional kernels, same outputs can be received even though the input data are completely different in Case 1 and 3.

https://doi.org/10.1371/journal.pone.0262810.g002

2.2 The architecture of the CFF-RCNN model

Because of the reasons mentioned above, in this paper, a CFF-RCNN is constructed. The dimensionality reduction part uses the CFF strategy, operating as a feature fusion part at the same time (a dimension reducer and feature fusor). For the RCNN architecture, the CNN plays a role as a feature extractor and the RNN as a sequence recognizer. The detailed structure is shown in Fig 4.

Initially, the sEMG signals recorded from the electrodes of *C* channels are split into segments using an *s*-length sliding window with a *p*-length overlap. Each subsegment is denoted as X_t', where X_t' contains *N* steps of data with *C* channels. Therefore, the size of X_t' will be (1×*L*), where $L = N \times C$ (*L* is the length). Subsequently, X_t' is reshaped into a 3D frame X_t with size ($W \times H \times C$), where $W \times H = N$ (*W* is the width and *H* is the height). The input data of the network are $X = \{X_1, X_2, \ldots, X_T\}$, where *T* is the total number of samples. A time-distributed CNN structure is applied as the feature extractor, where all the kernel sizes are 3 for the convolutions. The first layer is a 64 convolutional layer with a stride of 1. Then, the output feature

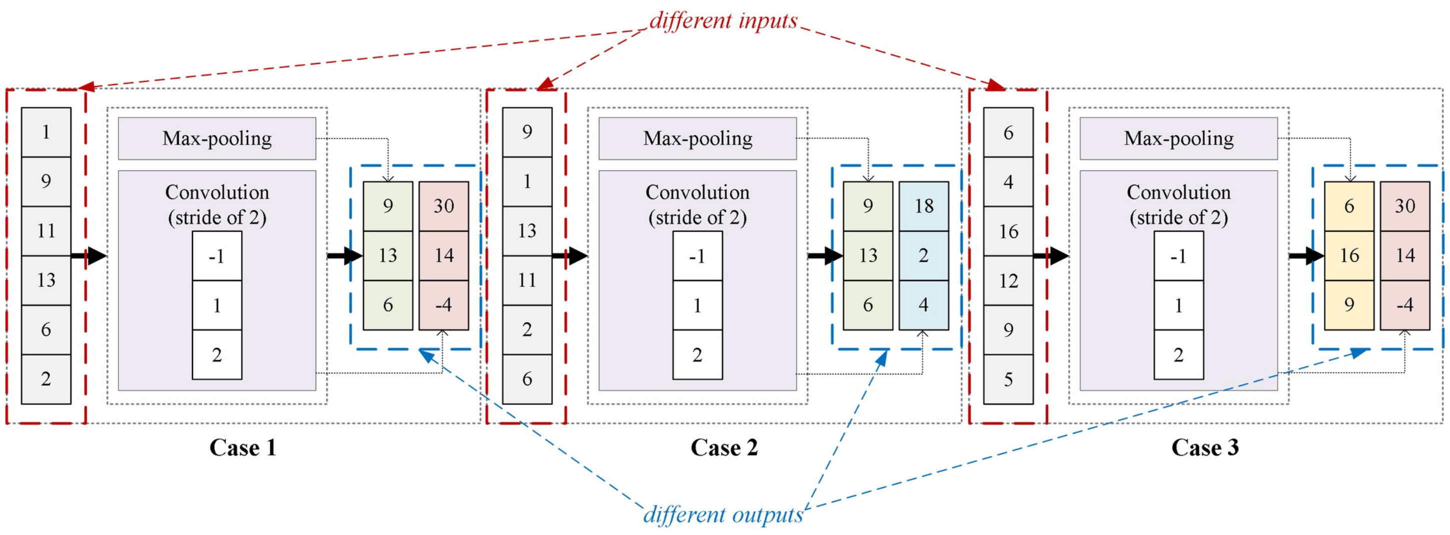


Fig 3. The procedure of the CFF strategy. Both the max-pooling and the 2-stride convolution are applied to the inputs. After doing so, distinguishable features are obtained to represent different sequential data.

https://doi.org/10.1371/journal.pone.0262810.g003

maps are transmitted to both the max-pooling layer and the 2-stride convolutional layer in parallel. The pooling size is 2, and the number of filters is 64. After that, the output map of the convolutional layer is concatenated immediately after that of the pooling layer, forming a feature map with 128 channels. Furthermore, to extract detailed features, another 1-stride convolutional layer with 64 filters is placed after the concatenate operation. In addition, a simple max-pooling layer is used for dimensionality reduction. The last three layers of the feature extractor are fully connected layers, with 512, 512, and 128 hidden units. The rectified linear unit (ReLU) is chosen as the activation function, and batch normalization is used for all the 1-stride convolutional layers mentioned above. For the sequence recognizer, an RNN will be connected with the outputs of the CNN feature extractor. Each LSTM unit has 512 nodes and a dropout rate of 0.5, followed by a fully connected layer and a softmax layer as the classifier. A comparison of structures between the RCNN and the CFF-RCNN is presented in Table 1.

3 Experiments

3.1 Experiment setups

Data acquisition: Performance of the novel CFF-RCNN model is evaluated using the first, second and fourth subdatasets of the NinaPro database [34, 44], denoted as DB1, DB2, and DB4.

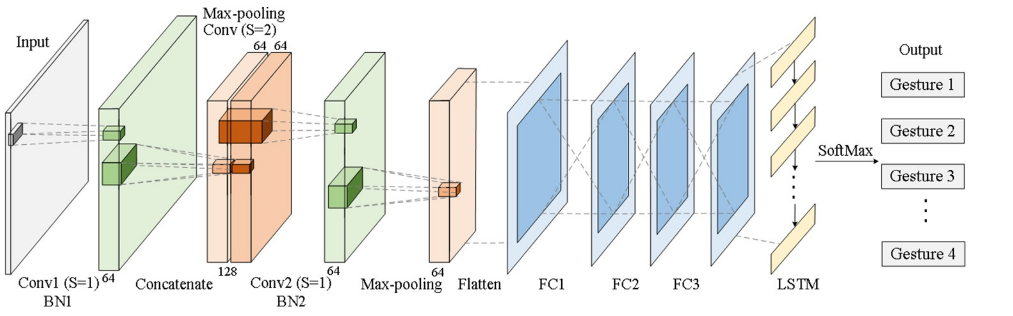


Fig 4. The structure of the CFF-RCNN.

https://doi.org/10.1371/journal.pone.0262810.g004

Table 1. The structure comparison between the RCNN and CFF-RCNN.

