Research Article

Feature Fusion-Based Improved Capsule Network for sEMG Signal Recognition

Wanliang Wang,1 Wenbo You,1 Zheng Wang,2 Yanwei Zhao,1 and Sheng Wei1

1College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, Zhejiang, China
2College of Modern Information Technology, Zhejiang Institute of Mechanical Electrical Engineering, Hangzhou 310053, Zhejiang, China

Correspondence should be addressed to Wanliang Wang; zjutwwl@zjut.edu.cn

Received 3 September 2021; Revised 26 November 2021; Accepted 31 December 2021; Published 21 January 2022

Academic Editor: Simone Ranaldi

Copyright © 2022 Wanliang Wang et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

This paper proposes a feature fusion-based improved capsule network (FFiCAPS) to improve the performance of surface electromyogram (sEMG) signal recognition with the purpose of distinguishing hand gestures. Current deep learning models, especially convolution neural networks (CNNs), only take into account the existence of certain features and ignore the correlation among features. To overcome this problem, FFiCAPS adopts the capsule network with a feature fusion method. In order to provide rich information, sEMG signal information and feature data are incorporated together to form new features as input. Improvements made on capsule network are multilayer convolution layer and e-Squash function. The former aggregates feature maps learned by different layers and kernel sizes to extract information in a multiscale and multiangle manner, while the latter grows faster at later stages to strengthen the sensitivity of this model to capsule length changes. Finally, simulation experiments show that the proposed method exceeds other eight methods in overall accuracy under the condition of electrode displacement (86.58%) and among subjects (82.12%), with a notable improvement in recognizing hand open and radial flexion, respectively.

1. Introduction

Surface electromyogram (sEMG), an electrical signal generated on skin surface during muscle contraction, contains rich information about muscle activity and can be used to identify the subject’s action intention [1]. Thus, it is commonly used in rehabilitation, intelligent bionics, and human–computer interaction [2]. In practical applications, accurate and fast classification of signals is the basic prerequisite. Since sEMG signal is susceptible to factors such as physiological state of the subject or physical characteristics of the device, accuracy of classification algorithms will vary from person to person or time to time. As a consequence, algorithms with higher accuracy under different conditions are in urgent need.

In early stages of sEMG recognition, feature extraction is often applied to time windows, and feature data is sent to machine learning models, such as support vector machine (SVM), random forest (RF), and linear discriminant analysis (LDA) [3–5]. However, methods as mentioned are already clearly defined, which means there are only limited ways to improve classification performance. That is, adjust hyperparameters or change the combination or order of features. And the effects of those methods are not that satisfying.

With the development of deep learning, more problems in fields like image processing or speech recognition are solved in a better way. Recently, deep learning methods are adopted in sEMG recognition, as deep learning models have strong abilities of feature extraction and model fitting [6]. Geng et al. [7] put forward the concept of sEMG image, which means the conversion from multichannel signal into images and completed the gesture recognition experiments based on convolution neural network (CNN), achieving the accuracy of 65.1% of NinaPro DB1 dataset. Chen et al. [8] designed a deep CNN model and tested it on NinaPro DB5 dataset, where the signal was processed with continuous
wavelet transform, thus reducing the number of parameters and achieving the accuracy of 69.62%, 67.42, and 61.63% for three subsets. Barron et al. [9] proposed to classify sEMG signals of different subjects by using recurrent neural network (RNN), which is higher in classification accuracy of 79.7% in their own dataset.

Although the recognition performance is gradually improving with the help of deep learning, classification of sEMG signal still faces challenges [10]. On the one hand, deep learning methods are usually applied to images and their appliance in one-dimensional signal is limited. To adapt to various models, sEMG signal is often transformed into two-dimensional data, like sEMG images [7]. For sEMG images that vary linearly from sEMG signal or sequences cut from sEMG signal [11], they contain a lot of redundant information and cannot reflect the characteristics in frequency domain or time-frequency domain well. On the other hand, pooling operation, which is commonly applied to downsample features and decrease parameters, suffers from information loss. Meanwhile, it is impossible to discover the potential links between features using convolution only, the most commonly used method to extract features from input. Such lost information and undiscovered correlation among features may contain key information for certain gestures and help to solve the problem where current algorithms fail to achieve satisfying results. Those problems may occur when there are small differences in the muscles contracted for some gestures or too many gestures to be recognized.

