MSFF-Net: Multi-Stream Feature Fusion Network for surface electromyography gesture recognition

Xiangdong Peng*, Xiao Zhou, Huaqiang Zhu, Zejun Ke, Congcheng Pan

School of Software and Internet of Things Engineering, Jiangxi University of Finance and Economics, Nanchang, Jiangxi, China

* pengxiangdong@jxufe.edu.cn

Abstract

In the field of surface electromyography (sEMG) gesture recognition, how to improve recognition accuracy has been a research hotspot. The rapid development of deep learning provides a new solution to this problem. At present, the main applications of deep learning for sEMG gesture feature extraction are based on convolutional neural network (CNN) structures to capture spatial morphological information of the multichannel sEMG or based on long short-term memory network (LSTM) to extract time-dependent information of the single-channel sEMG. However, there are few methods to comprehensively consider the distribution area of the sEMG signal acquisition electrode sensor and the arrangement of the sEMG signal morphological features and electrode spatial features. In this paper, a novel multi-stream feature fusion network (MSFF-Net) model is proposed for sEMG gesture recognition. The model adopts a divide-and-conquer strategy to learn the relationship between different muscle regions and specific gestures. Firstly, a multi-stream convolutional neural network (Multi-stream CNN) and a convolutional block attention module integrated with a resblock (ResCBAM) are used to extract multi-dimensional spatial features from signal morphology, electrode space, and feature map space. Then the learned multi-view depth features are fused by a view aggregation network consisting of an early fusion network and a late fusion network. The results of all subjects and gesture movement validation experiments in the sEMG signal acquired from 12 sensors provided by NinaPro’s DB2 and DB4 sub-databases show that the proposed model in this paper has better performance in terms of gesture recognition accuracy compared with the existing models.

1. Introduction

Surface electromyography (sEMG) is a signal graph that uses electrodes to measure muscle electrical activity from the surface of the skin, the recorded signals provide relevant information about human activities. sEMG has important practical value in clinical medicine, ergonomics, and rehabilitation medicine. Human-computer interaction methods based on sEMG are not only widely used in prosthetics [1], sign language recognition systems [2], intelligent
driving [3], virtual reality [4], etc. field. The main forms are wearable devices in the consumer field, and the product form is computer games or drone control by wristbands; auxiliary robotic arms in the industrial field; muscle kinematics analysis equipment, prosthetic hands, and sports exoskeletons in the medical and health field. In addition, sEMG also has considerable potential in assessing fatigue during exercise training [5] as well as in muscle type injury detection [6] and stroke rehabilitation [7].

According to the difference in electrode equipment, sEMG can be divided into sparse sEMG and high-density sEMG [8]. Sparse sEMG forms an image according to the signal amplitude in a certain time window, and the two-dimensional electrode array equipment used by high-density sEMG can directly construct an image according to the amplitude of the instantaneous signal [9]. In the review and research of sparse sEMG pattern recognition algorithm [10, 11], the whole process can be divided into three stages: (1) preprocessing. Remove the noise in the original signal and convert the long-time signal into an adaptive format. (2) Feature extraction. Extract high-level semantic features of time, frequency, time-frequency domain, or deep learning network for intention recognition. (3) Classification or regression. Label or number of forecast tasks.

The existing sparse sEMG pattern recognition methods can be roughly divided into two categories: (1) methods based on feature engineering (2) methods based on feature learning [12].

Feature engineering methods improve information quality and density by designing features such as feature extraction time and frequency and then selecting appropriate classifiers to complete the gesture recognition task. Such as linear discriminant analysis (LDA) [13], principal component analysis (PCA) [14], support vector machine (SVM) [15], random forest (RF) [16], and the k-nearest neighbor algorithm (k-NN) [17, 18]. These methods achieve good results on the recognition task for a small number of gestures, but the recognition rate decreases significantly as the number of recognized gestures increases. Moreover, finding the best feature collection is a very time-consuming task, which requires professional knowledge and experience, and its generalization performance is poor. Constructing a model that can automatically extract sEMG features for classification is the key to improving the recognition rate of multi-gesture classification.

In the feature learning method, features are automatically generated by the machine learning algorithm, therefore, the research focus has shifted from manual feature engineering to automatic feature learning. In the research on improving the accuracy and real-time performance of gesture prediction, Wei et al. [19] combined the traditional feature set with the convolutional neural network (CNN) model based on deep learning, and used multi-view learning to sEMG signal, and achieved good results on the NinaPro database. However, the evaluation and combination experiments of 11 feature sets are more complex, which does not better reflect the advantages of automatic feature extraction by deep learning. Tsinganos et al. [20] made use of the time series characteristics of sEMG and added a time convolution network and attention mechanism and achieved a good recognition effect. However, the model needs the whole sEMG gesture sequence, and the activity duration in real life is uncertain and lacks practicability. Rahimian et al [21] used expanded convolution to classify upper limb gestures, but the one-dimensional convolution network can only extract features along one direction of time or electrodes and did not make full use of the shape of different time signals in sEMG and the information of electrode space. Wei et al. [22] divided the data of different time frames and adjacent time frames into branches to extract the feature changes of different temporal actions but did not further test the effect of joint actions between electrode sensors on the recognition performance.
In summary, there is room to improve the recognition accuracy of deep learning-based methods on sparse EMG signals. Inspired by the studies above, we propose an sEMG gesture recognition model based on a multi-stream feature fusion network (MSFF-Net) on the DB2 and DB4 sub-datasets of the publicly available dataset NinaPro, which focuses on improving the accuracy of gesture recognition. In this model, we enrich the idea of the multi-stream convolution network and focus on the information of different muscle regions with three branches. The model uses a multi-stream convolutional neural network (Multi-stream CNN) and a convolutional block attention module integrated with a resblock (ResCBAM) to alternately extract the morphological features of signals in different periods and the spatial features of the different number of electrode acquisition channels. Then, the early signal features and the deep features after the late multi-stream convolution network are fused in proportion and sent fusion features to the classifier to output the classification results. Experiments show that this method is superior to other existing methods in experimental data processing and recognition accuracy.