RCNN	CFF-RCNN	
Conv. 64 3	Conv. 64 3	
Batch Normalization	Batch Normalization	
Max-pooling 2	Max-pooling 2	Conv. 64 3/ S = 2
	Concatenate	
Conv. 64 3	Conv. 64 3	
Batch Normalization	Batch Normalization	
Max-pooling 2	Max-pooling 2	
Flatten	Flatten	
Fully connected 512	Fully connected 512	
Fully connected 512	Fully connected 512	
Fully connected 128	Fully connected 128	
LSTM 512	LSTM 512	
SoftMax	SoftMax	

https://doi.org/10.1371/journal.pone.0262810.t001

Detailed information on the databases is shown in Table 2. DB1 contains a total of 53 gestures (rest included) from 27 subjects, including 12 basic movements of the fingers (Exercise A), 8 isometric and isotonic hand configurations and 9 basic movements of the wrist (Exercise B), and 23 grasping and functional movements (Exercise C). It was recorded by 10 electrodes with a sampling rate of 100 Hz. DB2 collects 50 movements from 40 subjects, including 8 isometric and isotonic hand configurations and 9 basic movements of the wrist (Exercise B), 23 grasping and functional movements (Exercise C), and 9 force patterns (Exercise D). DB4 has 53 movements from 10 subjects, including 12 basic movements of the fingers (Exercise A), 8 isometric and isotonic hand configurations and 9 basic movements of the wrist (Exercise B), and 23 grasping and functional movements (Exercise C). Twelve electrodes were used to record the sEMG signals at a sampling rate of 2 kHz for both DB2 and DB4.

Data preprocessing: Several signal processing steps including synchronization, relabeling and filtering were performed in the Ninapro databases [34]. All the data streams were synchronized to the high-resolution timestamps (100 Hz for DB1, 2 kHz for DB2 and DB4) using linear interpolation. The resulting erroneous movement labels have been corrected by applying the generalized likelihood ratio algorithm and the Lidierth threshold based algorithm. The Delsys electrodes for DB2 and Cometa sensors for DB4 are not shielded against power line interferences. Therefore, prior to synchronization, these signals were cleaned from 50 Hz (and harmonics) power-line interference using a Hampel filter. Standardization and normalization procedures were applied, and labels were encoded using one-hot encoder.

Data segmentation: As described in Section 2.2, the pre-processed data are segmented using the sliding window method. The raw subsegment X_t' contains N steps of data with C channels and is reshaped into a 3D frame X_t with size ($W \times H \times C$), where $W \times H = N$ (W is the width and H is the height). In this paper, a 250 ms window with a 200 ms overlap is used for segmentation. For DB2 and DB4, N is set to 500 by simply multiplying the window length and

Table 2. Details of the three sEMG benchmark databases.

Database	Gestures	Channels	Subjects	Sampling rate (Hz)	Window length (ms)
DB1	53 (Exercise A, B & C)	10	27	100	250
DB2	50 (Exercise B, C & D)	12	40	2000	250
DB4	53 (Exercise A, B & C)	12	10	2000	250

https://doi.org/10.1371/journal.pone.0262810.t002

the sampling frequency, W and H, which are set to 5 and 100, respectfully. This forms the input data with size (5×100×12). For DB1, N is 25. In addition, owing to the downsampling of DB1, the input data are reshaped into (5×25×2) to obtain larger data maps.

Training details: For the proposed CFF-RCNN and compared RCNN structures, stochastic gradient descent with momentum (SGDM) is chosen as the optimizer with a momentum of 0.95. The initial learning rate is set at 0.002, and the weight decay is set at 0.0005. The loss function is cross-entropy loss: $Loss = -[y\log\hat{y} + (1-y)\log(1-\hat{y})]$, where y is the original label, and $\hat{y}$ is the result of the classification. The minibatch size is 20. The training epochs are set to 30 for DB2 and DB4 and 50 for DB1 due to the smaller map size. A 5-fold cross validation is used for model comparison. The networks operated on a workstation with an Intel Xeon E5-4210v2 central processing unit@ 2.2 GHz, NVIDIA GeForce GTX 2080Ti, and 128 GB access memory. The TensorFlow and Keras deep learning frameworks are used to design and train the networks.

3.2 Performance evaluation

To test the performance of the RCNN and CFF-RCNN, comparisons are made with traditional machine learning methods PCA, LDA and ANN and other state-of-the-art methods using the databases mentioned above.

We select four commonly used features, mean absolute value (MAV), zero crossings (ZS), slope sign changes (SSC), and waveform length (WL) to examine feature extraction performance. These features are calculated from each sEMG channel in a 250 ms sliding window with a 200 ms overlap in all methods tested. Both the PCA and LDA are used to reduce the dimensionality and analyze components, together with a KNN operating as a classifier. For ANN, three fully connected layers are built with the activation of ReLU, and the numbers of nodes are 100, 200, and 160. A dropout layer is added with a rate of 0.5, and softmax is used for classification. The ANN shares the same training optimizer and loss function with CFF-RCNN and RCNN, where the batch size is set to 32 and the number of epochs is 100.

We also compared with models from other works on the same benchmark dataset with all gestures. Traditional machine learnings tested include support vector machines (named LS-SVM) [45], random forests (RF) [34, 44] and LDA [45]. Other state-of-the-art approaches are AtzoriNet [34], ZhaiNet [46], transfer learning multi-scale kernel convolutional neural network (TL-MKCNN) [47], few-shot learning- hand gesture recognition (FS-HGR) [48], RNN with weight loss [49], a long short-term memory network with a multi-layer perceptron (LSTM+MLP) [50], CNN-LSTM [51], and attention-based hybrid CNN-RNN with feature-signal-image1 [32].

3.3 Comparison of the RCNN and CFF-RCNN

Experiments are designed to compare **accuracy**, **efficiency**, and **tolerance** of the RCNN and CFF-RCNN.

3.3.1 Experiment 1: Accuracy and efficiency test. The first experiment is set to compare prediction accuracies, total training times and training times at different training epochs. To present the training process, the epoch number is changed from 30 to 10 with a step of -5 for DB2 and DB4 and changed from 50 to 20 with a step of -10 for DB1. The accuracy is viewed for the same training epoch. The efficiency of RCNN and CFF-RCNN is evaluated from three aspects: 1. The number of epochs for the same accuracy. 2. The training time for the same number of epochs. 3. The training time for the same accuracy. A 5-fold cross validation is used and classification accuracies on DB1, DB2 and DB4 are compared between RCNN and CFF-RCNN using Wilcoxon Matched-Pairs Signed-Ranks Test. The significance level is set to

0.05. We also tested the accuracy without using the k-fold cross validation, where the proportions of the training, testing and valid sets are 0.8, 0.1 and 0.1.

3.3.2 Experiment 2: Robustness test. The second experiment is designed to test the performance of the networks when operating on a complex and chaotic dataset. Because sEMG signals vary significantly between subjects even with a precise electrode-placement control [52], the networks are usually trained specifically for each user in practice. In this experiment, both the RCNN and CFF-RCNN are trained using the total data of all the subjects in DB4 in order to test the robustness. The number of training epochs is 160.

4 Results

4.1 Performance evaluation

The classification accuracies of the machine learning methods, RCNN, and CFF-RCNN are illustrated in Fig 5. The accuracy in the figure is the mean value of all the subjects.