To address the previously mentioned challenges, this paper proposes an improved capsule network (CapsNet) method based on feature fusion. Using fused features instead of sEMG images or sequences, the complexity of the model is reduced. Moreover, the structure of capsule network is improved to make the network more suitable for the characteristics of sEMG signals. The proposed method has achieved good results on the experiments of electrode displacement and different subjects respectively. The main contributions of this paper are summarized as follows.

First, this research proposes a new method to generate features by feature fusion. Fused features not only make up for information loss during feature extraction but also enable the model to explore the potential links between features.

Second, the original convolution layer is replaced with multilevel convolution layer, so that features of different layers and angles are concatenated to enrich the information for subsequent layers, and thus, the model is deepened the broadened.

Third, FiFiCaps modifies the squash function to suit the recognition of sEMG signal by adopting e-Squash. The effects of squash function will vary with datasets. The e-Squash enables higher growing rate as the capsule length grows to a certain extent to strengthen the model’s sensitivity.

The structure of this paper is as follows. The second section introduces the improved capsule network framework based on feature fusion, the third section carries out relevant experiments and verification on the proposed method, and the fourth section summarizes the full text.

2. The Proposed Method

In this section, we first introduce the method to combine sEMG signal and feature data. Then, we introduced the structure of CapsNet and proposed improvements are explained in detail.

2.1. Feature Fusion Method. This method mainly consists of two stages. The first one is dealing with feature data and sEMG signal information, while the other one is stacking them together to form new abstract feature maps. The detailed process is shown in Figure 1.

Suppose $m$ features are extracted from one channel of sEMG window of length $w$, and the feature vector $f$ can be expressed as

$$f = [1, x_1, x_2, \ldots, x_p, \ldots, x_m],$$

where $x_i$ denotes a single feature and 1 is added to maintain original features. Here, $m$ equals 14 and $w$ equals 300. The detailed description of features is shown in Table 1 in Section 3.2.

Feature vectors are combined according to the rules of matrix multiplication and transformed into feature maps. The calculation process is shown as follows:

$$F = G(\alpha \ast (f^T \times f) \beta),$$

where $\times$ denotes matrix multiplication, $\alpha$ and $\beta$ are conversion parameters, both equal to 0.5, $G$ is the sigmoid function shown in (3), and $F$ represents the feature map that contains fused abstract features.

$$G(\theta) = \text{sigmoid}(\theta) = \frac{1}{1 + e^{-\theta}}$$

where $\theta$ is the only input of the sigmoid function $G$ that is applied to restrict the output to range (0, 1).

As sEMG signal contains rich information, it is considered to be fused into feature maps to make up for information loss. Each sEMG window of length $w$ is cut into $n$ segments according to the length $l$. The parameters $n$ and $l$ meet the demand of

$$\text{dif} = \min(|n - l|) \quad \text{s.t} \quad n \ast l = w,$$

where dif means the minimum absolute value of difference between $n$ and $l$. Therefore, $n$ and $l$ are equal to 15 and 20, respectively. These segments were stacked to form a sEMG matrix of dimension $n \times l$.

Convolutions with different kernel sizes are applied to fuse the processed sEMG signal and feature matrix as their shapes are different. In this paper, 256 convolutions of kernel size $3 \times 8$ and of $3 \times 3$ are used to deal with sEMG matrix and feature matrix, respectively, using for both a stride equal to 2. Following the previously mentioned operations, feature maps generated from feature data and sEMG signal are concatenated on the channel, which is often the method to fuse feature maps. Moreover, $1 \times 1$ convolution with 256
channels is introduced to strengthen the expressiveness of network. Finally, as features of different sources vary from amplitudes, Min-Max normalization has to be done before training, as it is shown in the following equation:

\[
X_{\text{norm}} = \frac{X_i - X_{\text{min}}}{X_{\text{max}} - X_{\text{min}}}
\]

where \(X_i\) is a single data in a feature map, \(X_{\text{min}}\) means the minimum value in the feature map, \(X_{\text{max}}\) means the maximum value in the feature map, and \(X_{\text{norm}}\) means the normalized output. The normalization operation accelerates the process of gradient descent and enables features originated from sEMG signal and feature data to make the same contribution to results.

2.2. Improved CapsNet. Capsule network is a model with vectors as basic blocks [11], which not only reflects whether a certain feature is present or not but also has the ability to uncover the potential connections between features. The improved capsule network replaces the traditional single convolutional layer with a multilevel convolutional layer and squashed function with an e-Squash function on original capsule network. The specific structure is shown in Figure 2.

