Hindawi
Complexity
Volume 2020, Article ID 6642463, 15 pages
https://doi.org/10.1155/2020/6642463

Research Article

A Novel PSO-Based Optimized Lightweight Convolution Neural Network for Movements Recognizing from Multichannel Surface Electromyogram

Xiu Kan,[1,2] Dan Yang,[1] Le Cao,[1] Huisheng Shu,[3] Yuanyuan Li,[1] Wei Yao,[1] and Xiafeng Zhang[1]

[1]*School of Electronic and Electrical Engineering, Shanghai University of Engineering Science, Shanghai 201620, China*
[2]*School of Mathematics, Southeast University, Nanjing 210096, China*
[3]*School of Science, Donghua University, Shanghai 201620, China*

Correspondence should be addressed to Xiu Kan; xiu.kan@sues.edu.cn

Received 9 October 2020; Revised 29 October 2020; Accepted 18 November 2020; Published 4 December 2020

Academic Editor: Atif Khan

As the medium of human-computer interaction, it is crucial to correctly and quickly interpret the motion information of surface electromyography (sEMG). Deep learning can recognize a variety of sEMG actions by end-to-end training. However, most of the existing deep learning approaches have complex structures and numerous parameters, which make the network optimization problem difficult to realize. In this paper, a novel PSO-based optimized lightweight convolution neural network (PLCNN) is designed to improve the accuracy and optimize the model with applications in sEMG signal movement recognition. With the purpose of reducing the structural complexity of the deep neural network, the designed convolution neural network model is mainly composed of three convolution layers and two full connection layers. Meanwhile, the particle swarm optimization (PSO) is used to optimize hyperparameters and improve the autoadaptive ability of the designed sEMG pattern recognition model. To further indicate the potential application, three experiments are designed according to the progressive process of body movements with respect to the Ninapro standard data set. Experiment results demonstrate that the proposed PLCNN recognition method is superior to the four other popular classification methods.

1. Introduction

Human-computer interaction is one of the most popular topics in the field of signal processing. Detecting and judging the behavior intention leads to information communication between human and computer. Generally, physiological signals and RGB images are two mainstream pieces of information to capture human activities [1–5]. The recognition of physiological-signals-based body movements is the key to human-computer interaction [6–8], due to the problems of the image, such as occlusion by the environment, inability to be distinguished accurately, and difficulty in being segmented. In particular, as a physiological signal of the human body, sEMG is a complex electrical signal produced by muscle contraction and relaxation, which carries the movement information of corresponding parts. Decoding sEMG to recognize human behavior characteristics is called sEMG pattern recognition [9, 10]. Therefore, the machine is capable of representing the human movement intention; moreover, it is the popular research frontier in the field of human-computer interaction.

sEMG pattern recognition consists of the following three parts: signal preprocessing, feature extraction, and feature classification. Signal preprocessing involves filtering, denoising, and detecting active segments, etc. [11]. Feature extraction includes the following four aspects: time domain, frequency domain, time-frequency domain, and nonlinear dynamics; see [12, 13]. The papers [14–17] adopt different improved methods combining time domain and time-frequency domain to extract representative features of EMG.

After the sEMG feature is extracted, it is sent to the classifier for training. Generally, traditional machine learning algorithms such as support vector machine (SVM), random forest (RF), and k-nearest neighbour (KNN) are used in previous research. Particularly, sEMG is identified by changing the rejudgment condition of SVM's misclassification data in [18, 19]. It has been proven that traditional algorithms have achieved better results. However, since the voltage of sEMG is relatively weak (mV level), the voltage is susceptible to noise interference. It is difficult for the features extracted by traditional methods to achieve accurate results for actual requirements. Moreover, traditional methods need to select features and classifiers manually.

Compared with the traditional sEMG classification method, the neural network can avoid describing and extracting the features of weak sEMG data. It can learn the features independently and combine the extracted features with classification to improve recognition accuracy [20–24]. In recent years, EMG pattern recognition has been combined with deep learning and neural networks, such as artificial neural network (ANN), convolutional neural network (CNN), and recurrent neural network (RNN), which have been used to recognize sEMG. In the process of sEMG acquisition, the electrodes have different spatial positions. CNN is effective in sEMG acquisition because it can extract the local and global features of sEMG. Fifty-two kinds of sEMG gestures have been recognized for the first time by a simple CNN model that achieves the accuracy of general classical algorithms in [25]. A CNN network model, composed of four convolution layers, four pooling layers, and two fully connected layers, has been used to improve the variability of different testers in hand movements in [26]. Moreover, some improved algorithms based on CNN technique have been investigated to recognize the sEMG gesture during the past few years; see [27, 28].

Till now, sEMG research based on CNN has also faced some problems. Firstly, the complexity computation of CNN network with many layers is expensive, which leads to the delay in the interaction between humans and computers. For example, the recognition process speed in [29] is slow even eleven convolution layers are used to extract features. Secondly, it is difficult to manually set hyperparameters and conduct experiments to verify this setting. Most researchers set the network hyperparameters, which affect the accuracy of the model, based on previous experience or just pick them randomly [30]. These artificial settings accumulate the uncertainty of sEMG recognition. Finally, the accuracy of EMG pattern recognition can reach more than 90% when classifying less than ten gestures. On the contrary, the accuracy will be significantly reduced when more than ten gestures are required to be classified. This limits the ability of the model to interpret human motion intention. The result shows that the more the actions of dual-flow network based on CNN [30] are recognized, the less the accuracy of the network is.

Following the above discussions, the purpose of this paper is to propose a new hand movement recognition method with applications to the analysis of multichannel sEMG signals. PSO is used to adjust the network hyperparameters, eliminate interference, classify more gesture actions, and improve the accuracy of EMG pattern recognition. The proposed method can even recognize three groups of hand movements in terms of sensitivity, flexibility, and dynamics, which is superior to most of the state-of-the-art methods on the Ninapro standard data set. The main contributions are summarized as follows:

(1) A novel deep learning framework is constructed based on lightweight convolutional neural network (LWCNN). The proposed LWCNN, which just contains 10952 parameters and three convolution layers, is an end-to-end sEMG feature extraction network. It solves the problem of computational complexity and improves the efficiency of human-computer interaction.

(2) The PSO algorithm is exploited to optimize the hyperparameters of LWCNN, which leads to a novel PLCNN classification model. A linear decline strategy is used to redefine the inertia weight to avoid the situation in which PSO falls into local optimum. Meanwhile, the cross-entropy loss function of the network is set as the fitness function to make predicted results close to the real results.

(3) The proposed PLCNN deep learning algorithm is successfully employed to analyze the multichannel sEMG signals in order to verify hand movement recognition. With an appropriate recognition method, the speed and accuracy of hand movement recognition could be much increased. Thus, the efficiency of human-computer interaction can be improved.

The remainder of this paper is organized as follows. The proposed framework for sEMG pattern recognition and the designed algorithms are introduced in Section 2. Descriptions of data and experimental configuration are introduced in Section 3. The PLCNN method for sEMG pattern recognition is discussed and the experiments results, demonstrating the effectiveness of the developed framework, are provided in Section 4. Finally, conclusions are given in Section 5.

2. System Framework and Method

As shown in Figure 1, we propose a novel deep learning network for sEMG pattern recognition. The network is suitable for a simple one-dimensional signal, which has low complexity and few parameters. Moreover, the PSO algorithm is used to automatically select the network's hyperparameters to avoid expensive computation in gesture recognition.

2.1. Lightweight Convolution Neural Network (LWCNN) Model. Researchers have found that CNN has important spatial significance for sEMG, because CNN has excellent performance on image and speech processing. In respect of EMG pattern recognition, CNN can extract the spatial correlation features of sEMG signal according to the position of electrodes in space. Feature extraction is integrated into

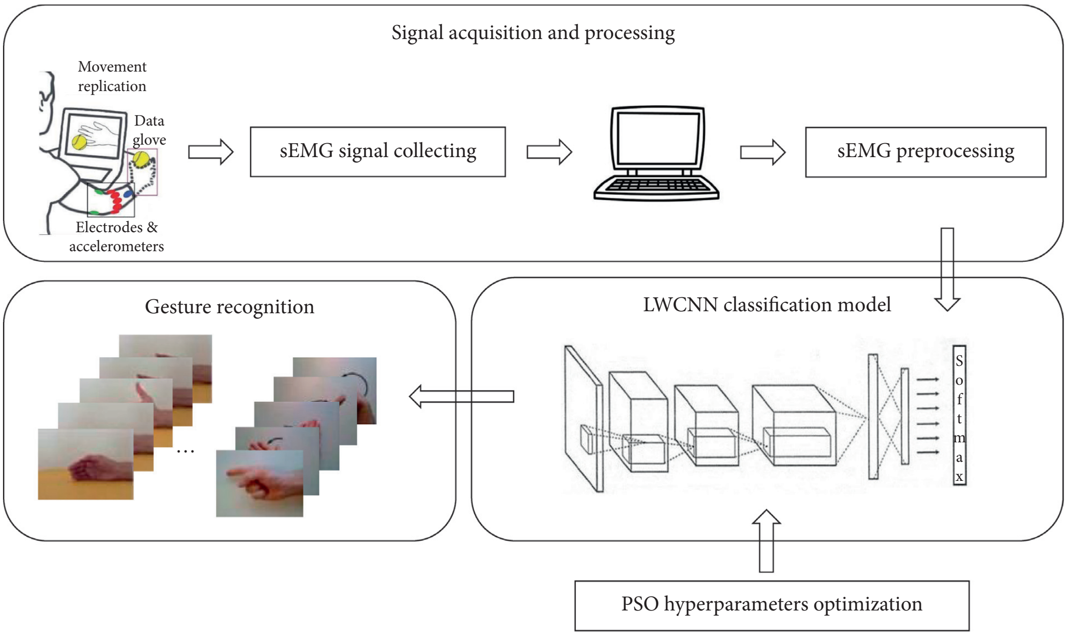

Signal acquisition and processing
Movement replication
Data glove
Electrodes & accelerometers
sEMG signal collecting
sEMG preprocessing
Gesture recognition
LWCNN classification model
Softmax
PSO hyperparameters optimization