The major contributions of this paper are summarized as follows:

1. We propose a method to analyze multi-channel sEMG signals separately according to muscle regions, which reduces the influence of different muscle regions in feature extraction and strengthens the connection of signals in the same region.

2. A novel MSFF-Net model for sEMG gesture recognition is proposed. Combined with the characteristics of the sEMG signal, the model extracts and fuses the features of the sEMG signal from the aspects of signal morphological features, electrode spatial features, and early-late stage feature fusion.

3. We developed an experiment for the proposed model using the sEMG signal obtained by 12 sensors provided by NinaPro’s DB2 and DB4 sub-databases. Compared with similar methods, it has better recognition accuracy.

The rest of this paper is organized as follows. In Section II, we introduce the sEMG signal preprocessing and describe the proposed MSFF-Net in detail. Section III presents the experiment process and the results of the proposed method. Section IV discusses BN layer order effect, early and late network weight effects, ablation studies, and a comparison of similar literature. Section V finally concludes our work.

2. Methods

In this part, we will first introduce the multi-stream feature fusion network overall framework for sEMG gesture classification, then introduce the steps and methods of sEMG signal preprocessing, and finally introduce the composition of the multi-stream feature fusion network in detail.

2.1 The overall framework

The overall frame diagram (Fig 1) shows the complete flow of our proposed multi-stream feature fusion network-based sEMG signal gesture classification. The data used in this article come from Ninapro DB2 and DB4 databases. More than half an hour of raw data per subject in the database needs to be preprocessed to fit the network input, including denoising, action segmentation, data normalization, and fragment EMG generation. The preprocessed data is divided into datasets and trained in the network by cross-validation. At the end of the training, the best model on the validation set is kept for testing on the test set.
2.2 Preprocessing

The sEMG is a nonlinear non-stationary time-series signal that can reflect information related to muscle and body behavior, generated by weak action potentials generated by muscle fibers on the skin surface when the skeletal muscle contracts. Like other physiological electrical signal measurements, they are easily corrupted by noise. Three types of noise appear: power-frequency interference, white Gaussian noise, and baseline wander, making sEMG signals difficult to analyze and having a low signal-to-noise ratio. To better analyze the sEMG signal, preprocessing is required. The preprocessing process mainly includes denoising, action segmentation, normalization, and data segmentation.

2.2.1 Denoising.

The amplitude of the sEMG signal collected by the electrode sensor is usually between 15 and 100 μV, and the energy of the useful signal is mainly distributed between 10 Hz and 500 Hz [23, 24]. At present, the collection equipment on the market has a certain filtering effect. In this paper, the fourth-order Butterworth filter is used for band-pass filtering, the pass-band boundary is 10~500 Hz, and the sEMG signal is simply denoised.

2.2.2 Action segmentation.

The filtered EMG signal retains the main energy part of the signal. The goal of sEMG signal pattern recognition is to identify specific actions that are in demand, and some studies have proposed active segment detection algorithms for signals [25]. For databases without action start times, an algorithm can be used for active segment detection. For the NiproDB2 database with movement activity segments labeled, we segmented each movement of each subject with the resting state as a separation, and we can obtain 49 action categories × 6 repetitions = 294 complete gestures.
2.2.3 Z-Score normalization. The data value of the sEMG signal collected after filtering and motion segmentation is extremely small, and the difference between the data is generally 100 times, which directly affects the experimental results. Normalization algorithms such as Min-Max normalization, Z-score normalization, or conversion to a fixed range are usually used. Our experiments achieve good results on Z-Score normalization. Its mathematical formula is as follows:

\[
y_i = \frac{x_i - \mu}{\sigma}
\]

\(\mu\) is the mean of the population data for a single electrode channel, \(\sigma\) is the standard deviation of the overall data for a single electrode channel. Z-Score normalization was performed on the 12-channel sEMG data in turn. The data of 400 sampling points with a duration of 200ms were selected for comparison.

As shown in Fig 3, the x-axis represents the sampling of different time frames, the y-axis represents different electrode channels, and the z-axis represents the amplitude of the signal. Z-Score normalization normalizes the value of the real signal from \(10^{-4}\) to around 1, this process preserves the same electrode channel signal distribution, reduces the influence of outliers, and concentrates the data into more easily distinguishable intervals.

2.2.4 Fragment EMG generation. After the sEMG signal is normalized, we decompose it into small window segments using a sliding window strategy and an overlapping window scheme to fully utilize the computational power of the system. To compare our proposed method with previous work, we follow the segmentation strategy in the former study [19, 26]. For NinaProDB2, the sliding window length (it is marked as ST in Fig 4) is fixed at 200ms, and
the step window length (it is marked as WT in Fig 4) is set at 50ms. The sliding segmentation process is separately segmented according to the obtained 294 actions to ensure the independence of each action.

The fragment EMG obtained by sliding segmentation is denoted as $h_{R}^{T \times E}$, $T$ is the number of time frames, and $E$ is the number of acquisition electrodes. we take $h_{R}^{400 \times 12}$, that is, 12 electrodes 400 times of sampling data.

2.3 The multi-stream feature fusion network

2.3.1 Network structure. The MSFF-Net model proposed in this paper is used for gesture recognition of EMG signals. The model structure is shown in Fig 5. The fragment EMG can achieve end-to-end action recognition on the input signal after supervised training of the network.

The overall network can be divided into three stages, data input stage, multi-stream convolution stage, and global feature aggregation output stage.