Classification accuracies of the proposed methods are evaluated using DB1, DB2, and DB4 data. On DB1 data, accuracies in the machine learning models using the same data are 43.80%, 77.60%, and 80.66%, with the highest being 80.66% when ANN is applied. CFF-RCNN model reaches an accuracy of 88.87% and improves the accuracy by at least 1.50% compared to the RCNN. The top three principal components of Subject 1 in DB1, obtained by the PCA and

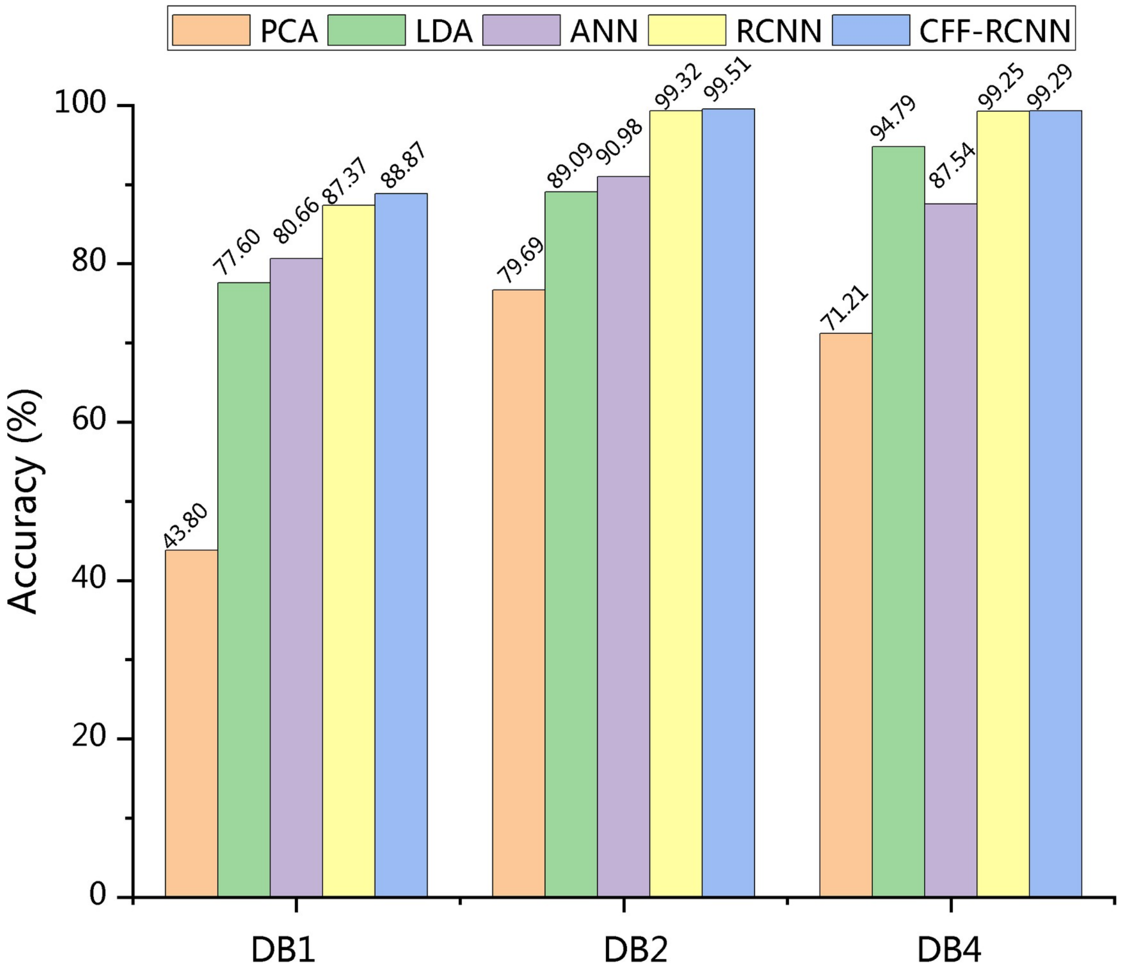


Fig 5. Classification accuracy of the proposed methods.

https://doi.org/10.1371/journal.pone.0262810.g005

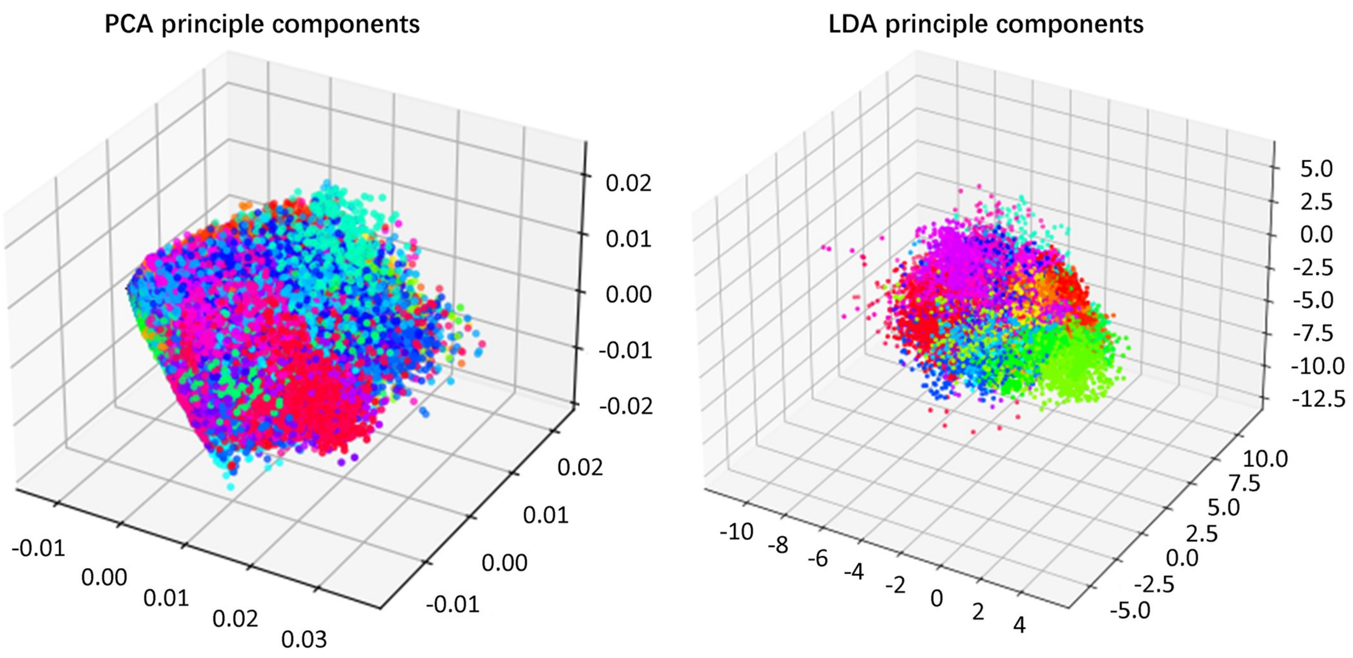


Fig 6. Principal components of DB1S1 obtained by the LDA and PCA.

https://doi.org/10.1371/journal.pone.0262810.g006

LDA, are illustrated in Fig 6. Both of the clusters are hard to distinguish with groups of plots overlapping each other, meaning that the classifiers cannot find the differences among gestures efficiently; the LDA clusters have circular groupings and can be better represented by the cluster centroids, resulting in a higher classification accuracy of 77.60%. On DB2 data, the traditional RCNN achieves 99.32% of accuracy, which is 8.34% higher than that of the ANN methods. CFF-RCNN increases the accuracy of the RCNN from 99.32% to 99.51%. On DB4 data, the LDA performs the best among all the tested machine learning methods, with an accuracy of 94.79%. RCNN has a much higher accuracy of 99.25%. CFF-RCNN shows a slightly higher (99.29%) accuracy than RCNN.

Accuracy results of RCNN, CFF-RCNN and models from other works can be found in Table 3. When classifying hand gestures with more than 50 gestures, CFF-RCNN shows better performance compared with machine learnings and other state-of-the-arts.