Figure 1: The proposed network architecture for sEMG recognition. The original sEMG is sequentially fed to LWCNN for feature extraction to obtain corresponding feature maps, which has been preprocessed by filtering and denoising. The feature map is then flattened into feature vectors to the fully connected layer for preliminary identification. To better represent different gesture feature, the PSO model is adopted finally.

feature classification by local connection and weight sharing, which removes the complexity of traditional recognition methods where feature extraction and data reconstruction are needed. The main network structure of CNN includes four parts: convolution layer, pooling layer, full connection layer, and activation function.

The general CNN network structure is too complex, which often leads directly to a slow convergence rate of training. Furthermore, complex structures can cause network overfitting and even reduce the accuracy of EMG pattern recognition. Moreover, due to the structural complexity increased in network structure, it is difficult to optimize the hyperparameters of the model. Therefore, we design a lightweight CNN structure network called LWCNN, which not only includes fewer parameters but also could accelerate training speed. As shown in Table 1, the convolution layers' parameters of LWCNN are significantly less than those of the state-of-the-art methods, which is more conducive to subsequent optimization.

It can be seen from Figure 2 that the designed LWCNN consists of four parts for sEMG pattern recognition, mainly inputting sEMG data, extracting sEMG features, classifying sEMG features, and outputting classification results, where a, b, c are the numbers of filters and d, e, f are filter sizes.

(i) **Part 1**: the first part is the input module. The preprocessed sEMG signal is inputted with the size of $1 \times$ insize, which is convenient for feature extraction in subsequent convolutional layers.

(ii) **Part 2:** the second part consists of six layers: three convolution layers, two pooling layers, and an activation function. Convolution layers are the most important layers in the whole structure: conv-1, conv-2, and conv-3. Pooling layers are pool-1 and pool-2. The first layer is conv-1, which has a filters of size $1 \times d$. The sEMG is convoluted by multiple filters to obtain the spatial features of the signal. The second layer is the ReLU activation function, which preserves and maps the activated features to increase the nonlinear representation of the network. The third layer is pool-1, which performs an average pooling for the features. It not only reduces the size of features to simplify the complexity of the network, but also extracts the representative features to the corresponding feature map. The fourth layer is conv-2 with b filters of size $1 \times e$. Afterwards, there is a pool-2 layer for the local average. The last layer is conv-3, which contains c filters with the size of $1 \times f$.

(iii) **Part 3:** the third part consists of a flatten layer, two dropout layers, and two full connection layers called dense-1 and dense-2. This part mainly synthesizes all the extracted sEMG features from the second part and expands them to obtain global information. Moreover, part features are discarded to prevent network overfitting. Finally, the processed features are mapped to the simple label space to prepare for gesture recognition.

(iv) **Part 4:** the fourth part is the output module, in which the softmax function is used to classify the features of the network. Then, we obtain the predicted results of hand movements as output.

Table 1: Various model parameters based on EMG pattern recognition.

Authors	Methods	Number of convolution layers	Convolution layer parameters
Manfredo et al. [25]	CNN	5	—
Runze et al. [30]	Dual-flow net	4	—
Ulysse et al. [29]	ConvNet	5	67179
Lin et al. [28]	EMGNet	4	34311
	LCNN	2	154353
Ours	LWCNN	3	10952

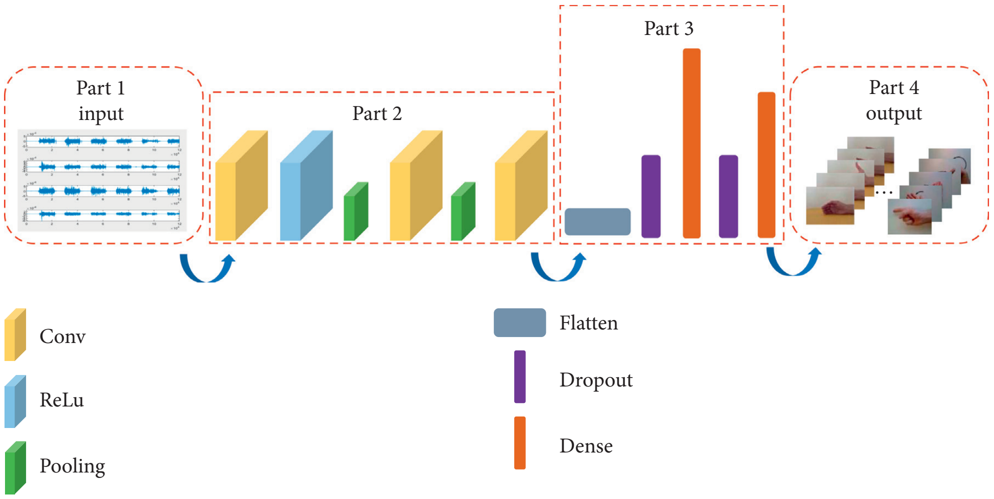


Figure 2: Network structure of the LWCNN.

Algorithm 1 gives the process of LWCNN model training for sEMG signal, where a, b, c, d, e, f, batch size, and learning rate are the network hyperparameters which PSO needs for optimization as shown in Table 2.

2.2. The PSO Optimized LWCNN Model. LWCNN model has a significant effect on the spatial processing of the sEMG signal, but the settings of hyperparameters can directly affect the performance of the model. Manually set hyperparameters not only complicate the work, but also slow down the EMG recognition process. Moreover, when the hyperparameters of LWCNN are optimized manually, the generalization ability of the model is weak.

PSO is an iterative optimization algorithm [31, 32]. Its idea comes from the observation of the foraging behavior of birds. The basic idea is that each individual in the population iteratively updates itself according to the initial value. To obtain the optimal result, individuals share information with each other to adjust the population. Due to the advantages of search speed, wide search range, simple structure, and easy implementation, PSO stands out among many optimization algorithms. In this paper, we utilize PSO to adjust the network hyperparameters and automatically find the most suitable deep learning structure for sEMG pattern recognition.

PLCNN takes 10 hyperparameters in LWCNN structure as 10 dimensions of each particle in PSO. These hyperparameters include the number of convolution kernels, the size of convolution kernels, activation function, batch size, and learning rate. According to the classic networks such as VGG and ResNet [33, 34], the initial value range of PLCNN is shown in Table 2.

Suppose that, in a ten-dimensional space, m particles constitute a population without mass and volume. Each particle is represented by the velocity vector and position vector in space. In the k-th iteration, the l-th particle velocity vector $v_l^k = (v_{l1}^k, v_{l2}^k, \ldots, v_{l10}^k)$, and the position vector $x_l^k = (x_{l1}^k, x_{l2}^k, \ldots, x_{l10}^k)$. Then the l-th particle performs the next update according to its best position in all iterations, where the best position is found by m particles in the population. The best position of it is $x_{lp} = (x_{lp}(1), x_{lp}(2), \ldots, x_{lp}(10))$ and the best solution to the population is $x_g = (x_g(1), x_g(2), \ldots, x_g(10))$. The velocity and position updating formulas of the l-th particle in the $(k+1)$-th iteration are

$$v_l^{k+1} = wv_l^k + c_1 r_1 \left(x_{lp}^k - x_l^k\right) + c_2 r_2 \left(x_g^k - x_l^k\right), \tag{1}$$

$$x_l^{k+1} = x_l^k + v_l^{k+1}, \tag{2}$$

where k is the number of iterations; x_{lp}^k is the best position of particle l after the k-th iteration; x_g^k is the best position of the population of particles l after the k-th iteration; w is the inertial weight; c_1,c_2 are the learning factors; and r_1,r_2 are the random numbers over the interval (0, 1).

In order to enhance the performance of PSO algorithm and balance the global with the local search of the population, linear decreasing weight (LDW) strategy is adopted to make the inertia weight w change dynamically [35].

```
Input: One dimensional sEMG signal $X$ with the size of $1 \times \text{insize}$
Output: $y$
Initialize learning rate and Loss
for Epoch = 1 to 300 do
    for $k = 1$ to (smaple number of $X$/batch size) do
        Feature extract: $X' \leftarrow \text{Convolution}(X)$
        Training: $y \leftarrow \text{train LWCNN}(X')$
        Min (loss): $\leftarrow \text{Loss} = -\sum_{j=1}^{M} y_{ij} \log(p_{ij})$ (according to equation (4))
        Update the variables by gradient: Adam (learning rate, loss)
    end for $k$
end for Epoch
Obtain the prediction probability $y$
```

Algorithm 1: The proposed lightweight convolution neural network.