The data entry stage for each fragment EMG is represented as $h_{R}^{T \times E}$. In the experiment, we found that the two-dimensional convolution method of treating the fragment EMG as a single-channel grayscale image is more effective than the one-dimensional convolution of time series. Two-dimensional convolution can be convolved in two directions. Taking Fragment EMG as an example, the convolution along the $T$ direction can obtain the morphological
characteristics of the signal of a single electrode channel, and the convolution along the E direction can obtain different electrode channels and spatial characteristics. Therefore, the dimension enhancement operation is performed on the Fragment EMG, and the new Fragment EMG after the dimension increase is obtained as \( h \in \mathbb{R}^{T \times E \times C} \), \( T \) (Time) can be regarded as the length of the image, \( E \) (Electrode) can be regarded as the width of the image, \( C \) (Channel) is the number of feature channels, and \( C = 1 \) means that the image is a single feature channel. The Fragment EMG after the dimension increase avoids the loss of information caused by the compression of the feature information matrix between different electrodes into vectors during the convolution process.
The new combination of images referenced the location of the electrode acquisition: eight
electrodes were equally spaced around the forearm, two electrodes were placed on the flexor
digitorum and extensor digitorum superficialis, and two electrodes were placed on the biceps
and triceps [27]. We divided the input data into three inputs by electrode distribution
$h_1 \in \mathbb{R}^{400 \times 8 \times 1}, h_2 \in \mathbb{R}^{400 \times 2 \times 1}, h_3 \in \mathbb{R}^{400 \times 2 \times 1}$. Each input stream focuses on the features of different
muscles. The multi-stream convolution stage has three convolutional network branches, corres-
ponding to the three data input streams. That is, different input streams use separate CNN
networks for feature extraction.

The role of the multi-stream convolution stage is to extract high-level semantic features.
Each branch contains batch normalized convolution modules and residual convolution atten-
tion mechanism modules. The batch normalization convolution module is mainly composed
of a convolution layer, a ReLU activation layer, and a batch normalization layer. The residual
convolutional attention module consists of feature channel attention, spatial attention, and
residual modules.

The global feature aggregation output stage consists of two sub-networks: an early fusion
network and a late fusion network. As shown in Fig 5, the early fusion network re-integrates
the three input data streams into the Fragment EMG after the dimension increase. The data
without segmentation and multi-layer convolution retains the early original features and then
increases the number of feature maps through 32 1x1 convolution kernels to obtain the output
$H_{\text{early}}(T = 400, E = 12, C = 32)$. The late fusion network fuses the outputs of the last
layer of the multi-stream convolution, extracts the high-level semantic features of the data, and
obtains the output $H_{\text{late}}(T = 400, E = 12, C = 128)$. Finally, the output of the early
fusion network and the late fusion network is aggregated by the feature channel Concatenation
to obtain the global feature fusion network layer and the output $H_{\text{final}}(T = 400, E = 12, C = 160)$.

The second layer is the global mean pooling layer, which adds and averages the pixel values
of each feature channel, and outputs a neuron for each channel to represent the corresponding
feature map. $H_{\text{final}}$ gets the output $H_{\text{GAP}}(T = 400, C = 160)$ after passing through the
global mean pooling layer.

The third layer is the fully connected layer, which re-assembles the local features into a
complete graph through the weight matrix, and then adds Dropout to prevent overfitting. The
last layer uses the fully connected layer of the Softmax activation function to obtain the final
classification result $H_{\text{out}}(n = \text{number of gesture categories})$. This layer obtains a label vec-
tor with a length equal to the number of gesture categories through the fully connected layer,
and then the Softmax function predicts the category probability distribution of the label vector.
Finally, the result with the highest probability of obtaining a vote is used as the predicted
category.

2.3.2 Batch-normalization convolutional module. In the multi-stream convolution
stage, we adopt two batch-normalization convolutional modules (BConv). Contains convolu-
tional layers, ReLU activation layers, and batch normalization layers (BN).

The Fragment EMG $h \in \mathbb{R}^{T \times E \times C}$ composed of multiple electrode channels can be
regarded as a grayscale image of a single feature channel. In the batch normalization module,
we learn the high-level semantic features hidden by the fragment EMG with 2D convolutional
kernels. For the convolutional layers, we use narrow and long convolution kernels that are
different from the standard size. Taking the first branch as an example, the size of the convolution
kernel is 8x1, which means that each convolution obtains 8 sampling points along the T direc-
tion, and obtains the data of one adjacent electrode sampling channel along the E direction,
that is, within 4ms of a single collection channel. This process separates different electrode
samples, focusing on the morphological characteristics of the signal of a single electrode sample at different periods. The size of the convolution kernel is $1 \times 8$, which means that each convolution obtains 1 sample along the T direction, and obtains the adjacent 8 electrode sampling channel data along the C direction, that is, the sampling data of the 8 electrodes at the same time. This process separates data at different times and links different electrode sampling channels. Each convolution kernel generates a corresponding feature map.

The convolutional data is activated by ReLU to make the network sparse and reduce the interdependence between parameters, which alleviates the occurrence of overfitting. After the data is processed by the BN layer, it is closer to the origin, so that the activation function in the convolution process of the next layer can obtain a larger gradient, and at the same time, the sparsely distributed data after activation is more closely linked. Data that is closely related is more likely to be fit by machine learning features.

2.3.3 CBAM module with residuals. In recent years, the Attention Model has gradually become an important concept in neural networks. By imitating the idea of human visual attention, applied research has been carried out in different application fields. The convolutional block attention module (CBAM) is an improved model based on CNN and attention mechanism, including two sub-modules of channel attention structure (CAS) and spatial attention structure (SAS). It was proposed and applied to image classification in 2018 [28], and the experimental results confirmed that CBAM outperforms all other methods on three different benchmark datasets. It is proved that the CBAM module is of great significance to improve the performance of the recognition model.

The CBAM module with residuals structure is shown in Fig 6(a). From a spatial perspective, channel attention is global, while spatial attention is local. In the experiment, we combined CBAM with ResNet, for the sequential arrangement of the two submodules, the existing experimental results show that the feature channel first is slightly better than the space first [28].
Table 1 shows the detailed network structure of the CBAM module with residuals of the first data input branch.

Considering the particularity of the Fragment EMG $h \in \mathbb{R}^{T \times E \times C}$ compared to the RGB image $h \in \mathbb{R}^{W \times H \times C}$, we made some improvements to CBAM. The feature channel attention part is reserved for extracting information between different feature map channels, and the weights of different feature maps are changed through the feature channel attention module, giving more weight coefficients to useful feature maps, and useless feature maps. It is suppressed to a certain extent, and its structure is shown in Fig 6(b).