Table 3. Classification accuracy of the compared methods.

Methods	DB1	DB2	DB4
LS-SVM (IAV+MAV+RMS+WL) [45]	85.19 ± 13.32%	-	-
LDA(IAV(or MAV)+ CC) [45]	84.23± 9.58%	-	-
RF [34, 44]	75.32%	75.27%	69.13 ± 7.77%
AtzoriNet [34]	-	60.3 ± 7.7%	-
ZhaiNet [46]	-	78.71%	-
TL-MKCNN [47]	-	86.67%	82.29%
FS-HGR [48]	-	85.94%	-
RNN with weight loss [49]	79.3%	78.0%	-
LSTM+MLP [50]	75.45±8.97%	-	-
CNN-LSTM [51]	-	79.329%	-
Attention-based hybrid CNN-RNN [32]	87%	82.2%	-
RCNN	87.37 ± 3.77%	99.32 ± 0.55%	99.25 ± 0.13%
CFF-RCNN	88.87 ± 3.63%	99.51 ± 0.12%	99.29 ± 0.10%

https://doi.org/10.1371/journal.pone.0262810.t003

Table 4. Predicting accuracy, total training time, and training time per epoch of the RCNN and CFF-RCNN on DB1 at different epochs, along with the statistical analyses results.

Number of epochs	DB1						
	Predicting Accuracy			Total training time		Training time per epoch	
	RCNN	CFF-RCNN	p-value	RCNN	CFF-RCNN	RCNN	CFF-RCNN
50	87.37 ± 3.77%	88.63 ± 3.66%	<0.05	2899.955s	2761.043s	57.999s	55.221s
40	86.33 ± 3.95%	87.92 ± 3.71%	<0.05	2353.686s	2245.716s	58.842s	56.143s
30	85.08 ± 4.06%	86.42± 4.01%	<0.05	1680.139s	1727.415s	56.005s	57.581s
20	82.85 ± 4.34%	84.23 ± 4.26%	<0.05	1210.431s	1097.884s	60.522s	54.894s

https://doi.org/10.1371/journal.pone.0262810.t004

Based on these comparisons over the classification accuracies on DB1, DB2, and DB4 databases, the CFF strategy-integrated CFF-RCNN model outperforms traditional machine learning models and other state-of-the-arts, and further improves the classification result of a traditional RCNN architecture for all the tested datasets.

4.2 Comparison of the RCNN and CFF-RCNN

4.2.1 Experiment 1: Accuracy and efficiency test. The prediction accuracies, total training times and training times per epoch are shown in Tables 4–6, corresponding to DB1, DB2 and DB4. The 5-fold cross validation is used for model comparison and the accuracy in the table is the mean value ± standard deviation of that of all the subjects.

The results shown in Tables 4–6 are analyzed by the accuracy and the efficiency (using epoch numbers and training times). We want to understand, when recognizing gestures with many classes, how the CFF strategy impacts the classification accuracy and the training time, and whether this new dimensionality reduction strategy is valid in feature extractions and improves the performance of the network.

First, to analyze accuracies, results are viewed for the same training epoch. Conclusions can be drawn that at each sampling point of the epochs, the CFF-RCNN has a higher classification accuracy than the RCNN on all the datasets. When the networks are trained on DB1, the accuracy of the CFF-RCNN is approximately 1–2% higher than that of the RCNN (see Table 4). For DB2 and DB4, the improvements, which are 1–2%, are more clear for smaller epochs (see Tables 5 and 6). Although the gap of accuracy between RCNN and CFF-RCNN decreases with a larger number of epochs, CFF-RCNN still achieves a higher accuracy result. This is likely because CFF-RCNN converges much faster than RCNN when moving close to their saturation stages.

The efficiency of RCNN and CFF-RCNN are evaluated using the epoch numbers. The CFF-RCNN achieves higher accuracy with fewer training epochs. For DB1, the RCNN reaches

Table 5. Predicting accuracy, total training time, and training time per epoch of the RCNN and CFF-RCNN on DB2 at different epochs, along with the statistical analyses results.

Number of epochs	DB2						
	Predicting Accuracy			Total training time		Training time per epoch	
	RCNN	CFF-RCNN	p-value	RCNN	CFF-RCNN	RCNN	CFF-RCNN
30	99.32 ± 0.55%	99.51 ± 0.12%	<0.05	966.385s	1033.514s	32.213s	34.451s
25	99.15 ± 0.71%	99.44 ± 0.13%	<0.05	816.173s	910.904s	32.647s	36.436s
20	98.96 ± 0.67%	99.27 ± 0.20%	<0.05	698.677s	643.676s	34.934s	32.184s
15	98.06 ± 1.44%	98.80 ± 0.47%	<0.05	484.844s	542.245s	32.323s	36.150s
10	94.69 ± 2.61%	96.15 ± 1.51%	<0.05	323.722s	335.076s	32.372s	33.508s

https://doi.org/10.1371/journal.pone.0262810.t005

Table 6. Predicting accuracy, total training time, and training time per epoch of the RCNN and CFF-RCNN on DB4 at different epochs, along with the statistical analyses results.

Number of epochs	DB4						
	Predicting Accuracy			Total training time		Training time per epoch	
	RCNN	CFF-RCNN	p-value	RCNN	CFF-RCNN	RCNN	CFF-RCNN
30	99.25 ± 0.13%	99.29 ± 0.10%	0.333	976.322s	1057.923s	32.544s	35.264s
25	99.12 ± 0.14%	99.23 ± 0.12%	<0.05	924.086s	950.205s	36.963s	38.008s
20	98.86 ± 0.19%	99.03 ± 0.22%	<0.05	731.771s	673.798s	36.589s	33.690s
15	97.88 ± 0.50%	98.30 ± 0.39%	<0.05	512.327s	576.734s	34.155s	38.449s
10	93.56 ± 1.49%	94.98 ± 1.34%	<0.05	378.690s	338.427s	37.869s	33.843s

https://doi.org/10.1371/journal.pone.0262810.t006

86% of accuracy at 40 epochs and 87% of accuracy at 50 epochs, respectively. The CFF-RCNN reaches 86% of accuracy at 30 epochs and 87% of accuracy at 40 epochs, respectively. The CFF-RCNN experienced 10 epochs (20% of the total training epochs) fewer to reach the same level of accuracy as RCNN. To reach 99% of accuracy on DB2, CFF-RCNN uses 15 epochs, a 33% reduction in the number of epochs as compared to 25 epochs with RCNN. For DB4, the RCNN is trained with 20 and 30 epochs to achieve 98% and 99% of accuracy, respectively. The CFF-RCNN only uses 15 and 25 epochs, which is 5 epochs fewer for the same level of accuracy, corresponding to a reduction of 17% in the number of training epochs.

The efficiency of the CFF strategy is also assessed with the training times from the last four columns shown in Tables 4–6. Considering the training time for the same number of epochs, the CFF-RCNN sometimes has a slightly longer training time per epoch in general, increasing by 1–4 s, which is within tolerance (30 epochs in DB1; 30, 25, 15 and 10 epochs in DB2; 30, 25 and 15 epochs in DB4). At other epoch numbers, CFF-RCNN learns faster in a single epoch (50, 40 and 20 epochs in DB1; 20 epochs in DB2; 20 and 10 epochs in DB4). When trained on DB1, the CFF-RCNN has a better performance except at 30 epochs, illustrating that the CFF strategy can improve the feature extraction ability when the map size is small or the amount of data is limited.