Table 2: Hyperparameters setting to be optimized.

Dimensions of each particle	Hyperparameter	Initialization range
1	a	(30, 60)
2	b	(40, 80)
3	c	(80, 120)
4	d	(2, 7)
5	e	(2, 7)
6	f	(2, 7)
7	Activation function of dense-1	ReLU, sigmoid, tanh
8	Activation function of dense-2	ReLU, sigmoid, tanh
9	Batch size	(32, 50)
10	Learning rate	0.1, 0.001

When w is large, the global optimization ability is strong, while the local optimization ability is weak; when w is small, the situation is just the opposite. It can effectively prevent the PSO algorithm from falling into the local optimal solution situation. The formula is

$$w = (w_{\text{ini}} - w_{\text{end}})\left(\frac{G_k - g}{G_k}\right) + w_{\text{end}}, \tag{3}$$

where w_{ini} is the initial inertia weight, w_{end} is the final inertia weight, g is the current number of iterations, and G_k is the maximum number of iterations.

In this paper, the cross-entropy loss function obtained by PLCNN is used as the fitness function, and the formula is

$$\text{Loss} = -\sum{}_{j=1}^{M} y_{ij} \log\left(p_{ij}\right), \tag{4}$$

where M is the number of categories classified; y_{ij} is the variable 0 or 1 (if the j category is the same as the observation sample i, it is 1; otherwise, it is 0); and p_{ij} is the prediction probability that the observation sample i belongs to j category.

The pseudocode of PLCNN is given in Algorithm 2.

The specific implementation of the PLCNN model is shown in Figure 3. In step 1, we initialize the basic parameters of PSO and set the hyperparameters in Table 1 as the position vector of each particle. In step 2, each particle is passed to LWCNN for iterative training, and the cross-entropy loss function value is used as the fitness function value. In step 3, the individual optimal value x_{lp}^{k} and the population optimal value x_{g}^{k} are updated according to the fitness function value. Step 4 is to judge whether iterations have reached the maximum number. If it meets the requirements, step 6 is executed; otherwise, step 5 is executed. Step 5 is to update the velocity vector, position vector, and inertia weight according to (1)–(3). Then, it returns to step 2. Step 6 is to output the optimization results.

3. Data and Experimental

3.1. Data Acquisition. Our approach is evaluated on the Ninapro Database 1 (DB1), which is an open standard data set of multichannel gesture sEMG signals in Sweden. The data is similar to that acquired in real-life conditions, so it is widely used in sEMG pattern recognition [36, 37]. In DB1, the sEMG data from twenty-seven healthy subjects who performed fifty-two hand movements are recorded.

The fifty-two movements are composed of Exercise A, Exercise B, and Exercise C, which are twelve finger movements, seventeen wrist movements, and twenty-three hand grabbing movements, respectively. This motion involves robotics and rehabilitation medicine, covering most of the hand movements encountered in daily life. It represents the sensitivity, flexibility, and dynamics of the hand. Figure 4 shows some example fingers, wrists, and grabbing movements.

During the process of data collection, the subjects are asked to repeat the action of their right hand every three seconds, while each gesture action is repeated ten times with

```
Input: One dimensional sEMG signal $X$ with size of $1 \times insize$
Output: Optimal hyperparameter (particle position $x_g^k$ under minimum fitness value)
Initialize $x_g^k$ according to Table 2
for $k = 1$ to maximum iterations $G_k$ do
  for $l = 1$ to particle number $m$ do
    Calculate the fitness value according to equation (4), save particle current position parameter $x_l^k$
  end for $l$
  According to equation (4): update the best position of each particle $x_{lp}^k$ and the best position of population $x_g^k$
  According to equations (1) and (2): update the particle velocity $v_l^{k+1}$ and position $x_l^{k+1}$
  According to equation (3): update inertia weight $w$
end for $k$
Obtain the optimal hyperparameters $x_g^k$
```

Algorithm 2: Pseudocode of EMG pattern recognition based on PLCNN.

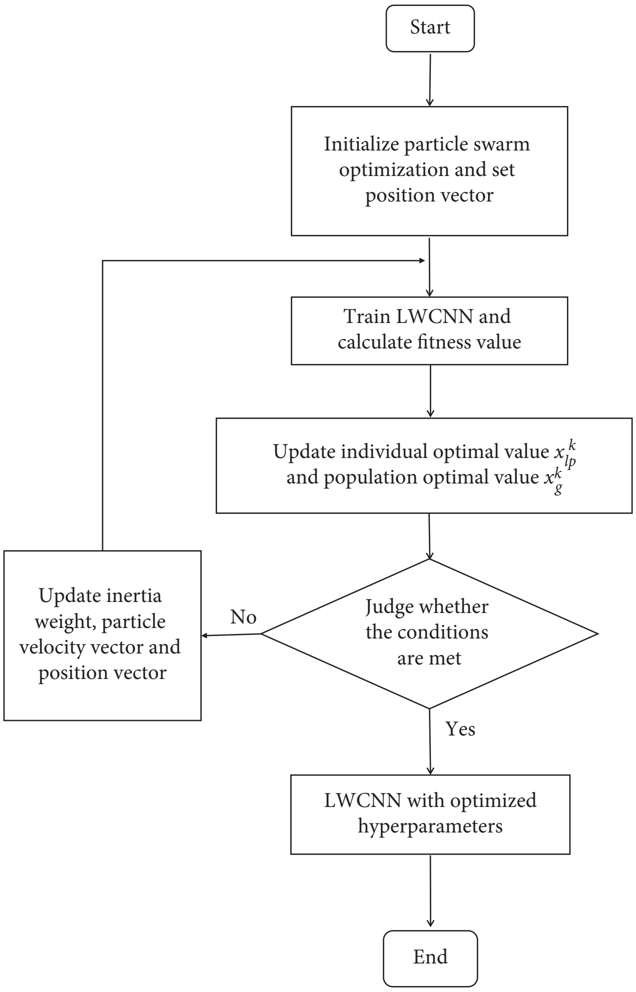

Start
Initialize particle swarm optimization and set position vector
Train LWCNN and calculate fitness value
Update individual optimal value x_{lp}^k and population optimal value x_g^k
Judge whether the conditions are met
No
Update inertia weight, particle velocity vector and position vector
Yes
LWCNN with optimized hyperparameters
End

Figure 3: The flowchart of PLCNN model.

five seconds. Ten Ottobock electrodes are used to collect the data of radial brachial joint, flexor digitorum superficialis, and extensor digitorum superficialis of each subject's forearm at a sampling frequency of 100 Hz. Then the collected data will go through Hampel filter to eliminate 50 Hz power line interference. More details of the data can be found in the official literature [38].

3.2. Preprocessing and Experimental Configuration. We cut out the active sEMG action according to the labels. For each action, the sEMG data of each channel is taken as a sample. We randomly take 90% of the samples as the training set and 10% as the test set. Since the action will not be completely synchronized with the stimulus from the acquisition software with different reaction time and experimental

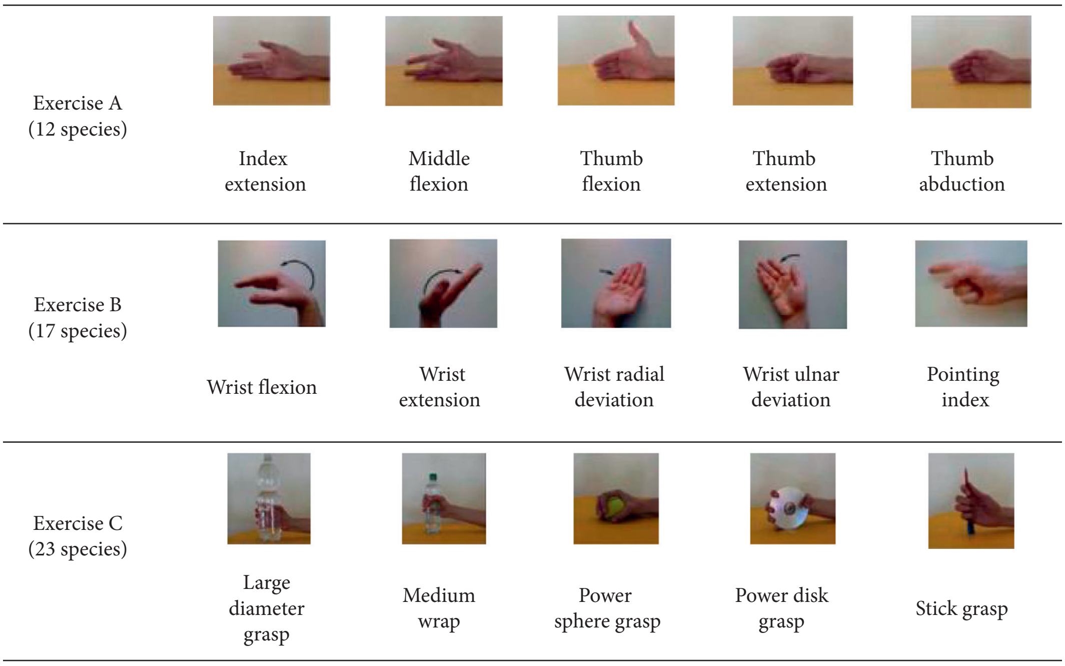


Figure 4: Partial action diagram.

conditions, the generalized likelihood ratio algorithm [39] is used to relabel the data. The relabeled data is added with zero to make up the uniform length and then normalized. The processed signal length insize corresponding to Exercise A, B, and C is 597, 594, and 556, respectively.