We follow the approach of Woo et al. [28] and utilize max pooling and average pooling outputs in the feature channel attention sub-module. First, use the average pooling and max pooling operations to aggregate the spatial information of the feature maps to generate two different feature space descriptors $F_{avg}$ and $F_{max}$. These two feature space descriptors are sent to a shared network to generate our Channel attention $M_c \in \mathbb{R}^{1 \times 1 \times C}$. The shared network consists of a multilayer perceptron (MLP) and a hidden layer. To reduce parameter overhead, the hidden activation size is set to $\mathbb{R}^{1 \times 1 \times C/r}$, where $r$ is the reduction rate. After the shared network is applied to each descriptor, we merge the output feature vectors using element-wise summation. The channel attention is calculated as follows:

$$M_c(F) = \sigma(\text{MLP}(\text{AvgPool}(F)) + \text{MLP}(\text{MaxPool}(F)))$$

$$= \sigma(W_0F_{avg} + W_1F_{max})$$

(2)

where $\sigma$ represents the sigmoid function, $W_0 \in \mathbb{R}^{C/r \times C}$ and $W_1 \in \mathbb{R}^{C/r \times C}$. The two inputs share the MLP weights $W_0$ and $W_1$.

Spatial attention focuses on the location information of information on feature maps, which is complementary to feature channel attention. To compute spatial attention, we apply

---

Table 1. The CBAM module with residuals detailed network structure.

| Name                | Layer          | Output Shape | Activation function | Connected to          |
|---------------------|----------------|--------------|---------------------|-----------------------|
| CBAM module with residuals | Input          | (400,8,64)   | -                   | BConv2D               |
| CAS                 | Maxpool        | (1,1,64)     | -                   | Input                 |
| CAS                 | Avg_pool       | (1,1,64)     | -                   | Input                 |
| CAS                 | Dense          | (1, 1, 32)   | ReLU                | Max_pool (Avg_pool)   |
| CAS                 | Dense_1        | (1, 1, 64)   | ReLU                | Dense (Dense [2])     |
| CAS                 | Add            | (1,1,64)     | -                   | Dense_1               |
| CAS                 | Activation     | (1,1,64)     | Sigmoid             | Add                   |
| SAS                 | Multiply       | (400,8,64)   | -                   | Input CAS             |
| SAS                 | Max_pool_1     | (400,8,1)    | -                   | Multiply              |
| SAS                 | Avg_pool_1     | (400,8,1)    | -                   | Multiply              |
| SAS                 | Concat_1       | (400,8,2)    | -                   | Max_pool_1            |
| SAS                 | Concat_1       | (400,8,1)    | -                   | Avg_pool_1            |
| SAS                 | Conv2d_1       | (400, 8, 1)  | -                   | Conv2d_1              |
| SAS                 | Activation     | (400, 8, 1)  | Sigmoid             | Conv2d_1              |
| SAS                 | Multiply_1     | (400, 8, 64) | -                   | Multiply SAS          |
| SAS                 | Add            | (400,8,64)   | -                   | Multiply_1 Input      |

[2] indicates the second execution.
() indicates that the network layer input is shared.

https://doi.org/10.1371/journal.pone.0276436.t001
average pooling and max pooling operations along the feature channel axis and concatenate the two to generate efficient feature descriptors. Applying pooling operations along the feature channel axis can effectively highlight informative regions [29]. On the concatenated feature descriptor, we use convolution to generate the spatial attention map \( M_s(F) \in \mathbb{R}^{H \times W} \). The spatial attention structure is shown in Fig 6(c). The detailed operations are as follows: First, generate two two-dimensional maps through two merging operations to aggregate the channel information of a feature map: \( F_{avg}^s \in \mathbb{R}^{H \times W \times 1} \) and \( F_{max}^s \in \mathbb{R}^{H \times W \times 1} \). They represent the mean pooling features and max-pooling features in the feature channel, respectively. Then, they are connected and convolved through a standard convolutional layer to generate a 2D spatial attention map, calculated as follows.

\[
M_s(F) = \sigma(f_{a \times a}([\text{AvgPool}(F); \text{MaxPool}(F)]))
\]

\[
= \sigma(f_{a \times a}([F_{avg}^s; F_{max}^s]))
\]

(3)

Where \( \sigma \) represents the sigmoid function, \( f_{a \times a} \) represents the convolution kernel size \( a \times a \) is the convolution process.

We changed the size of the \( a \times a \) convolution kernel in the spatial attention part based on image processing, using a \( 2 \times 1 \) convolution kernel to extract the morphological features of adjacent temporal signals, and a \( 1 \times 2 \) convolution kernel is used to extract the spatial features of adjacent electrode acquisition channels. This process corresponds to the Conv2D network layer of the SAS module in Table 2. Additionally, we replace max and mean pooling in this process with global max pooling, global mean pooling, and reshape operations.

3. Experiments and results

3.1 Database

In this study, NinaPro databases DB2 and DB4, which contain tasks related to upper extremity movement, were used for the experiments. Table 2 summarizes the information and descriptions of the two databases.

NinaPro is a publicly accessible database that has previously been used for myoelectric interface implementations to decode human hand movements. The DB2 sub-database collected sparse sEMG data from 40 healthy subjects including 11 females and 29 males using 12 Delsys wireless electrodes on the subject’s forearm surface and filtered through a Hampel filter to eliminate 50 Hz power frequency interference [27]. The DB4 sub-database collected sparse sEMG data from 10 healthy subjects including 6 males and 4 females using 12 Cometa wireless electrodes on the subject’s forearm surface [30].

Both databases follow the same experimental acquisition protocol, and the captured gesture actions are divided into 3 exercises, the detailed actions of which are shown in Fig 7. Each

| Table 2. The Ninapro database summary table. |
|---------------------------------------------|
| Number of classified gestures | NinaproDB2 | NinaproDB4 |
| Intact subjects | 49 | 52 |
| Electrodes | 12 Delsys | 12 Cometa |
| Sampling rate | 2000Hz | 2000Hz |
| Number of trials | 6 | 6 |
| Exercise1 | Exercise B | Exercise A |
| Exercise2 | Exercise C | Exercise B |
| Exercise3 | Exercise D | Exercise C |

https://doi.org/10.1371/journal.pone.0276436.t002
movement was repeated 6 times, with each exercise lasting 5 s and alternating with a resting position lasting 3 s. The biggest difference between the two is that the data collected by DB2 is at the microvolt level (μV), and the data collected by DB4 is at the volt level (V) after amplification.