2.2.1. CapsNet. Capsule network is a deep learning model based on capsules. A capsule consists of multiple neurons, and each neuron represents a certain attribute of a specific entity in the input. The vector length of capsule ranges from \([0, 1]\), with larger lengths representing higher possibility of a certain entity to exist. And the vector direction suggests the instanitated parameter. Different from CNN, capsule network abandons pooling layer and adopts dynamic routing mechanism instead to connect capsules at different levels, thus achieving robust and reliable results [14]. The process of dynamic routing can not only obtain the spatial relationship between the whole and parts but also route the information between capsules by strengthening the connection between capsules, so that capsules at different levels can achieve high consistency [15]. The specific process is shown in Figure 3.

As shown in Figure 3, \(u_i\) is the output of primary capsule \(i\), and it is multiplied by transformation matrix \(W_{ij}\) to obtain prediction vector \(\hat{u}_{ji}\). All prediction vectors \(\hat{u}_{ji}\) are

---

**Table 1: Features selected in the experiment.**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Name</th>
<th>Description</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>(x_1)</td>
<td>RMS [12]</td>
<td>Relates to amplitude changes</td>
<td>(\sqrt{\frac{1}{w} \sum_{i=1}^{w} S_i^2} )</td>
</tr>
<tr>
<td>(x_2)</td>
<td>MAV [12]</td>
<td>Reflection on muscle contraction levels</td>
<td>(\frac{1}{w} \sum_{i=1}^{w}</td>
</tr>
<tr>
<td>(x_3)</td>
<td>WL [12]</td>
<td>Accumulated wavelength</td>
<td>(\sum_{i=1}^{w} S_i - S_{\text{max}} )</td>
</tr>
<tr>
<td>(x_4)</td>
<td>ZC [12]</td>
<td>Times that sEMG signal crosses zero in amplitude</td>
<td>(\sum_{i=1}^{w} \text{sgn}(S_i \cdot S_{\text{max}}) \cap</td>
</tr>
<tr>
<td>(x_5)</td>
<td>DASDV [12]</td>
<td>The standard deviation value of wavelength</td>
<td>(\frac{1}{w-1} \sum_{i=1}^{w-1} (S_{i+1} - S_i)^2 )</td>
</tr>
<tr>
<td>(x_6)</td>
<td>LOG [12]</td>
<td>Estimation of muscle contraction force</td>
<td>(\exp(\frac{1}{w} \sum_{i=1}^{w} S_i) )</td>
</tr>
<tr>
<td>(x_7)</td>
<td>SSI [12]</td>
<td>Energy of the signal</td>
<td>(\sum_{i=1}^{w}</td>
</tr>
<tr>
<td>(x_8)</td>
<td>TM3 [12]</td>
<td>The third moment of sEMG signal</td>
<td>(\frac{1}{w} \sum_{i=1}^{w}</td>
</tr>
<tr>
<td>(x_9)</td>
<td>TM4 [12]</td>
<td>The fourth moment of sEMG signal</td>
<td>(\frac{1}{w} \sum_{i=1}^{w}</td>
</tr>
<tr>
<td>(x_{10})</td>
<td>TM5 [12]</td>
<td>The fifth moment of sEMG signal</td>
<td>(\frac{1}{w} \sum_{i=1}^{w}</td>
</tr>
<tr>
<td>(x_{11})</td>
<td>FR [12]</td>
<td>Degree of muscle contraction and relaxation</td>
<td>(\int_{\text{UHC}}^{\text{UHC}} P(S') dS' )</td>
</tr>
<tr>
<td>(x_{12})</td>
<td>IEMG [12]</td>
<td>Estimation of signal power</td>
<td>(\int_{0}^{\infty} S' dS' )</td>
</tr>
<tr>
<td>(x_{13})</td>
<td>MFMN [13]</td>
<td>Average frequency of sEMG</td>
<td>(\frac{1}{2} \int_{0}^{\infty} P(S') dS' )</td>
</tr>
<tr>
<td>(x_{14})</td>
<td>MFMD [13]</td>
<td>Median frequency of sEMG</td>
<td>(\frac{1}{2} \int_{0}^{\infty} P(S') dS' )</td>
</tr>
</tbody>
</table>
multiplied and accumulated by their coupling coefficients $c_{ij}$ and then, the output is mapped to a restricted range by squash function to obtain the output $v_j$ of action capsule $j$. The formula of calculation process is as follows:

$$\hat{v}_{ji} = W_{ij}u_i,$$

where $W_{ij}$ is the transformation matrix that connects primary capsule $i$ and action capsule $j$, and $\hat{v}_{ji}$ is the prediction vector of the $i$-th primary capsule for the $j$-th action capsule. The coupling coefficients $c_{ij}$ is calculated as

$$c_{ij} = \frac{\exp(h_{ij})}{\sum_k \exp(h_{jk})}$$

where $c_{ij}$ is the coupling coefficient between primary capsule $i$ and action capsule $j$. The sum of all coupling coefficients between primary capsule $i$ and all action capsules is 1. For the value of the coupling coefficient $c_{ij}$, it is determined by the prior probability $b_{ij}$ that the primary capsule $i$ is coupled to action capsule $j$ and the initial value of $b_{ij}$ is set to 0. Prior probability $b_{ij}$ will be updated in the subsequent iterations to update the coupling coefficient $c_{ij}$.

$$s_j = \sum_i c_{ij} \hat{v}_{ji},$$

where $s_j$ is the weighted sum of all prediction vectors $\hat{v}_{ji}$ and is fed into the squash function.

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} s_j,$$

where $v_j$ is the output of action capsule $j$. In order to restrict the length of the vector to $[0,1)$ to represent the probability of entity existence, a nonlinear squash function is used to reduce the vector size to 0 for shorter vector lengths and to slightly less than 1 for longer vector lengths.

$$b_{ij} = b_{ij} + \hat{v}_{ji} \cdot v_j.$$

Before calculating new coupling coefficient $c_{ij}$, value of the prior probability $b_{ij}$ is updated and the product of predicted vector $\hat{v}_{ji}$ and output vector $v_j$ is added to the original base. A larger value of the product represents a closer orientation of a lower level capsule to a higher level capsule and a larger coupling coefficient between the capsules. In this way, the coupling between two similar capsules will be tighter during iterations.

For each action capsule, a separate margin loss function is used to calculate the loss. The total loss of the network is equal to the sum of the losses of all action capsules.

$$L_k = \sum_{k=1}^N \left( T_k \max(0, m^+ - \|v_k\|) \right)^2 + \lambda (1 - T_k) \max(0, \|v_k\| - m^-)^2,$$

(11)

where value of $T_k$ is either 0 or 1. The value of 1 for $T_k$ means that the entity represented by this action capsule exists and vice versa. For hyperparameters $m^+$ and $m^-$, they are set to 0.9 and 0.1, which mean loss is 1 if the possibility is less than 0.1 and 0 if the possibility is greater than 0.9, respectively. Weight parameter $\lambda$ serves to determine the influence of predicting an incorrect label and it is set to be 0.5. The number of iterations for dynamic routing is 3.

### 2.2.2. Multilevel Convolution

Original capsule network extracts features by using a convolution layer and then feeds the output to subsequent layers.

Instead of a single convolution layer, a multilevel convolutional layer is proposed, where convolutional kernels of different scales are adopted to extract information in a multiscale and multiangle manner. To be specific, we adopt 256 convolutions of kernel size $3 \times 3$ and 256 convolutions of kernel size $5 \times 5$ and stack the feature maps gained by these operations on the channel. So as not to miss the information of the previous layer, the output of previous layer is stacked directly with obtained feature maps.

As stacking on the channels brings an increase in the number of channels and parameters, a bottleneck layer, 256 convolutions with a kernel size of $1 \times 1$, is introduced to reduce the number of channels and parameters and shorten the calculation time.

### 2.2.3. $e$-Squash

The squash function of capsule network can be divided into two parts according to their functions: one is to find the unit vector of the input vector, and the other is to compress the length of this unit vector to $[0,1)$ by a nonlinear function, thus realizing the activation function of the capsule. Improvements are made for the latter part.

The original squash function grows rapidly at the starting stage, and even if the length of a particular capsule is small, it is still able to obtain a relatively large activation value. As the length of the capsule grows, the function value slowly approaches to 1. In order to suit the characteristics of sEMG signal, the growth rate of the function needs to be appropriately changed. However, too large growth rate in the initial stage is likely to lead to too large activation values. The probability of the existence of the entity represented by the capsule will be amplified due to the sensitivity of dynamic routing, leading to a decrease in classification accuracy. Therefore, the squash function needs to be improved to
maintain its growth rate at the initial stage and to enhance
the function growth rate in the subsequent stages to find out
valid information.