We set the number of particles m to 10, the maximum iteration G_k is 30, the learning factor $c_1 = c_2 = 2$ in (1), and initial inertia weight $w_{\text{ini}} = 1$ in (3). As shown in Figure 5, the population has converged after continuous iteration. Accordingly, the optimal hyperparameters found by the PLCNN are shown in Table 3.

In addition, as shown in Table 4, we compare CNN, LWCNN, and PLCNN to verify the high recognition ability of PLCNN. The network structure of PLCNN is simple, and the hyperparameters are obtained by PSO, while the hyperparameters of LWCNN network are set manually. Compared with these network structures, CNN is more complex, and its hyperparameters are also set manually.

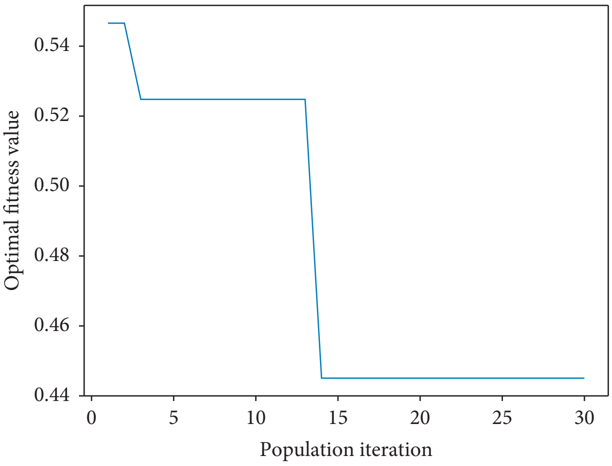


Figure 5: The trend graph of fitness value change.

4. Analysis of Results

Four evaluation indexes, accuracy (5), Kappa coefficient (6), Hamming loss (8), and Jaccard similarity coefficient (9), are used to comprehensively evaluate the performance of different recognition algorithms on DB1 training set.

4.1. Accuracy

$$\text{acc} = \frac{\text{tp} + \text{tn}}{\text{tp} + \text{tn} + \text{fp} + \text{fn}}, \quad (5)$$

where tp is the number of samples correctly classified into the current gesture type, fp is the number of samples that do not belong to the current gesture type but are misclassified into it, tn is the number of samples that belong to the current gesture type but are misclassified into other types, and fn is the number of samples that do not belong to the current gesture type and are misclassified into other types.

4.2. Kappa Coefficient

$$K = \frac{\text{acc} - \text{pe}}{1 - \text{pe}}, \quad (6)$$

where standard pe is usually expressed as follows:

$$\text{pe} = \frac{\sum_{i=1}^{M} a_i \cdot b_i}{N^2}, \quad (7)$$

where M is the number of categories classified, a_i is the actual number of class i samples, b_i is the predicted number of class i samples, and N is the total number of samples. As

Table 3: Optimal hyperparameters setting.

Dimensions of each particle	Hyperparameter	x_g^{30}
1	a	31
2	b	64
3	c	119
4	d	5
5	e	5
6	f	3
7	Activation function of dense-1	ReLU
8	Activation function of dense-2	ReLU
9	Batch size	42
10	Learning rate	0.001

Table 4: settings of CNN, LWCNN, and PLCNN.

Methods	Number of convolution layers	Number of pooling layers	Hyperparameter
CNN	5	1	Manual setting
LWCNN	3	2	Manual setting
PLCNN	3	2	PSO (see Table 3)

shown in Table 5, if the parameter K $(K \in [0,1])$ is larger, the classification accuracy of the model is higher.

Table 5: Classification of Kappa coefficient.

K	0~0.2	0.2~0.4	0.4~0.6	0.6~0.8	0.8~1.0
Consistency level	Low	Common	Secondary	High	Almost the same

4.3. Hamming Loss

$$H = \frac{1}{N}\sum_{i=1}^{M}\frac{\mathrm{XOR}(Y_i, P_i)}{n}, \tag{8}$$

where Y_i is the true value of type i sample, P_i is the predicted value of the class i sample, $\mathrm{XOR}(0,1) = \mathrm{XOR}(1,0) = 1$, and $\mathrm{XOR}(0,0) = \mathrm{XOR}(1,1) = 0$. Obviously, when the Hamming loss H $(H \in [0,1])$ is smaller, the difference between predicted and actual is smaller, and the classification ability of the used algorithm is stronger.

4.4. Jaccard Similarity Coefficient

$$J = \frac{1}{N}\sum_{i=1}^{M}\frac{|Y_i \cap P_i|}{|Y_i \cup P_i|}. \tag{9}$$

For EMG pattern recognition problem, if the Jaccard similarity coefficient J is larger, the prediction results are more consistent with the actual results, and the performance of model action classification is better.

For the traditional machine learning algorithm, the recognition task is generally divided into feature extraction and feature classification. In the experiment, we extract four common EMG signal features: integrated EMG [40], root mean square [41], mean absolute value [42], and waveform length [43]. Then, they are sent to three kinds of feature classifiers: SVM [44], RF [45], and KNN [46]. Moreover, ordinary CNN, LWCNN with manually set hyperparameters, and optimized PLCNN based on deep learning algorithm can directly extract and classify the original EMG signals end-to-end.

The performance test for the proposed PLCNN and other five popular recognition algorithms are described by radar chart based on the four evaluation indexes. Figures 6–8 show the comparison results on the three exercise sets A, B, and C, respectively. It is obvious that the closer the point to the center, the lower the index value.

As can be seen from Figures 6–8, the accuracy of the PLCNN model in Exercise A, B, and C reaches 86.67%, 90.06%, and 89.57%, respectively. This means that our model outperforms the other five algorithms by 19%–38%, 12%–17%, and 4%–7%. Among the three traditional machine learning classifiers, RF has the highest accuracy due to its strong resistance to overfitting and excellent stability. It is difficult for SVM to adapt to the multiclassification problem, and KNN has poor fault tolerance for training data. Compared with the traditional algorithms, deep learning models are more representative and recognizable to extract sEMG features.

In terms of Kappa coefficient, Jaccard similarity coefficient, and Hamming loss, the performance of the three traditional machine learning algorithms is significantly worse than the neural network algorithms. In particular, the Kappa coefficients of PLCNN on Exercise A, B, and C are 0.8545, 0.8938, and 0.8909, respectively. The Jaccard similarity coefficients are 0.7511, 0.8182, and 0.7869, respectively. Compared with the other five algorithms, these two evaluation indicators of PLCNN are the highest. The Hamming loss of PLCNN is 0.1333, 0.1, and 0.1043, respectively, which is the lowest among all the six algorithms. The results illustrate that the prediction results of PLCNN model are accurate, and its classification is precise.

In conclusion, LWCNN and PLCNN can recognize sEMG signals effectively. However, the PLCNN involves

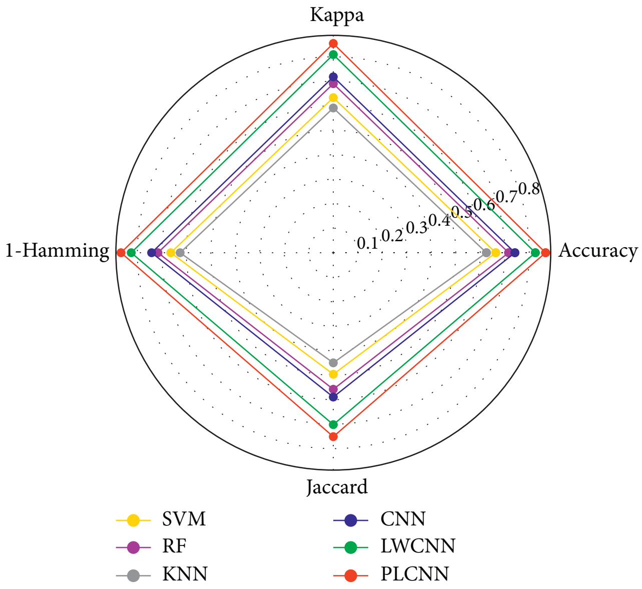


Figure 6: Performance test for Exercise A.