It is worth mentioning that the Ninapro database has two sets of movement classification labels and corresponding movement repetition labels; the stimulus records the labels generated for each sample using the stimulus generator; the restimulus records the posterior labels of the

Fig 7. The Ninapro database gesture action chart.

https://doi.org/10.1371/journal.pone.0276436.g007
movements. The processes associated with movement durations in the posterior labels are refined to represent real movements [31]. It is shown in Fig 8.

The repetition is temporally synchronized with the stimulus recordings, recording the number of repetitions and the duration of each action, and can be used to do segmentation of the active segment of the signal. We experimentally did both kinds of data separately considering the differences in labeling, which is something that has been rarely mentioned by others except Rahimian et al [21].

For dataset partitioning, we follow the dataset partitioning strategy in the former study [19, 26]. After action segmentation, according to the number of repetitions of each action, the 1st, 3rd, 4th, and 6th repetitions are used as the training set, and the 2nd and 5th repetitions are used as the test set.

Due to the small number of complete gestures in each category in the Ninapro database and the sparsity of EMG signals, overfitting is easy occurs. Furthermore, the amplitude and duration of sEMG cannot be fully replicated during motion repetition acquisition, we time-warped the training set data [32] and expanded the training data to twice the original size. The time-warped data enhancement comparison chart is shown in Fig 9.

We take the data of 500 samples as an example. The time-warped data randomly changes the timeline of the original data. The degree of time-warping is controlled by the number of speed changes and the ratio of the maximum/minimum speed. The warped data retains the difference between different channels. real-time and signal amplitude. We take the data of 500 samples as an example. The time-warped data randomly changes the timeline of the original data. The degree of time warping is controlled by the number of speed changes and the ratio of the maximum/minimum speed. The time-warped data preserve the synchronization and amplitude between the different electrodes.

3.2 Evaluation metric

We adopted the same intrasubject schemes as those were most commonly used in existing studies on the NinaPro database [9, 33, 34]. In intrasubject evaluation, the deep learning model is trained on a part of the data from one subject and tested on the non-overlapping part of the data from the same subject. We follow this evaluation scheme, specifically, we used
repetitions 1, 3, 4, and 6 for training purposes and repetitions 2 and 5 for testing. The final gesture recognition accuracy is obtained by averaging the accuracy achieved by all subjects.

Classification Accuracy: Classification accuracy is defined as the ratio between the number of correctly classified gesture segments in a trial and the total number of gesture segments tested. The Accuracy (Acc) of the target object is calculated as follows:

\[
Acc = \frac{\text{Number of correct classifications}}{\text{Total number of test samples}}
\]  

(4)

Overall classification accuracy: Overall classification accuracy (Overall Accuracy, OA) is defined as the average of the classification accuracies of all experimental individuals and is calculated as follows:

\[
OA = \frac{1}{M} \sum_{n=1}^{M} Acc
\]  

(5)

where M is the number of subjects.

3.3 Experimental setup

The network proposed in this paper is implemented based on Keras with Tensorflow as the backend and is trained using RTX2080ti. The loss function uses the cross-entropy function, and the model is trained using the Adam optimization algorithm. The number of training sessions is set to 50 epochs, and the learning rate is set to 0.001. During the training process, the model with the highest validation set Accuracy will be saved as the final model.

3.4 Experimental results

Our proposed method follows the dataset partition of the former study. To compare with more researchers, we use the repeated gesture cross-validation method to conduct experiments on the stimulus data and the restimulus data respectively, and the result is taken as the average.
of the overall recognition accuracy in multiple experiments. As shown in Table 3. The cross-validation results can test the generalization ability of the network.

The results in Table 3 show that our network can achieve better classification results when there are differences in data annotation. Unexpectedly, the recognition accuracy of the stimulus label data is higher than that of the restimulus label data. The reason is that although the restimulus label is more in line with human motion during the data labeling process, the actual labeling time of some gestures is much higher or lower than the standard test time, which makes the data imbalance between gesture actions and affects the result. The same result was confirmed in Rahimian et al. [21].

To compare the differences between different subjects, we show the average recognition accuracy of 49 categories of actions for 40 subjects during four cross-validation processes on the DB2 database. As shown in Fig 10.

The smaller box of the subject cross-validation results indicates that our network can generalize to the recognition of repeated gestures from the same subject. To further analyze the reasons for the discrepancy in recognition accuracy, we combine the attributes already given by the database (Table 4) and take 85% accuracy as the baseline. Marks higher than the baseline were marked as high, otherwise marked as low.

Finally, a bivariate graph is drawn according to the subject attribute information, as shown in Fig 11. It shows the number distribution of subjects whose ACC was marked as high and low on attributes such as Laterality, Gender, Age, Height, and Weight, and the accuracy shows significant differences in gender and preference hands.

Our inference is that the apparent difference in gender attribute accuracy may be because females are generally inferior to males in the intensity of action stimuli. Differences in the accuracy of the laterality attribute may be influenced by minority subjects and by motor differences between left-handed and right-handed individuals.
To compare the difference in the accuracy of different action recognition, we randomly show the accuracy of 49 categories of action recognition in a cross-validation experiment for 1 subject in DB2. As shown in Fig 12, among actions with lower than average recognition accuracy, types 9, 10, and 11 are wrist rotations with high similarity. Types 18 and 22 are grasping plastic bottles of different sizes. Types 32, 33, and 35 are grasping small objects. In general, these movements have the characteristics of small stimulation and similar movements. It is difficult to further improve the recognition rate by analyzing only the sEMG collected by the arm. The Ninapro database also records electrode gloves and triaxial acceleration values. The research [35] combining the signals collected by different devices is an effective method to solve this problem.