The CFF-RCNN takes less time to train for the same accuracy. To achieve accuracies of 86%, 99% and 98% for DB1, DB2 and DB4, the CFF-RCNN uses 1727.415 s, 542.245 s and 576.734 s. The training time of the RCNN is 2353.686 s, 816.173 s and 731.771 s to arrive at the same accuracies. The CFF strategy reduces the training time by 26.61%, 33.56% and 21.19%, respectively.

Fig 7 shows comparisons of the training loss, validation loss, training accuracy, and validation accuracy of RCNN and CFF-RCNN. The curves illustrate that the proposed CFF-RCNN has a smaller loss value, higher accuracy, and faster convergence speed than the RCNN.

The statistical analysis results are illustrated in Tables 4–6 and Fig 8. Classification accuracies without using the k-fold cross validation can be found in S1–S3 Tables. Statistic analysis using the Wilcoxon Matched-Pairs Signed-Ranks Test revealed that CFF-RCNN significantly outperforms RCNN on DB1, DB2 and some sets of DB4 ($p<0.05$) (see Tables 4–6 and Fig 8). No significant difference ($p>0.05$) is found between the two methods when the number of epochs on DB4 is set to 30 (see Table 6 and Fig 8). Typically, when the number of epochs is smaller, the differences between RCNN and CFF-RCNN are more significant, indicating that CFF-RCNN has a better feature-extracting and learning ability with less training epochs.

4.2.2 Experiment 2: Robustness test. As shown in Table 7, when trained using the total data of all the subjects in DB4, the classification accuracy of the CFF-RCNN in the testing set is 83.0495%, much higher than that of the RCNN (74.6528%). In the prediction set, RCNN has an accuracy of 74.18%, and CFF-RCNN has an accuracy of 83.1931%, which is 9% higher than

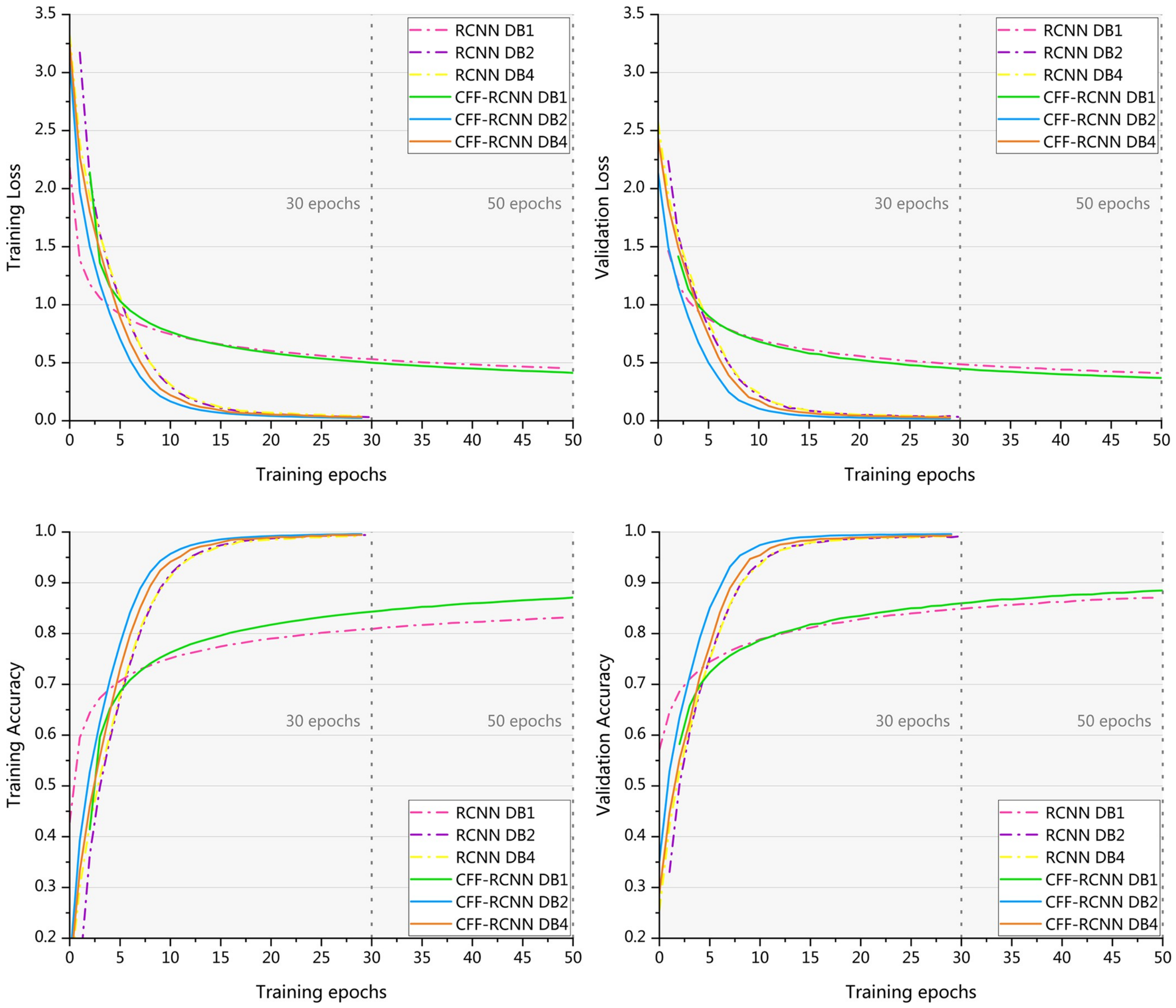


Fig 7. Loss and accuracy of the training and validation sets of the RCNN and CFF-RCNN on DB1, DB2 and DB4.

https://doi.org/10.1371/journal.pone.0262810.g007

its conventional counterpart. The training procedures are shown in Fig 9. In both loss and accuracy curves, the CFF-RCNN performs better in terms of a lower loss value, higher classification accuracy and faster convergence rate. The results indicate that the CFF-RCNN also shows a powerful data processing capability when dealing with mass and complicated datasets.

4.2.3 Summary of the experiments. The CFF strategy is superior in two aspects. First, it helps to achieve a higher accuracy in gesture classification than the traditional RCNN from the training process to the end. Second, it has a faster convergence rate (1727.415 s, 542.245 s and 576.734 s to achieve accuracies of 86%, 99% and 98% for DB1, DB2 and DB4) compared to the RCNN (2353.686 s, 816.173 s and 731.771 s), and therefore shorten the training process (a reduction of 26.61%, 33.56% and 21.19% of the training time). These advantages play an important role in making the CFF-RCNN a qualified method of classifying hand gestures with more than 50 movements.