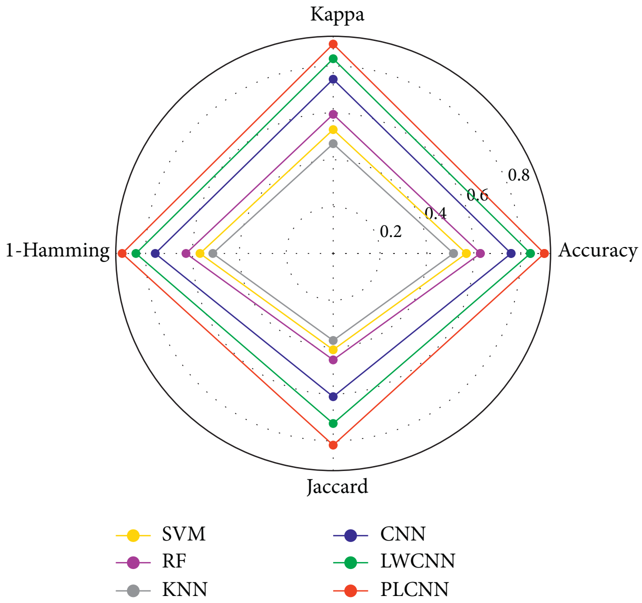


Figure 7: Performance test for Exercise B.

PSO to determine the optimal hyperparameters automatically and achieves better performance than LWCNN which adjusts hyperparameters manually.

The running time represents the complexity of the algorithm and reflects the speed of sEMG pattern recognition. Table 6 shows the average test time of the six algorithms in three exercises and all fifty-two actions. PLCNN takes 5 ms, 7 ms, 9 ms, and 19 ms to recognize twelve, seventeen, twenty-three, and fifty-two actions, respectively. It is faster than the traditional machine learning algorithms, CNN, and LWCNN by 20–400 ms, 3–9 ms, and 1–3 ms. At the same time, the average test time of PLCNN is significantly shorter than that of the paper [25].

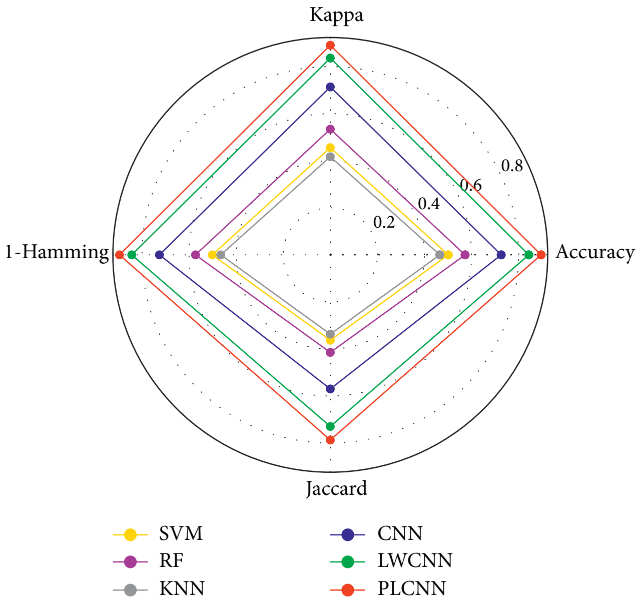


Figure 8: Performance test for Exercise C.

As one of the effective indexes in evaluating and identifying tasks, the confusion matrix map is widely used in various classification networks. As shown in Figures 9–11, in order to further compare the performance of three networks based on deep learning, we take Exercise A as the visual experimental data set and give the corresponding confusion matrix diagram.

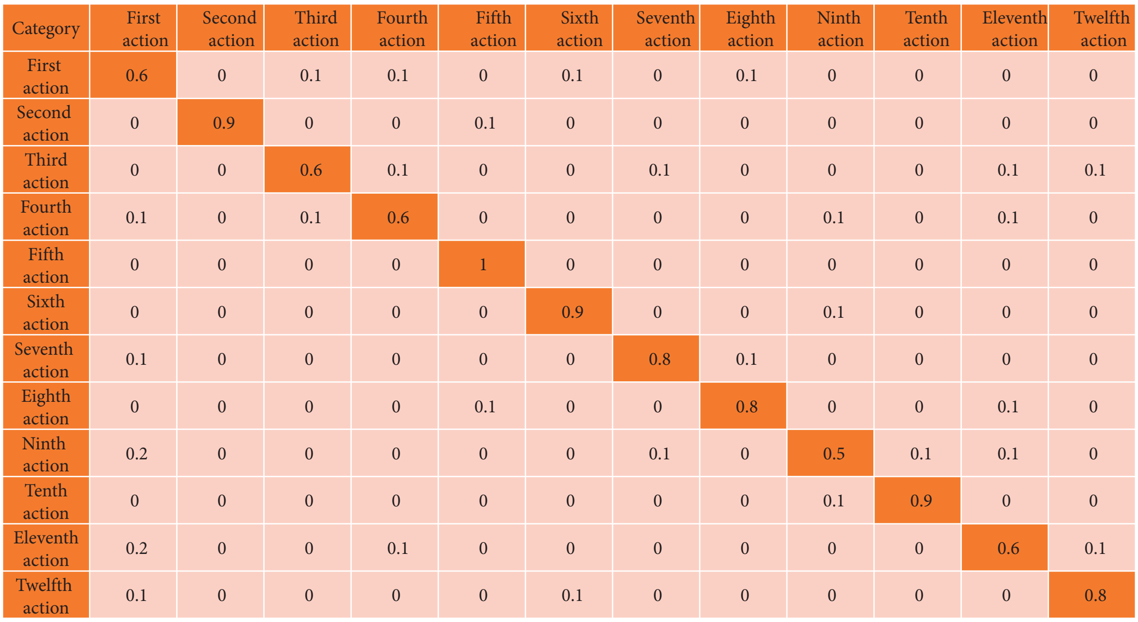
As shown in Figures 9–11, the ordinary CNN model can achieve high accuracy for the classification of the fifth type of action (ring flexion) since the action amplitude is large and easy to be recognized when the ring finger is bent. LWCNN and PLCNN also have good performance in this kind of action recognition. However, the accuracy of CNN for the remaining eleven movements are not very high, especially in the ninth movement classification; it can only reach 50%, which is 30% lower than LWCNN. LWCNN network structure is simpler than CNN, for which it is not easy to saturate and classify sEMG signals accurately. PLCNN has precise recognition ability that divides five actions correctly in 12 movements, and the accuracy rate of PLCNN is higher than that of LWCNN in the ninth category by 10%.

In order to verify the robustness and convergence of the model, the loss value and accuracy of LWCNN and PLCNN are visualized in the process of training and testing. Figures 12(a), 13(a), and 14(a) represent the loss change graph of LWCNN and PLCNN on Exercise A, B, and C, respectively. The corresponding loss values are smoothed by the log function. Figures 12(b), 13(b), and 14(b) show the accuracy variation diagram of LWCNN and PLCNN on Exercise A, B, and C, respectively.

In the training process, the loss values of LWCNN and PLCNN decrease with iteration increasing and effectively converge, as shown in Figures 12(a), 13(a), and 14(a). However, the convergence speed of PLCNN is obviously faster than that of LWCNN, and the global loss value is lower than that of LWCNN. This shows that the optimized

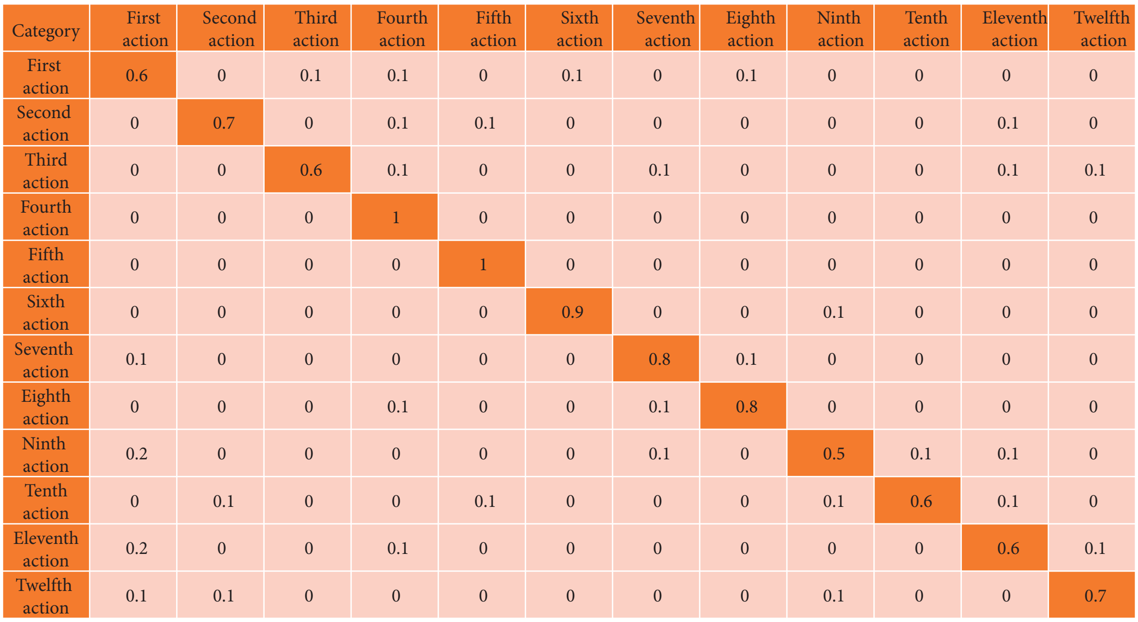
Table 6: Average test time of six algorithms.