To test the effectiveness of our proposed method in more databases, we also conducted the same experiment on 10 subjects in the DB4 database, and the cross-validation results of repeated gestures four times on the training set are shown in Fig 13. The cross-validation results verify the generalization ability of the proposed method on the DB4 database, and Table 3 presents the average results of four experiments. Comparing the acquisition methods of DB2 and DB4 databases, the difference in data level between the two is eliminated by Z-score standardization, because the electrode distribution and acquisition protocol are the same, the number of classified gestures is similar, and the final overall classification results are also similar, which meets the experimental expectations.

4. Discussion

In our proposed EMG gesture recognition network, the standardization of input data, the number and size of convolution kernels in the multi-stream convolution stage, the dropout
rate in the early and late aggregation stages, and the weights of early fusion and late fusion all affect the final recognition accuracy. To determine the ideal parameter settings, we selected the raw label data of 3 subjects in the DB2 database as the base data and analyzed the effects using the same experimental settings. Where S1 is male, S26 is left-handed, and S38 is female. The subject’s classification Accuracy (Acc) was used as the evaluation metric.

Fig 11. DB2 subject identification accuracy and information bivariate plot.
https://doi.org/10.1371/journal.pone.0276436.g011

Fig 12. The recognition accuracy of 49 classes of movements of subjects.
https://doi.org/10.1371/journal.pone.0276436.g012
4.1 BConv module settings effects

The general batch normalized convolution process order is a convolutional layer, BN layer, and activation function [36]. In our experiments, we found that the BN layer performs better after the ReLU activation function. The experimental results are shown in Table 5.

Analyzing the experimental results, we believe that some of the output of the features after convolution may be negative, and these features will be truncated by ReLU nonlinearity. If normalization is done before ReLU activation, including negative values in the normalization before culling from the feature space brings a suppressive effect. The activated BN will normalize the positive features and pass them to the lower convolution, without counting the unpassable features.

4.2 Early and late network weight effects

In our proposed EMG gesture recognition network, the global feature aggregation output stage performs feature-level fusion on the outputs of the two sub-networks of early fusion and late fusion. We obtained the results after multiple experiments as shown in Table 6.

4.3 Ablation studies

To better analyze the source of our results, we refer to the method of ablation experiments [37] to analyze the variables of the proposed model layer by layer. The results of the experiments (E1-E7) are shown in Table 7 below.

Table 5. The BN layer sequence experiment.

| BConv module settings       | Acc (%) |
|-----------------------------|---------|
|                            | S1      | S26     | S38     |
| Conv + BN + ReLU           | 75.31   | 76.79   | 81.41   |
| Conv+ ReLU +BN             | 87.57   | 87.31   | 89.41   |

The average gesture recognition accuracy of DB4 subjects.

Fig 13. The average gesture recognition accuracy of DB4 subjects.

https://doi.org/10.1371/journal.pone.0276436.g013
It can be seen from Table 7 that BConv has the greatest improvement in recognition accuracy, which is since batch normalization reduces the effect of internal covariate bias and normalizes the increasingly biased output distribution to the ideal range, allowing the activation input values to fall in sensitive regions, avoiding gradient disappearance while speeding up the training speed. In experiments 2, 3, and 4, we tested the scheme of all sensors as one input, the scheme of separate branches for each electrode sensor, and the scheme of dividing branches according to sensor distribution. Overall, our method is optimal, because the multi-stream convolution process adopts the divide-and-conquer idea for extracting the features of different muscles independently and prevents the interference of irrelevant muscles. Experiment 5 shows that the added ResCBAM sequentially derives the attention map along two independent dimensions of channel and space, and performs adaptive feature extraction, the new feature and the original feature increase the feature amount by adding, and the effect is improved. Experiment 6 shows that the added early and late feature aggregation network extracts early original features and late high-level semantic features, which further improves the recognition rate. Finally, time warping is added for data enhancement, which makes the data more diverse, prevents overfitting, and achieves the highest overall average recognition accuracy.

The multi-stream feature fusion network proposed in this paper uses multi-stream convolution and spatial attention, and according to the characteristics of multi-channel EMG signals, adaptively extracts the morphological features of the signal of a single electrode acquisition channel and the spatial features of multiple electrode acquisition channels at the same time. The feature channel attention mechanism is used to adaptively assign different weights to different feature maps, increase the weight of effective feature maps, suppress invalid feature maps, and obtain more detailed features. Finally, an aggregation network is used to aggregate early primitive features and late high-level semantic features to output classification results. The average recognition accuracy of 49 types of gestures on the stimulus label data reaches 87.02%, which brings optimization from the classification performance and data requirements.

### Table 6. The influence of early and late network weights on classification performance.

| Early and late network weight | Acc (%) |
|------------------------------|---------|
|                             | S1      | S26     | S38     |
| \(H_{\text{early}} = 0, H_{\text{late}} = 128\) | 87.36   | 86.47   | 88.95   |
| \(H_{\text{early}} = 32, H_{\text{late}} = 128\) | 87.57   | 87.31   | 89.41   |
| \(H_{\text{early}} = 64, H_{\text{late}} = 128\) | 85.29   | 87.62   | 89.26   |
| \(H_{\text{early}} = 128, H_{\text{late}} = 128\) | 88.31   | 86.47   | 88.72   |

https://doi.org/10.1371/journal.pone.0276436.t006

### Table 7. The classification and performance comparison of Ablation studies.

| Methods                        | Acc (%) |
|--------------------------------|---------|
| E1: CNN                        | 79.19   | 78.20   | 82.11   |
| E2: BConv                      | 85.89   | 85.28   | 87.83   |
| E3: 12-stream BConv            | 86.80   | 84.55   | 88.69   |
| E4: Multi-stream BConv         | 87.41   | 86.07   | 88.76   |
| E5: E4+ResCBAM                 | 87.36   | 86.47   | 88.95   |
| E6: E5+ Early and late feature aggregation | 87.57   | 87.31   | 89.41   |
| E7: E6+ Time-warped data enhancement | **89.52** | **87.76** | **89.34** |

https://doi.org/10.1371/journal.pone.0276436.t007
4.4 Comparison of similar literature

We compare the recognition model in this paper with gesture recognition models that have been studied on the Ninapro database in recent years in Table 8, and our method excels in both the number of gesture classifications and the overall recognition accuracy (OA).