Based on the previously mentioned discussion, an im-
provement for the nonlinear function part of the Squash
function was proposed in this paper and the new function is
called e-Squash as follows:

\[
v_j = \left(1 - e^{-\|y_j\|_1}\right) \frac{y_j}{\|f_j\|_1}
\] (12)

The function curve is almost identical to that of the
original Squash function when the vector length is small and
rises faster than the original Squash function after the vector
length reaches a certain value, which improves the sensitivity
of the response and is eventually stable.

3. Experiments and Analysis

3.1. The sEMG Dataset. The sEMG dataset was acquired
using ELONXI device developed by the team at University of
Portsmouth, UK [16]. This device supports a maximum of 16
channels with a sampling resolution of 24 bits and a sam-
pling frequency between 1000 Hz and 2000 Hz. In this
dataset, 16 channels with a sampling frequency of 1000 Hz
mode were selected, and the filtered signal was obtained
using the filter that was built within this system. In specific,
the signal passed through a band-pass filter (20–500 Hz), and
then, a band-stop filter at 50 Hz is used to remove the power
line interference.

Eight subjects’ sEMG signals were collected for six
different time periods, and the same five gestures were
collected for each time period. The movements are hand
closed (HC), hand open (HO), radial flexion (RF), wrist
flexion (WF), and wrist extension (WE). To simulate dif-
dent situations, three time periods are executed in the
morning and the rest three time periods are in the afternoon,
with the equipment worn once in the morning and once in
the afternoon, respectively. In order to reduce the influence
of muscle fatigue, all subjects rested for 10 seconds between
every two movements and 30 minutes between every two

3.2. Preprocessing and Feature Extraction. Considering that
sEMG is a period of time series, window analysis is ap-
plied to preprocess the signal. This method mainly in-
volves two parameters, namely, the length of the window
and the increment interval. The length of window rep-
resents the unit length of signal processing, which di-
rectly affects the recognition accuracy. The incremental
interval \( \tau \) affects response time of the system, which is a
key factor for the application of sEMG signals. The
window cutting method is shown in Figure 4, where \( \omega \)
is the window length and \( \tau \) is the increment interval. Here,
\( \omega \) equals 300 and \( \tau \) equals 50.

As the purpose of this paper is to explore a new method
for sEMG recognition instead of feature selection, feature
selection is not discussed in detail, and 14 commonly used
features are selected, provided by time domain and fre-
quency domain.

In Table 1, \( S_{ij} \) represents the signal of a window. \( S_{fj} \)
represents the spectrum obtained by Fast Fourier transform
of the window, and \( P(S_j) \) represents the power spectrum
intensity obtained by calculating the square of the norm of
spectrum \( S_j \). The features used in the experiment are shown in Table 1.

3.3. Recognizing sEMG Signal under Different Conditions.
In current applications of sEMG signal, there are two main
difficulties getting in the way. First, there are differences in
the position of the sEMG acquisition device each time it is
worn, resulting in different positions of the electrodes
respecting muscles. Besides, it is also common for
sEMG acquisition device to shift the position due to external
forces or other factors in actual use. Second, sEMG signals
have large differences in values between different people, so
it is necessary to dig deeply into the universal features of
sEMG signals. To address the previously mentioned issues,
two experiments are designed in this paper to test the
performance of the proposed method in case of electrode
displacement and different people.

The experimental environment is Windows 10, CPU i7-
9750H, GPU 1660Ti, and tensor flow. Structural parameters
of network are shown in Table 2.

3.3.1. Recognizing sEMG under Electrode Displacement.
The specific situation of electrode displacement is as follows:
sEMG from one subject was repeatedly collected three times
in the morning as the training set, and the data collected in
the afternoon was used as the testing set. Each subject wore
the device once in the morning and once in the afternoon,
thus simulating the case of electrode displacement. The
experiment was set with a batch-size of 32, and Adam al-
gorithm was used to optimize the loss. Different methods are
tested to see whether the proposed method works or not. The
results are displayed in Table 3.

FE means feature extraction, and FF means feature fu-
sion. Bolded fonts stand for best performance in each
column.