Average test time (ms)	SVM	RF	KNN	CNN	LWCNN	PLCNN
Exercise A	26.0057	20.0042	31.0068	8.0020	5.0017	5.0013
Exercise B	69.0145	28.0063	61.0132	10.0023	7.0042	7.0013
Exercise C	154.0343	41.0096	93.0204	14.0037	12.0027	9.9928
All	954.2138	99.0221	255.0597	28.0072	21.0042	19.0043



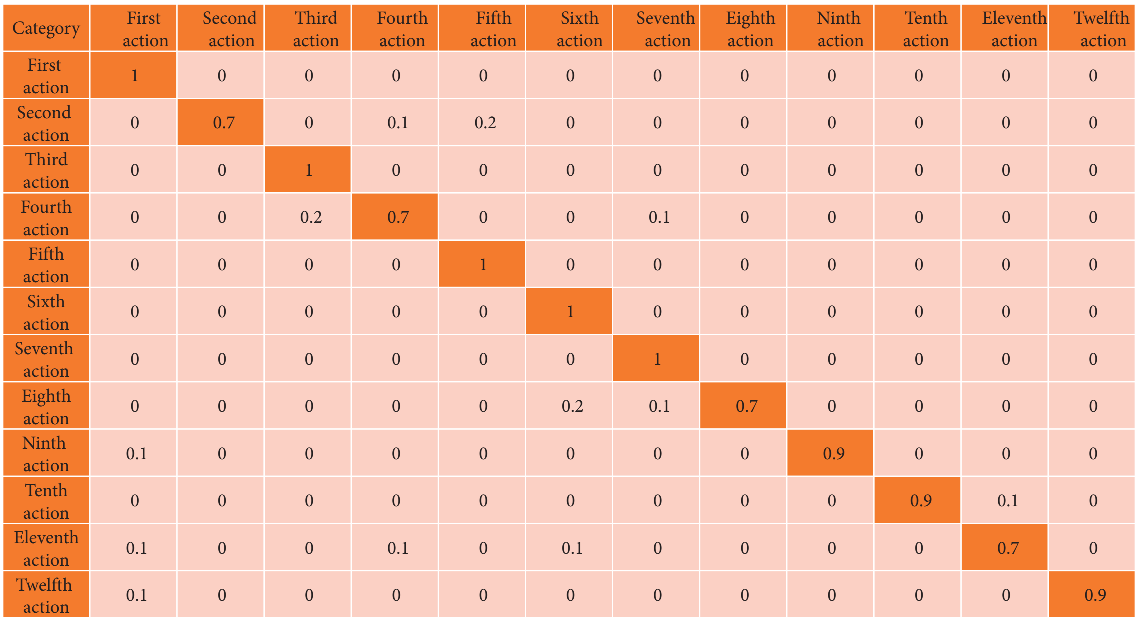
Category	First action	Second action	Third action	Fourth action	Fifth action	Sixth action	Seventh action	Eighth action	Ninth action	Tenth action	Eleventh action	Twelfth action
First action	0.6	0	0.1	0.1	0	0.1	0	0.1	0	0	0	0
Second action	0	0.9	0	0	0.1	0	0	0	0	0	0	0
Third action	0	0	0.6	0.1	0	0	0.1	0	0	0	0.1	0.1
Fourth action	0.1	0	0.1	0.6	0	0	0	0	0.1	0	0.1	0
Fifth action	0	0	0	0	1	0	0	0	0	0	0	0
Sixth action	0	0	0	0	0	0.9	0	0	0.1	0	0	0
Seventh action	0.1	0	0	0	0	0	0.8	0.1	0	0	0	0
Eighth action	0	0	0	0	0.1	0	0	0.8	0	0	0.1	0
Ninth action	0.2	0	0	0	0	0	0.1	0	0.5	0.1	0.1	0
Tenth action	0	0	0	0	0	0	0	0	0.1	0.9	0	0
Eleventh action	0.2	0	0	0.1	0	0	0	0	0	0	0.6	0.1
Twelfth action	0.1	0	0	0	0	0.1	0	0	0	0	0	0.8

Figure 9: Confusion matrix of CNN in Exercise A.



Category	First action	Second action	Third action	Fourth action	Fifth action	Sixth action	Seventh action	Eighth action	Ninth action	Tenth action	Eleventh action	Twelfth action
First action	0.6	0	0.1	0.1	0	0.1	0	0.1	0	0	0	0
Second action	0	0.7	0	0.1	0.1	0	0	0	0	0	0.1	0
Third action	0	0	0.6	0.1	0	0	0.1	0	0	0	0.1	0.1
Fourth action	0	0	0	1	0	0	0	0	0	0	0	0
Fifth action	0	0	0	0	1	0	0	0	0	0	0	0
Sixth action	0	0	0	0	0	0.9	0	0	0.1	0	0	0
Seventh action	0.1	0	0	0	0	0	0.8	0.1	0	0	0	0
Eighth action	0	0	0	0.1	0	0	0.1	0.8	0	0	0	0
Ninth action	0.2	0	0	0	0	0	0.1	0	0.5	0.1	0.1	0
Tenth action	0	0.1	0	0	0.1	0	0	0	0.1	0.6	0.1	0
Eleventh action	0.2	0	0	0.1	0	0	0	0	0	0	0.6	0.1
Twelfth action	0.1	0.1	0	0	0	0	0	0	0.1	0	0	0.7

Figure 10: Confusion matrix of LWCNN in Exercise A.



Category	First action	Second action	Third action	Fourth action	Fifth action	Sixth action	Seventh action	Eighth action	Ninth action	Tenth action	Eleventh action	Twelfth action
First action	1	0	0	0	0	0	0	0	0	0	0	0
Second action	0	0.7	0	0.1	0.2	0	0	0	0	0	0	0
Third action	0	0	1	0	0	0	0	0	0	0	0	0
Fourth action	0	0	0.2	0.7	0	0	0.1	0	0	0	0	0
Fifth action	0	0	0	0	1	0	0	0	0	0	0	0
Sixth action	0	0	0	0	0	1	0	0	0	0	0	0
Seventh action	0	0	0	0	0	0	1	0	0	0	0	0
Eighth action	0	0	0	0	0	0.2	0.1	0.7	0	0	0	0
Ninth action	0.1	0	0	0	0	0	0	0	0.9	0	0	0
Tenth action	0	0	0	0	0	0	0	0	0	0.9	0.1	0
Eleventh action	0.1	0	0	0.1	0	0.1	0	0	0	0	0.7	0
Twelfth action	0.1	0	0	0	0	0	0	0	0	0	0	0.9

Figure 11: Confusion matrix of PLCNN in Exercise A.

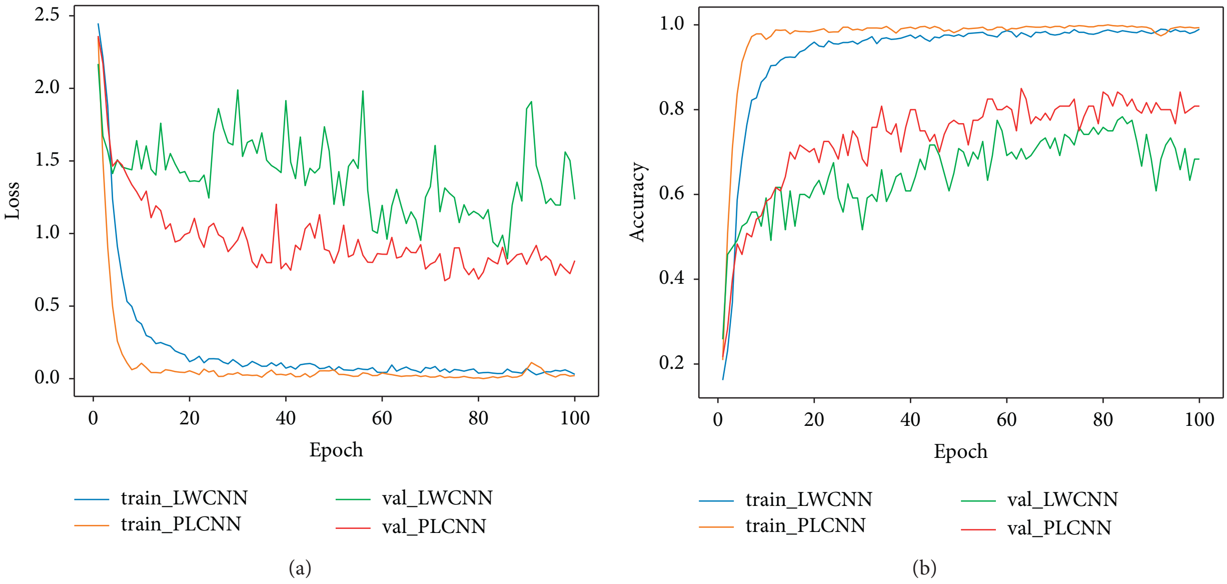


Figure 12: Loss function trend chart and precision change trend chart for Exercise A: (a) loss tend; (b) accuracy trend.