Among them, Hu [26], Ding [38], Gulati [33], and others only focus on extracting the morphological or temporal features of the signal in the sliding window, ignoring the differences and connections between different electrode channels. The artificial features designed by Tosin [40], Kim [39], Wei [19], and others have high requirements on researchers’ experience, and to some extent destroy the hidden connections between real signals.

Compared to these studies using the Ninapro database, our network does not need a professional artificial feature set design, directly uses the original EMG signal as input, and uses four-time cross-validation of repeated gestures on the training set to make the results more reliable. But nearly a quarter of the real data in the cross-validation process is not directly involved in network training, so we enrich the training set with time-warped data augmentation. We excluded rest states in the number of classified gestures because most of the existing studies in the Ninapro database [26, 32, 35–37] excluded rest, which is the easiest to distinguish and accounts for half of the total number of movements, but not the focus of the study. In addition, restimulus recordings marked some missing EMG data as rest, and these erroneously labeled data could not be used to examine network performance. Finally, good classification results are obtained on differently labeled data and different databases. Compared with similar methods, this method has better recognition performance.

In these studies using other databases, Compared to [20], the input of MSFF-Net is not the whole gesture, it can recognize actions with large differences in duration. Compared to [21], the BConv stage uses a 2D narrow convolution kernel to extract the time and feature information of different electrode sensor channels, respectively. Compared to [22], MSFF-Net discusses the advantages of sensor shunting by region over input shunting and individual shunting with ablation experiments.

5. Conclusion

In this paper, a novel MSFF-Net sEMG signal gesture recognition model is proposed. The biggest feature of this model is that the signals collected in different areas are analyzed separately according to the electrode position, which can more fully extract the features of the multi-
channel sEMG signal. In the signal preprocessing stage of this method, the denoised sEMG signal is segmented to ensure the independence of each action, and then the order of magnitude of the signal is changed by Z-score standardization to facilitate feature extraction. Finally, sliding segmentation is performed on the action segment to identify the signal from the perspective of the image. In the feature extraction stage, the multi-stream convolution network isolates the signal interference of different muscle regions and retains the signal correlation of the same muscle region. ResCBAM module further extracts deep features from signal shape, electrode acquisition space, and feature space, and the early-late aggregation network integrates the original features after signal standardization and the high-level semantic features of multi-stream convolution. Experiments show that the average recognition accuracy of the proposed model for 49 types of gestures of 40 healthy subjects in NinaPro’s DB2 database and 52 types of gestures of 10 healthy subjects in the DB4 database is better than the existing similar methods. The proposed network model helps to improve the accuracy of gesture recognition based on sEMG and provides a new idea for the current research of human–computer interaction based on sEMG. In the process of motion acquisition of sEMG signal, the phenomenon of sensor dislocation or movement is inevitable, which may affect the performance of the reference electrode arrangement classification method proposed in this paper. In the follow-up research, we will further try to solve this problem by combining feature set images and signal images.

Author Contributions

Data curation: Xiao Zhou.

Formal analysis: Xiangdong Peng, Zejun Ke.

Investigation: Xiangdong Peng, Huaqiang Zhu, Congcheng Pan.

Methodology: Xiangdong Peng, Xiao Zhou.

Software: Xiao Zhou, Congcheng Pan.

Supervision: Huaqiang Zhu, Zejun Ke.

Validation: Xiao Zhou, Huaqiang Zhu, Zejun Ke.

Visualization: Congcheng Pan.

Writing – original draft: Xiao Zhou.

Writing – review & editing: Xiangdong Peng.

References

1. Sharma T, Sharma K P, Veer K. Decomposition and evaluation of SEMG for hand prostheses control [J]. Measurement, 2021, 186: 110102.

2. Pereira-Montiel E, Pérez-Giraldo E, Mazo J, et al. Automatic sign language recognition based on accelerometry and surface electromyography signals: A study for Colombian sign language [J]. Biomedical Signal Processing and Control, 2022, 71: 103201.

3. Nacpil EJC, Wang Z, Yan Z, et al. Surface Electromyography-Controlled Pedestrian Collision Avoidance: A Driving Simulator Study[J]. IEEE Sensors Journal, 2021, 21(12): 13877–13885.

4. Sugiarto T, Hsu CL, Sun CT, et al. Surface EMG vs. High-Density EMG: Tradeoff Between Performance and Usability for Head Orientation Prediction in VR Application[J]. IEEE Access, 2021, 9: 45418–45427.

5. Elshafei M, Shihab E. Towards detecting biceps muscle fatigue in gym activity using wearables[J]. Sensors, 2021, 21(3): 759. https://doi.org/10.3390/s21030759 PMID: 33498702
6. Sturma A, Hruby L A, Prahm C, et al. Rehabilitation of upper extremity nerve injuries using surface EMG biofeedback: protocols for clinical application[J]. Frontiers in neuroscience, 2018, 12: 906.

7. Potočnik B, Divjak M, Urb F, et al. Estimation of muscle co-activations in wrist rehabilitation after stroke is sensitive to motor unit distribution and action potential shapes[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, 28(5): 1208–1215.

8. Du Y, Jin W, Wei W, et al. SEMG-based inter-session gesture recognition enhanced by deep domain adaptation[J]. Sensors, 2017, 17(3): 458.

9. Geng W, Du Y, Jin W, et al. Gesture recognition by instantaneous sEMG images[J]. Scientific reports, 2016, 6(1): 1–8.

10. Simao M, Mendes N, Gibaru O, et al. A review on electromyography decoding and pattern recognition for human-machine interaction[J]. IEEE Access, 2019, 7: 39564–39582.

11. Xiong D, Zhang D, Zhao X, et al. Deep learning for EMG-based human-machine interaction: A review [J]. IEEE/CAA Journal of Automatica Sinica, 2021, 8(3): 512–533.

12. Phinyomark A, Scheme E. EMG pattern recognition in the era of big data and deep learning[J]. Big Data and Cognitive Computing, 2018, 2(3): 21.