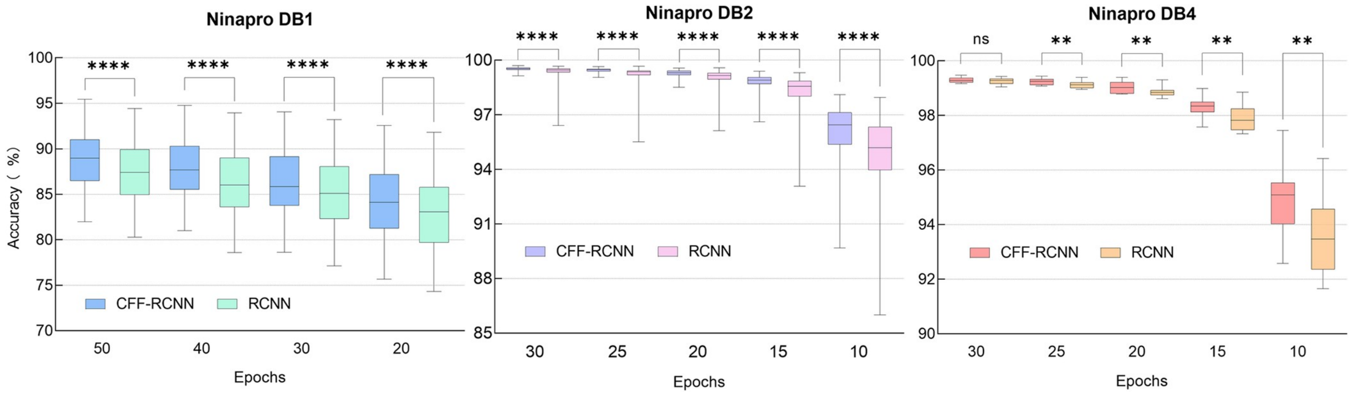


Fig 8. Comparisons between CFF-RCNN and RCNN on DB1, DB2 and DB4. P value style: 0.1234(ns), 0.0332(*), 0.0021(**),0.0002(***), <0.0001(****).

https://doi.org/10.1371/journal.pone.0262810.g008

5 Discussion and conclusions

In this article, we proposed and applied CFF strategy as a new dimensionality reduction strategy in an improved CFF-RCNN structure for sEMG-based hand gesture recognition tasks with more than 50 classes. The CFF strategy concatenates a pooling layer and a convolutional layer with a stride of 2, which reduces the information-loss that normally arises from models carrying a single pooling or convolutional layer.

Evaluations have been made thoroughly using three benchmark databases in the NinaPro database: DB1, DB2 and DB4. We compared the classification accuracies of the proposed CFF-RCNN model with traditional machine learnings and other state-of-the-art methods. Results showed that the CFF-RCNN achieved better accuracies of 88.87% in DB1 (53 gestures), 99.51% in DB2 (50 gestures) and 99.29% in DB4 (53 gestures). To demonstrate that the CFF strategy can also improve efficiency, we further designed the training process test and the complex dataset tolerance test to compare the RCNN and CFF-RCNN, and found that CFF-RCNN improved the convergence rate by 26.61%, 33.56% and 21.19%, which significantly shortened the required training time.

The present study is aimed at optimizing the network architecture for a better performance of feature-extracting in experiment settings. However, there are three main limitations of the present work. First, all the experiments are based on the NinaPro database. To explore potential clinical application of the CFF-RCNN model, accuracy and convergence rate can be tested using clinical data. Second, the present network is limited to offline training and testing, whereas in clinic applications such as prosthesis operation and stroke rehabilitation. Third, the parameters of the network were trained specifically for each user. Deep learning requires an unreasonable amount of effort from a single user in order to generate tens of thousands of examples as training data [53]. Therefore, new technologies such as transfer learning (TL) may be considered in order to lighten the user's workload [53].

Table 7. Comparison of RCNN and CFF-RCNN on all the subjects in DB4.

DB4	RCNN	CFF-RCNN
	160 epochs	**160 epochs**
Training accuracy	71.3608%	79.8770%
Testing accuracy	74.6528%	83.0495%
Predicting accuracy	74.1787%	83.1931%

https://doi.org/10.1371/journal.pone.0262810.t007

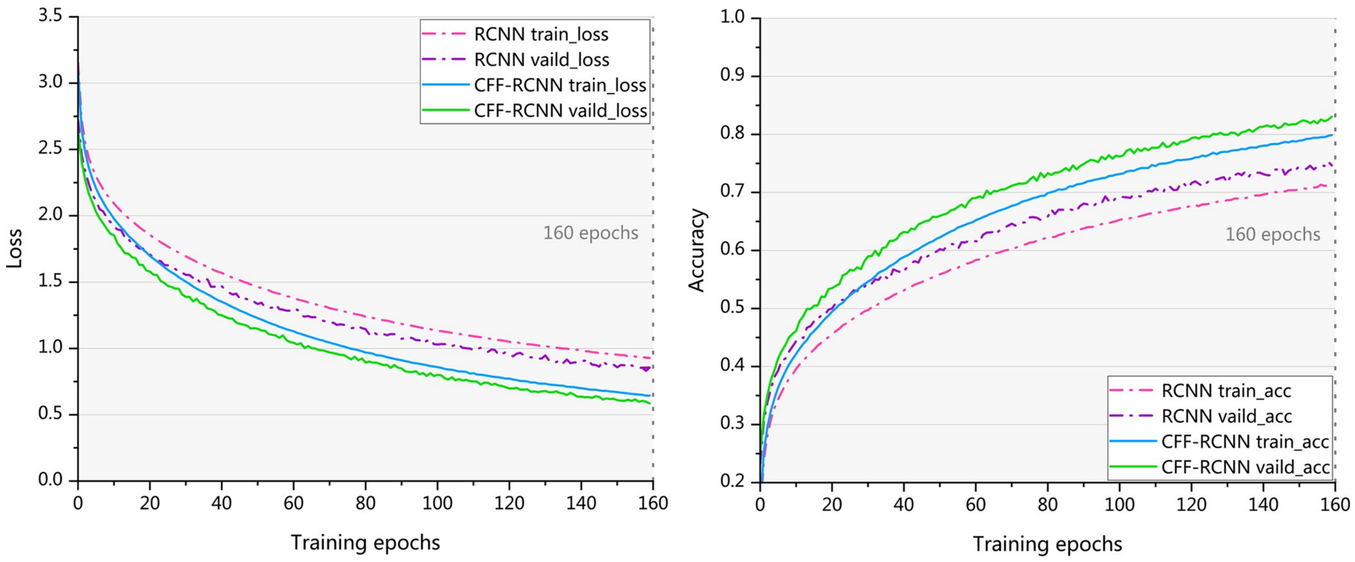


Fig 9. Loss and accuracy of the training and validation sets of the RCNN and CFF-RCNN on the total data of DB4.

https://doi.org/10.1371/journal.pone.0262810.g009

In the future, more works are required to overcome the drawbacks mentioned above. We can test the performance of the CFF-RCNN using data recorded by our sensing system. Experiments can be done on both intact and amputated subjects. The structure of the proposed model will be further implemented locally on wearable devices to achieve real-time classification. To reduce the training effort of a single person, new technologies, such as TL and other state-of-the-art scheme, can be added to the present structure as well. With these future improvements, the CFF strategy together with the CFF-RCNN model could enable a wider range of applications requiring multi-class recognitions, real-time classifications, and low-workload training demands.

Supporting information

S1 Table. Predicting accuracy without k-fold cross validation on DB1.
(DOCX)

S2 Table. Predicting accuracy without k-fold cross validation on DB2.
(DOCX)

S3 Table. Predicting accuracy without k-fold cross validation on DB4.
(DOCX)

Author Contributions

Conceptualization: Pufan Xu.

Funding acquisition: Fei Li, Haipeng Wang.