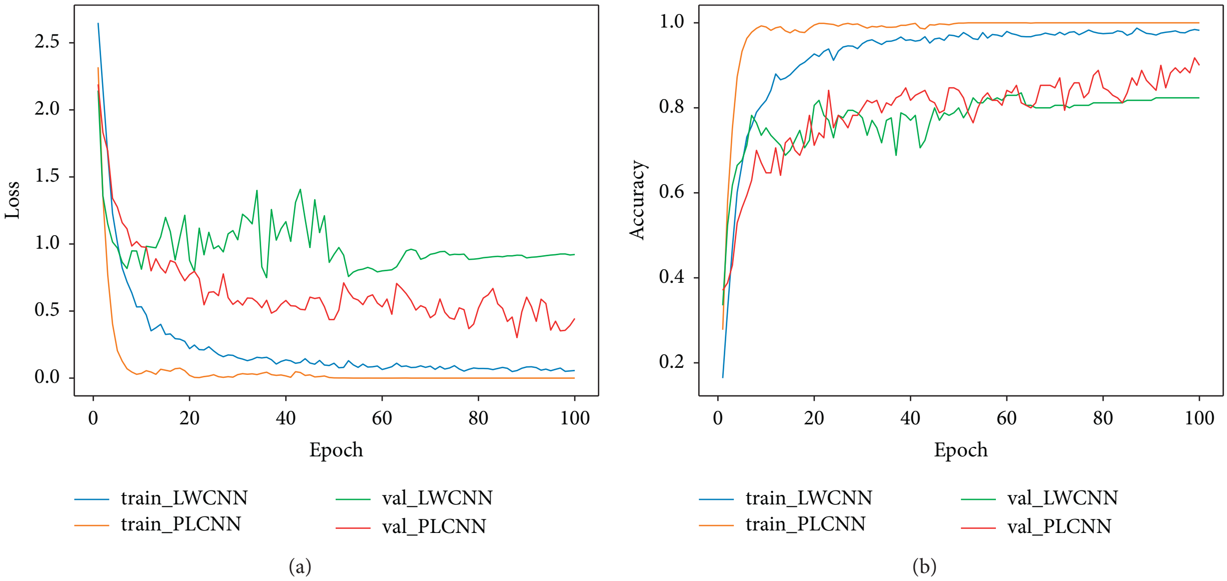


Figure 13: Loss function trend chart and precision change trend chart for Exercise B: (a) loss tend; (b) accuracy trend.

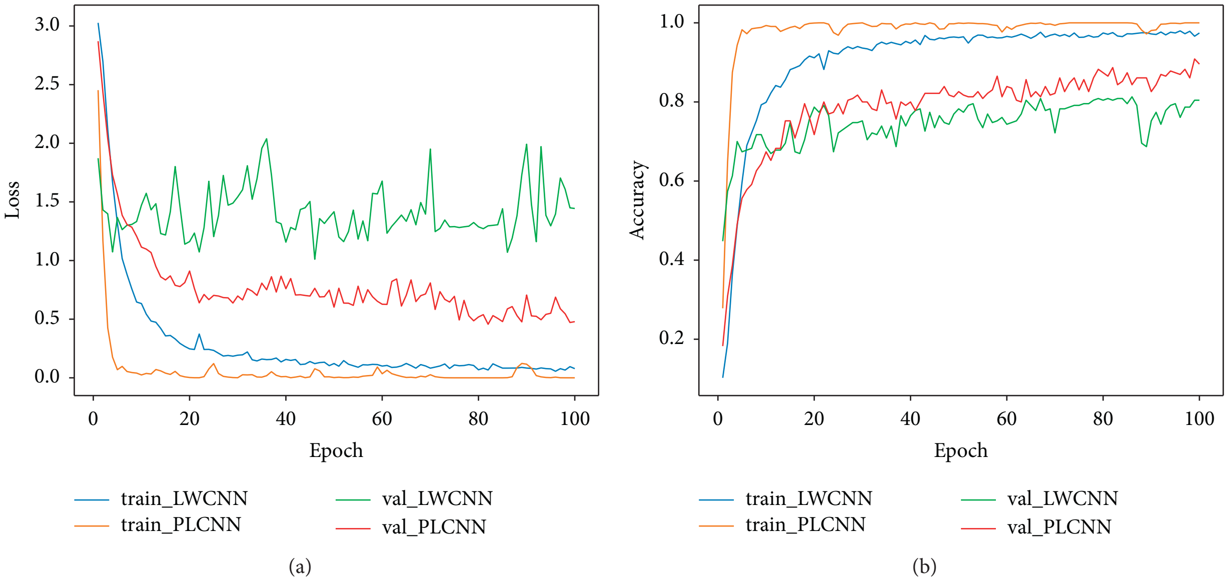


Figure 14: Loss function trend chart and precision change trend chart for Exercise C: (a) loss tend; (b) accuracy trend.

hyperparameters of PLCNN model improve the performance of the model and accuracy of the predicted category. Moreover, the loss on the verification set satisfies the same distribution, and the overall loss value is slightly higher than the training process. These effectively prove the good convergence and stability of the model. Figures 12(b), 13(b), and 14(b) illustrate that the accuracy of training and testing increases with a larger number of iterations. Therefore, the trend of accuracy accords with the corresponding loss distribution, which further demonstrates the robustness of PLCNN model and highlights its high recognition ability in sEMG signals.

5. Conclusion

In this paper, a novel PLCNN method is proposed to recognize EMG signals quickly and accurately. The whole network is compact and simple, including three convolution layers and two full connection layers to extract and classify sEMG features. The PSO algorithm is exploited to search the autoadaptive hyperparameters, which significantly improve the efficiency of the traditional manual debugging hyperparameters. Moreover, the experiments illustrate that the proposed model reduces manual uncertainty and improves recognition performance. Notice that the proposed method obtains recognition accuracy of 86.67%, 90.06%, and 89.57% for Exercise A, B, and C. It is worth mentioning that the experiments in this paper cover almost all human hand movements under Ninapro Database 1, and the proposed method outperforms the five other state-of-the-art methods in terms of speed and accuracy. In the future, we will investigate other evolutionary algorithms [47–49] and machine learning techniques [50] to further improve the recognition rate and recognition speed, and will take the hardware system design into account to research the EMG pattern recognition problems.

Data Availability

The EMG data used to support the findings of this study have been deposited in the Ninapro repository (https://doi.org/2010.1109/BioRob.2012.6290287).

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

The authors would like to gratefully acknowledge the support by the National Natural Science Foundation of China (NSFC) for the research program (grant numbers 61703270, 61803255, 61803254, and 62073071).

References

[1] M. Hallett, W. de Haan, G. Deco et al., “Human brain connectivity: clinical applications for clinical neurophysiology,” *Clinical Neurophysiology*, vol. 131, no. 7, pp. 1621–1651, 2020.

[2] S. Wang, S. Tan, Y. Gao et al., “Learning joint-sparse codes for calibration-free parallel MR imaging,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 1, pp. 251–261, 2018.

[3] O. Oktay, E. Ferrante, K. Kamnitsas et al., “Anatomically constrained neural networks (ACNNs): application to cardiac image enhancement and segmentation,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 2, pp. 384–395, 2018.

[4] Q. Meng, J. Housden, D. Rueckert et al., “Weakly supervised estimation of shadow confidence maps in fetal ultrasound imaging,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 12, pp. 2755–2767, 2019.

[5] G. V. Steenkiste, G. van Loon, and G. Crevecoeur, “Transfer learning in ECG classification from human to horse using a novel parallel neural network architecture,” *Scientific Reports*, vol. 10, no. 1, Article ID 186, 2020.

[6] W. Lukasz, B. Simon J H, C. Marcello et al., “A novel method for electrophysiological analysis of EMG signals using MesaClip,” *Frontiers in Physiology*, vol. 11, Article ID 484, 2020.

[7] A. Maalej, M. Ben-Romdhane, M. Tlili et al., “On the wavelet-based compressibility of continuous-time sampled ECG signal for e-health applications,” *Measurement*, vol. 164, Article ID 108031, 2020.

[8] S. H. Patel, R. P. Carson, L. C. Jordan, and L. M. Pagano, “Rapid ictal transition of focal epilepsy to infantile spasms in neurofibromatosis type 1 captured with EEG,” *Epilepsy & Behavior Reports*, vol. 14, Article ID 100374, 2020.

[9] D. M. Cristiano, R. Simone, S. Mariano et al., “Modular motor control of the sound limb in gait of people with trans-femoral amputation,” *Journal of NeuroEngineering and Rehabilitation*, vol. 16, no. 1, Article ID 132, 2019.

[10] V. H. Cene and A. Balbinot, “Enhancing the classification of hand movements through sEMG signal and non-iterative methods,” *Health and Technology*, vol. 9, no. 4, pp. 561–577, 2019.

[11] T. Przybyła, M. Kotas, and J. Łęski, “On clustering based nonlinear projective filtering of biomedical signals,” *Biomedical Signal Processing and Control*, vol. 44, pp. 237–246, 2018.