13. Liang S, Chen J, Wu Y, et al. Recognition of Subtle Gestures by 2-Channel sEMG Using Parameter Estimation Classifiers Based on Probability Density[J]. IEEE Access, 2020, 8: 169835–169850.

14. Qi J, Jiang G, Li G, et al. SEMG hand gesture recognition system based on PCA and GRNN[J]. IEEE Transactions on Neural Systems and Rehabilitation Engineering, 2020, 32(10): 6343–6351.

15. Narayanan A, Ahlawat V, Kumar S. Pattern recognition of sEMG signals using DWT based feature and SVM Classifier[J]. Pattern recognition, 2020, 29(10S): 2243–2256.

16. Li Z, Guan X, Zou K, et al. Estimation of knee movement from sEMG using random forest with principal component analysis[J]. Electronics, 2020, 9(1): 43.

17. Bergell O, Oral C, Ergul EU. Efficient Hand Movement Detection Using k-Means Clustering and k-Near-est Neighbor Algorithms[J]. Journal of Medical and Biological Engineering, 2021, 41: 11–24.

18. Wei C, Wang H, Lu Y, et al. Recognition of lower limb movements using empirical mode decomposition and k-nearest neighbor entropy estimator with surface electromyogram signals[J]. Biomedical Signal Processing and Control, 2022, 71: 103198.

19. Wei W, Dai Q, Wong Y, et al. Surface-electromyography-based gesture recognition by multi-view deep learning[J]. IEEE Transactions on Biomedical Engineering, 2019, 66(10): 2964–2973.

20. Tsinganos P, Cornelis B, Cornelis J, et al. Improved gesture recognition based on sEMG signals and TCN[C]//ICASSP 2019–2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019: 1169–1173.

21. Rahimian E, Zabihi S, Atashzar SF, et al. SEMG-based hand gesture recognition via hybrid and dilated deep neural network architectures for neurorobotic prostheses[J]. Journal of Medical Robotics Research, 2020, 5(01n02): 2041001.

22. Wei W, Wong Y, Du Y, et al. A Multi-stream Convolutional Neural Network for sEMG-based Gesture Recognition in Muscle-Computer Interface[J]. Pattern Recognition Letters, 2017: S0167865517304439.

23. Zhang A, Niu Y, Gao Y, et al. Second-Order Information Bottleneck based Spiking Neural Networks for sEMG Recognition[J]. Information Sciences, 2021.

24. Xi X, Zhang Y, Zhao Y, et al. Denoising of surface electromyogram based on complementary ensemble empirical mode decomposition and improved interval thresholding[J]. Review of Scientific Instruments, 2019, 90(3): 035003.

25. Fajardo JM, Gomez O, Prieto F. EMG hand gesture classification using handcrafted and deep features[J]. Biomedical Signal Processing and Control, 2021, 63: 102210.

26. Hu Y, Wong Y, Wei W, et al. A novel attention-based hybrid CNN-RNN architecture for sEMG-based gesture recognition[J]. PloS one, 2018, 13(10): e0206049. https://doi.org/10.1371/journal.pone.0206049 PMID: 30375667

27. Atzori M, Gijsberts A, Castellini C, et al. Electromyography data for non-invasive naturally-controlled robotic hand prostheses[J]. Scientific data, 2014, 1(1): 1–13. https://doi.org/10.1038/sdata.2014.53 PMID: 25977804

28. Woo S, Park J, Lee J Y, et al. Cbam: Convolutional block attention module[C]//Proceedings of the European conference on computer vision (ECCV). 2018: 3–19.

29. Komodakis N, Zagoruyko S. Paying more attention to attention: improving the performance of convolutional neural networks via attention transfer[C]//ICLR. 2017.
30. Pizzolato S, Tagliapietra L, Cognolato M, et al. Comparison of six electromyography acquisition setups on hand movement classification tasks[J]. PloS one, 2017, 12(10): e0186132. https://doi.org/10.1371/journal.pone.0186132 PMID: 29023548

31. Atzori M, Gijsberts A, Heynen S, et al. Building the Ninapro database: A resource for the biorobotics community[C]//2012 4th IEEE RAS & EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob). IEEE, 2012: 1258–1265.

32. Um T T, Pfister F M J, Pichler D, et al. Data augmentation of wearable sensor data for parkinson’s disease monitoring using convolutional neural networks[C]//Proceedings of the 19th ACM International Conference on Multimodal Interaction. 2017: 216–220.

33. Gulati P, Hu Q, Atashzar SF. Toward Deep Generalization of Peripheral EMG-Based Human-Robot Interfacing: A Hybrid Explainable Solution for NeuroRobotic Systems[J]. IEEE Robotics and Automation Letters, 2021, 6(2): 2650–2657.

34. Wei W, Hong H, Wu X. A Hierarchical View Pooling Network for Multichannel Surface Electromyography-Based Gesture Recognition[J]. Computational Intelligence and Neuroscience, 2021.

35. Betthausler J L, Krall J T, Bannowsky S G, et al. Stable responsive EMG sequence prediction and adaptive reinforcement with temporal convolutional networks[J]. IEEE Transactions on Biomedical Engineering, 2019, 67(6): 1707–1717. https://doi.org/10.1109/TBME.2019.2943309 PMID: 31545709

36. Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]//International conference on machine learning. PMLR, 2015: 448–456.

37. Peng S, Jiang W, Pi H, et al. Deep snake for real-time instance segmentation[C]//Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020: 8533–8542.

38. Ding Z, Yang C, Tian Z, et al. sEMG-based gesture recognition with convolution neural networks[J]. Sustainability, 2018, 10(6): 1865.

39. Kim J S, Kim MG, Pan SB. Two-Step Biometrics Using Electromyogram Signal Based on Convolutional Neural Network-Long Short-Term Memory Networks[J]. Applied Sciences, 2021, 11(15): 6824.

40. Tosin M C, Cene V H, Balbinot A. Statistical feature and channel selection for upper limb classification using sEMG signal processing[J]. Research on Biomedical Engineering, 2020, 36(4): 411–427.