Investigation: Pufan Xu.

Methodology: Pufan Xu.

Project administration: Fei Li, Haipeng Wang.

Software: Pufan Xu.

Supervision: Fei Li, Haipeng Wang.

Validation: Fei Li, Haipeng Wang.

Writing – original draft: Pufan Xu.

Writing – review & editing: Fei Li, Haipeng Wang.

References

1. CJ DL. Decomposition of the EMG signal into constituent motor unit action potentials. Muscle Nerve. 1996; 18(12):1492–1494.
2. Chen M, Zhang X, Chen X, Zhou P. Automatic Implementation of Progressive FastICA Peel-Off for High Density Surface EMG Decomposition. IEEE transactions on neural systems and rehabilitation engineering. 2018; 26(1):144–152. https://doi.org/10.1109/TNSRE.2017.2759664 PMID: 28981419
3. Ozdemir MA, Kisa DH, Guren O, Onan A, Akan A. EMG based Hand Gesture Recognition using Deep Learning. 2020 Medical Technologies Congress (TIPTEKNO); 2020. https://doi.org/10.1109/TIPTEKNO50054.2020.9299264
4. Alibhai Z, Burreson T, Stiller M, Ahmad I, Clark A. A Human-Computer Interface For Smart Wheelchair Control Using Forearm EMG Signals. 2020 3rd International Conference on Data Intelligence and Security (ICDIS); 2020. https://doi.org/10.1109/ICDIS50059.2020.00011
5. Zhang M, Zhang W, Zhang B, Wang Y, Li G. Feature selection of mime speech recognition using surface electromyography data. 2019 Chinese Automation Congress (CAC); 2019. https://doi.org/10.1109/CAC48633.2019.8996646
6. Artemiadis PK, Kyriakopoulos KJ. EMG-Based Control of a Robot Arm Using Low-Dimensional Embeddings. IEEE Transactions on Robotics. 2010; 26(2):393–398. https://doi.org/10.1109/tro.2009.2039378
7. Karlik B, Tokhi MO, Alci M. A fuzzy clustering neural network architecture for multifunction upper-limb prosthesis. IEEE Transactions on Biomedical Engineering. 2003; 50(11):1255–1261. https://doi.org/10.1109/TBME.2003.818469 PMID: 14619995
8. Asif AR, Waris A, Gilani SO, Jamil M, Ashraf H, Shafique M, et al. Performance Evaluation of Convolutional Neural Network for Hand Gesture Recognition Using EMG. Sensors (Basel). 2020; 20(6). https://doi.org/10.3390/s20061642 PMID: 32183473
9. Al-Timemy AH, Bugmann G, Escudero J, Outram N. Classification of finger movements for the dexterous hand prosthesis control with surface electromyography. IEEE Journal of Biomedical and Health Informatics. 2013; 17(3):608–618. https://doi.org/10.1109/jbhi.2013.2249590 PMID: 24592463
10. Castiblanco JC, Ortmann S, Mondragon IF, Alvarado-Rojas C, Jöbges M, Colorado JD. Myoelectric pattern recognition of hand motions for stroke rehabilitation. Biomedical Signal Processing and Control. 2020;57. https://doi.org/10.1016/j.bspc.2019.101737
11. Xiong D, Zhang D, Zhao X, Zhao Y. Deep Learning for EMG-based Human-Machine Interaction: A Review. IEEE/CAA Journal of Automatica Sinica. 2021; 8(3):512–533. https://doi.org/10.1109/jas.2021.1003865
12. Naik GR, Acharyya A, Nguyen HT. Classification of finger extension and flexion of EMG and Cyberglove data with modified ICA weight matrix. Conference of the IEEE Engineering in Medicine and Biology Society. 2014; 2014:3829–3832. https://doi.org/10.1109/EMBC.2014.6944458 PMID: 25570826
13. Liu J, Zhou P. A novel myoelectric pattern recognition strategy for hand function restoration after incomplete cervical spinal cord injury. IEEE Trans Neural Syst Rehabil Eng. 2013; 21(1):96–103. https://doi.org/10.1109/TNSRE.2012.2218832 PMID: 23033334
14. Naik GR, Selvan SE, Gobbo M, Acharyya A, Nguyen HT. Principal Component Analysis Applied to Surface Electromyography: A Comprehensive Review. IEEE Access. 2016; 4:4025–4037. https://doi.org/10.1109/access.2016.2593013
15. Saeed B, Zia-ur-Rehman M, Gilani SO, Amin F, Waris A, Jamil M, et al. Leveraging ANN and LDA Classifiers for Characterizing Different Hand Movements Using EMG Signals. ARABIAN JOURNAL FOR SCIENCE AND ENGINEERING. 2021; 46(2):1761–1769. https://doi.org/10.1007/s13369-020-05044-x
16. Zhang X, He H, Yang Q. Real-time implementation of a self-recovery EMG pattern recognition interface for artificial arms. 2013 35th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). 2013:5926–5929. https://doi.org/10.1109/EMBC.2013.6610901
17. Simao M, Mendes N, Gibaru O, Neto P. A Review on Electromyography Decoding and Pattern Recognition for Human-Machine Interaction. IEEE Access. 2019:39564–39582. https://doi.org/10.1109/ACCESS.2019.2906584

18. Nasri N, Orts-Escolano S, Gomez-Donoso F, Cazorla M. Inferring Static Hand Poses from a Low-Cost Non-Intrusive sEMG Sensor. Sensors (Basel). 2019; 19(2). https://doi.org/10.3390/s19020371 PMID: 30658480

19. Phinyomark A, Phukpattaranont P, Limsakul C. Feature reduction and selection for EMG signal classification. Expert Systems with Applications. 2012; 39(8):7420–7431. https://doi.org/10.1016/j.eswa.2012.01.102

20. Mujahid A, Awan MJ, Yasin A, Mohammed MA, Damasevicius R, Maskeliunas R, et al. Real-Time Hand Gesture Recognition Based on Deep Learning YOLOv3 Model. Applied Sciences-Basel. 2021; 11(9).

21. Cote-Allard U, Fall CL, Campeau-Lecours A, Gosselin C, Laviolette F, Gosselin B. Transfer learning for sEMG hand gestures recognition using convolutional neural networks. 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC); 2017. https://doi.org/10.1109/SMC.2017.8122854

22. Ameri A, Akhaee MA, Scheme E, Englehart K. Real-time, simultaneous myoelectric control using a convolutional neural network. PLoS One. 2018; 13(9):1–13. https://doi.org/10.1371/journal.pone.0203835 PMID: 30212573

23. Manfredo A, Matteo C, Henning MJFiN. Deep Learning with Convolutional Neural Networks Applied to Electromyography Data: A Resource for the Classification of Movements for Prosthetic Hands. 2016; 10.

24. Borra D, Andalò A, Severi S, Corsi C. On the Application of Convolutional Neural Networks for 12-lead ECG Multi-label Classification using Datasets from Multiple Centers. 2020 Computing in Cardiology Conference; 2020.