[12] A. A. B. Nadzri, S. A. Ahmad, M. H. Marhaban, and H. Jaafar, “Characterization of surface electromyography using time domain features for determining hand motion and stages of contraction,” *Australasian Physical & Engineering Sciences in Medicine*, vol. 37, no. 1, pp. 133–137, 2014.

[13] K. V. Vidya and E. Priya, “Frailty analysis of sEMG signals for different hand movements based on temporal and spectral approach,” *Biomedical Sciences Instrumentation*, vol. 51, pp. 91–98, 2015.

[14] Q. Zhou, Y. Chen, C. Ma, and X. Zheng, “Evaluation of upper limb muscle fatigue based on surface electromyography,” *Science China Life Sciences*, vol. 54, no. 10, pp. 939–944, 2011.

[15] Y. Zhang, P. Xu, P. Li et al., “Noise assisted multivariate empirical mode decomposition for multichannel EMG signals,” *Biomedical Engineering Online*, vol. 16, Article ID 107, 2017.

[16] T. Tuncer, S. Dogan, and A. Subasi, “Surface EMG signal classification using ternary pattern and discrete wavelet transform based feature extraction for hand movement recognition,” *Biomedical Signal Processing and Control*, vol. 58, Article ID 101872, 2020.

[17] O. Fatih and M. Ahmet, “Phasor represented EMG feature extraction against varying contraction level of prosthetic control,” *Biomedical Signal Processing and Control*, vol. 59, Article ID 101881, 2020.

[18] S. A. Abboud, S. Al-Wais, S. H. Abdullah, F. Alnajjar, and A. Al-Jumaily, "Label self-advised support vector machine (LSA-SVM)-automated classification of foot drop rehabilitation case study," *Biosensors-Basel*, vol. 9, no. 4, Article ID 114, 2019.

[19] Y. Cai, Y. Guo, H. Jiang, and M.-C. Huang, "Machine-learning approaches for recognizing muscle activities involved in facial expressions captured by multi-channels surface electromyogram," *Smart Health*, vol. 5-6, pp. 15–25, 2018.

[20] F. Di Nardo, C. Morbidoni, G. Mascia, F. Verdini, and S. Fioretti, "Intra-subject approach for gait-event prediction by neural network interpretation of EMG signals," *Biomedical Engineering Online*, vol. 19, Article ID 58, 2020.

[21] W. Tao, Z. Lai, M. Leu, and Z. Yin, "Worker activity recognition in smart manufacturing using IMU and sEMG signals with convolutional neural networks," in *Proceedings of the 46-th North American Manufacturing Research Conference*, Texas A&M University, College Station, TX, USA, June 2018.

[22] Y. Huang, K. Chen, X. Zhang, K. Wang, and J. Ota, "Joint torque estimation for the human arm from sEMG using backpropagation neural networks and autoencoders," *Biomedical Signal Processing and Control*, vol. 62, Article ID 102051, 2020.

[23] Z. Yang, Y. Chen, J. Wang, and H. Gong, "Recognizing the breathing resistances of wearing respirators from respiratory and sEMG signals with artificial neural networks," *International Journal of Industrial Ergonomics*, vol. 58, pp. 47–54, 2017.

[24] R. Retheep, R. Rejith, and K. S. Sivanandan, "Real time identification of human forearm kinematics from surface EMG signal using artificial neural network models," in *Proceeding of the First Global Colloquium on Recent Advancements and Effectual Researches in Engineering, Science and Technology*, Palai, India, April 2016.

[25] M. Atzori, M. Cognolato, and H. Müller, "Deep learning with convolutional neural networks applied to electromyography data: a resource for the classification of movements for prosthetic hands," *Frontiers in Neurorobotics*, vol. 10, Article ID 9, 2016.

[26] K. Park and S. Lee, "Movement intention decoding based on deep learning for multiuser myoelectric interfaces," in *Proceeding of the 4th International Winter Conference on Brain-Computer Interface*, Seoul, Republic of Korea, February 2016.

[27] K.-T. Kim, C. Guan, and S.-W. Lee, "A subject-transfer framework based on single-trial EMG analysis using convolutional neural networks," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 28, no. 1, pp. 94–103, 2020.

[28] L. Chen, J. Fu, Y. Wu, H. Li, and B. Zheng, "Hand gesture recognition using compact CNN via surface electromyography signals," *Sensors*, vol. 20, no. 3, Article ID 672, 2020.

[29] U. Cote-Allard, C. L. Fall, A. Drouin et al., "Deep learning for electromyographic hand gesture signal classification using transfer learning," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 27, no. 4, pp. 760–771, 2019.

[30] R. Tong, Y. Zhang, H. F. Chen et al., "Learn the temporal-spatial feature of sEMG via dual-flow network," *International Journal of Humanoid Robotics*, vol. 16, no. 4, Article ID 1941004, 2019.

[31] W. Liu, Z. Wang, X. Liu, N. Zeng, and D. Bell, "A novel particle swarm optimization approach for patient clustering from emergency departments," *IEEE Transactions on Evolutionary Computation*, vol. 23, no. 4, pp. 632–644, 2019.

[32] N. Zeng, H. Qiu, Z. Wang, W. Liu, H. Zhang, and Y. Li, "A new switching-delayed-PSO-based optimized SVM algorithm for diagnosis of Alzheimer's disease," *Neurocomputing*, vol. 320, pp. 195–202, 2018.

[33] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceeding of the Third International Conference on Learning Representations*, San Diego, CA, USA, May 2015.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision & Pattern Recognition*, pp. 770–778, Las Vegas, NV, USA, June 2016.

[35] M. A. Jaradat, L. S. Sawaqed, and M. M. Alzgool, "Optimization of PIDD2-FLC for blood glucose level using particle swarm optimization with linearly decreasing weight," *Biomedical Signal Processing and Control*, vol. 59, Article ID 101922, 2020.

[36] W. Geng, Y. Du, W. Jin et al., "Gesture recognition by instantaneous surface EMG images," *Scientific Reports*, vol. 6, Article ID 36571, 2016.

[37] S. Pancholi, P. Jain, A. Varghese, and A. M. Joshi, "A novel time-domain based feature for EMG-PR prosthetic and rehabilitation application," in *Proceedings of the 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pp. 5084–5087, Berlin, Germany, July 2019.

[38] M. Atzori, A. Gijsberts, C. Castellini et al., "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Scientific Data*, vol. 1, Article ID 140053, 2014.

[39] R. I. Jónsson, T. Björgvinsson, M. Blanke, N. K. Poulsen, S. Højsgaard, and L. Munksgaard, "Oestrus detection in dairy cows using likelihood ratio tests," *IFAC Proceedings Volumes*, vol. 41, no. 2, pp. 658–663, 2008.

[40] J. C. Selinger, C. E. Gooyers, J. M. Stevenson, P. A. Costigan, and G. S. Chafe, "Case study: a novel biomechanical approach for evaluating extended body armor systems," *Military Medicine*, vol. 175, no. 11, pp. 871–875, 2010.

[41] B. Hu, X. Zhang, J. Mu, M. Wu, and Y. Wang, "Spasticity assessment based on the Hilbert-Huang transform marginal spectrum entropy and the root mean square of surface electromyography signals: a preliminary study," *Biomedical Engineering Online*, vol. 17, Article ID 27, 2018.

[42] A. A. Al-Taee and A. Al-Jumaily, "Optimal feature set for finger movement classification based on sEMG," in *Proceedings of the 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pp. 5228–5231, Honolulu, HI, USA, 2018.

[43] M. Jochumsen, A. Waris, and E. N. Kamavuako, "The effect of arm position on classification of hand gestures with intramuscular EMG," *Biomedical Signal Processing and Control*, vol. 43, pp. 1–8, 2018.

[44] N. Cristianini and J. S. Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Publishing House of Electronics Industry, Beijing, China, 2004.

[45] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.

[46] R. O. Duda, P. E. Hart, and D. G. Stork, "Pattern Classification," *Journal of Classification*, vol. 24, pp. 305–307, 2007.

[47] X. Luo, Y. Yuan, S. Chen, N. Zeng, and Z. Wang, "Position-transitional particle swarm optimization-incorporated latent factor analysis," *IEEE Transactions on Knowledge and Data Engineering*, p. 1. In press, 2020.

[48] N. Zeng, Z. Wang, H. Zhang, K.-E. Kim, Y. Li, and X. Liu, "An improved particle filter with a novel hybrid proposal distribution for quantitative analysis of gold immunochromatographic strips," *IEEE Transactions on Nanotechnology*, vol. 18, no. 1, pp. 819–829, 2019.

[49] H. Wang, W. Song, E. Zio, A. Kudreyko, and Y. Zhang, "Remaining useful life prediction for lithium-ion batteries using fractional brownian motion and fruit-fly optimization algorithm," *Measurement*, vol. 161, Article ID 107904, 2020.

[50] N. Zeng, Z. Wang, B. Zineddin et al., "Image-based quantitative analysis of gold immunochromatographic strip via cellular neural network approach," *IEEE Transactions on Medical Imaging*, vol. 33, no. 5, pp. 1129–1136, 2014.