25. Plizzari C, Cannici M, Matteucci M. Skeleton-based Action Recognition via Spatial and Temporal Transformer Networks. Computer Vision and Image Understanding. 2020:208–209. https://doi.org/10.1016/j.cviu.2021.103219

26. Li C, Luo Y, Han C, Li J, Yoshioka T, Zhou T, et al. Dual-Path RNN for Long Recording Speech Separation. 2021 IEEE Spoken Language Technology Workshop (SLT); 865–872. https://doi.org/10.1109/slt48900.2021.9383514

27. Wang S, Yao R, Tsiftsis TA, Miridakis NI, Qi N. Signal Detection in Uplink Time-Varying OFDM Systems Using RNN With Bidirectional LSTM. IEEE Wireless Communications Letters. 2020; 9(11):1947–1951. https://doi.org/10.1109/lwc.2020.3009170

28. Zhao T. Deep Multimodal Learning: An Effective Method for Video Classification. 2019 IEEE International Conference on Web Services (ICWS); 398–402. https://doi.org/10.1109/icws.2019.00071

29. Sun P, Zhang R, Jiang Y, Kong T, Xu C, Zhan W, et al. Sparse R-CNN: End-to-End Object Detection with Learnable Proposals. 2020. Available from: https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.2011.12450&lang=zh-cn&site=eds-live.

30. Wu Z, Wang X, Jiang YG, Ye H, Xue X. Modeling Spatial-Temporal Clues in a Hybrid Deep Learning Framework for Video Classification. IEEE Transactions on Multimedia; 2015. https://doi.org/10.1109/TMM.2015.2432671 PMID: 26557047

31. Mao Q, Zhu Q, Rao Q, Jia H, Luo S. Learning Hierarchical Emotion Context for Continuous Dimensional Emotion Recognition From Video Sequences. IEEE Access. 2019; 7:62894–62903. https://doi.org/10.1109/access.2019.2916211

32. Hu Y, Wong Y, Wei W, Du Y, Kankanhalli M, Geng W. A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition. PLoS One. 2018; 13(10):1–18. https://doi.org/10.1371/journal.pone.0206049 PMID: 30376567

33. Zhang L, Liu G, Han B, Wang Z, Zhang T. sEMG Based Human Motion Intention Recognition. Journal of Robotics. 2019; 2019(1):1–12. https://doi.org/10.1155/2019/3679174

34. Atzori M, Gijsberts A, Castellini C, Caputo B, Hager A-GM, Elsig S, et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Scientific Data. 2014; 1. https://doi.org/10.1038/sdata.2014.53 PMID: 25977804

35. Atzori M, Gijsberts A, Kuzborskij I, Elsig S, Hager AM, Deriaz O, et al. Characterization of a Benchmark Database for Myoelectric Movement Classification. IEEE transactions on neural systems and rehabilitation engineering. 2015; 23(1):73–83. https://doi.org/10.1109/TNSRE.2014.2328495 PMID: 25486646

36. Manfredo A, Arjan G, Claudio C, Barbara C, Mittaz AG, Simone E, et al. Effect of clinical parameters on the control of myoelectric robotic prosthetic hands. Journal of Rehabilitation Research & Development. 2016; 53(3):345–358. https://doi.org/10.1682/JRRD.2014.09.0218 PMID: 27272750

37. Wu Y, Zheng B, Zhao Y. Dynamic Gesture Recognition Based on LSTM-CNN. 2018 Chinese Automation Congress (CAC); 2018 30 Nov.-2 Dec. 2018. https://doi.org/10.1109/CAC.2018.8623035

38. Springenberg JT, Dosovitskiy A, Brox T, Riedmiller M. Striving For Simplicity: The All Convolutional Net. 2014. Available from: http://arxiv.org/abs/1412.6806.

39. Sabour S, Frosst N, Hinton GE. Dynamic Routing Between Capsules. 2017. Available from: https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1710.09829&lang=zh-cn&site=eds-live.

40. Ruderman A, Rabinowitz NC, Morcos AS, Zoran D. Pooling is neither necessary nor sufficient for appropriate deformation stability in CNNs. 2018. Available from: http://arxiv.org/abs/1804.04438.

41. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521(7553):436–444. https://doi.org/10.1038/nature14539 PMID: 26017442

42. Jaderberg M, Simonyan K, Zisserman A, Kavukcuoglu K. Spatial Transformer Networks. 2015. Available from: https://search.ebscohost.com/login.aspx?direct=true&db=edsarx&AN=edsarx.1506.02025&lang=zh-cn&site=eds-live.

43. Teh YW. Learning to parse images. 2000. Available from: https://search.ebscohost.com/login.aspx?direct=true&db=edsndl&AN=edsndl.oai.union.ndltd.org.LACETR.oai.collectionscanada.gc.ca.OOAMICUS.25612529&lang=zh-cn&site=eds-live.

44. Pizzolato S, Tagliapietra L, Cognolato M, Reggiani M, Muller H, Atzori M. Comparison of six electromyography acquisition setups on hand movement classification tasks. PLoS One. 2017; 12(10):1–17. https://doi.org/10.1371/journal.pone.0186132 PMID: 29023548

45. Nazemi A, Maleki A. Artificial neural network classifier in comparison with LDA and LS-SVM classifiers to recognize 52 hand postures and movements. International Econference on Computer & Knowledge Engineering; 2014. https://doi.org/10.1109/ICCKE.2014.6993343

46. Zhai X, Tin C, Jelfs B, Chan RHM. Self-Recalibrating Surface EMG Pattern Recognition for Neuroprosthesis Control Based on Convolutional Neural Network. Frontiers in Neuroscience. 2017; 11. https://doi.org/10.3389/fnins.2017.00379 PMID: 28744189

47. Zou Y, Cheng L. A Transfer Learning Model for Gesture Recognition Based on The Deep Feature Extracted by CNN. IEEE Transactions on Artificial Intelligence. 2021:1. https://doi.org/10.1109/tai.2021.3098253

48. Rahimian E, Zabihi S, Asif A, Farina D, Atashzar SF, Mohammadi A. FS-HGR: Few-shot Learning for Hand Gesture Recognition via ElectroMyography. IEEE transactions on neural systems and rehabilitation engineering. 2021; 29:1004–1015. https://doi.org/10.1109/TNSRE.2021.3077413 PMID: 33945480

49. Koch P, Phan H, Maa M, Katzberg F, Mertins A. Recurrent Neural Networks with Weighting Loss for Early Prediction of Hand Movements. 26th European Signal Processing Conference (EUSIPCO 2018); 2018.

50. He Y, Fukuda O, Nan B, Okumura H, Yamaguchi N. Surface EMG Pattern Recognition Using Long Short-Term Memory Combined with Multilayer Perceptron. 2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC); 2018. https://doi.org/10.1109/EMBC.2018.8513595

51. Huang D, Chen B. Surface EMG Decoding for Hand Gestures Based on Spectrogram and CNN-LSTM. 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI); 2019. https://doi.org/10.1109/CCHI.2019.8901936

52. Castellini C, Fiorilla AE, Sandini G. Multi-subject/daily-life activity EMG-based control of mechanical hands. J Neuroeng Rehabil. 2009; 6:41. https://doi.org/10.1186/1743-0003-6-41 PMID: 19919710

53. Cote-Allard U, Fall CL, Drouin A, Campeau-Lecours A, Gosselin C, Glette K, et al. Deep Learning for Electromyographic Hand Gesture Signal Classification Using Transfer Learning. IEEE transactions on neural systems and rehabilitation engineering. 2019; 27(4):760–771. https://doi.org/10.1109/TNSRE.2019.2896269 PMID: 30714928