For traditional machine learning methods SVM and RF,
which used extracted features as the input, they have ex-
cellent results in distinguishing certain actions, reaching
100% accuracy in action WE and WF, but fail to reach high
accuracy in action HO and HC. Thus, the overall perfor-
mance is not that satisfying. It shows traditional machine
learning methods have some defects to some extent.

CNN is a commonly used model for deep learning. In
this experiment, sEMG signal, extracted features, and fused
features are used as inputs for comparison.

Using sEMG signal as input of CNN, the accuracy is the
lowest among all three inputs because sEMG signal contains
a lot of redundant information and can hardly reflect the
features of other domains. In particular, the accuracy of
action HO and RF are less than 15%. The accuracy of CNN
with features as input can hardly recognize action HO,
indicating that it is difficult to obtain the information
characterizing action HO. It is worth noting that overall accuracy of using the fused features is the highest among three inputs, and it performs relatively better on action HO than discrete features or original signal, though it is still at a low level, indicating that this feature fusion method can characterize some of the properties of action HO. For DNN which uses extracted features as input, it has a low overall accuracy due to poor performance on action HO.

TDACAPS, CAPS, and FFiCAPS are models based on capsule networks. The difference is that TDACAPS adds attention mechanism to capsule network, and FFiCAPS makes improvements to the structure of capsule network. Besides, TDACAPS takes extracted features as input, while CAPS and FFiCAPS takes fused features as input. FFiCAPS improves the convolution layer and squash function of the capsule network compared with CAPS, thus making the capsule network more suitable for the recognition of sEMG signals. The overall accuracy of FFiCAPS is higher than that of CAPS and TDACAPS and is the highest among all methods. More importantly, FFiCAPS achieves the best

Table 2: Structural parameters of improved CapsNet.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Operation</th>
<th>Kernel/stride</th>
<th>Output/channel</th>
<th>Padding</th>
</tr>
</thead>
<tbody>
<tr>
<td>Input</td>
<td>-</td>
<td>-/-</td>
<td>7 × 7/256</td>
<td>-</td>
</tr>
<tr>
<td>Conv3×3</td>
<td>-</td>
<td>3 × 3/1 × 1</td>
<td>7 × 7/256</td>
<td>Y</td>
</tr>
<tr>
<td>Conv5×5</td>
<td>-</td>
<td>5 × 5/1 × 1</td>
<td>7 × 7/256</td>
<td>Y</td>
</tr>
<tr>
<td>ML-Conv</td>
<td>Concat</td>
<td>-/-</td>
<td>7 × 7/768</td>
<td>-</td>
</tr>
<tr>
<td>Conv1×1</td>
<td>1 × 1/1 × 1</td>
<td>7 × 7/256</td>
<td>N</td>
<td></td>
</tr>
<tr>
<td>PrimaryCaps</td>
<td>Reshape</td>
<td>-/-</td>
<td>7 × 7/8/32</td>
<td>-</td>
</tr>
<tr>
<td>ActionCaps</td>
<td>-</td>
<td>-/-</td>
<td>5/16</td>
<td>-</td>
</tr>
<tr>
<td>Class</td>
<td>-</td>
<td>-/-</td>
<td>5/-</td>
<td>-</td>
</tr>
</tbody>
</table>

Besides, detailed process and parameters of feature fusion method are shown in Figure 5.

![Figure 4: Window analysis method. sEMG signal is cut into time windows of the same length (w) with the same time interval r.](image1)

![Figure 5: The process of feature fusion method. Detailed parameters and operations are shown.](image2)
accuracy of 80.61% for action HO, which indicates that the proposed method can indeed mine the information characterizing action HO.

It is worth noting that no method has accuracy higher than 85% for HO. Some gestures trigger similar changes in sEMG signal to others and are therefore difficult to classify accurately. This may account for poor classification performance for HO. Comparing three types of inputs, sEMG signal, extracted features, and fused features, the experimental results illustrate the validity of proposed feature fusion method. Meanwhile, it can also be concluded that improvements on CapsNet are suitable for sEMG signal recognition, considering FFiCAPS achieves the highest accuracy of 80.61%.

In addition, the accuracy and loss performance of each method is compared in this section. Figure 6(a) shows the curve of testing accuracy of each method, and Figure 6(b) shows the loss curve of each method. From Figure 6(b), it can be seen that the loss of each method decreases and tends to be smooth with the increase of iterations. And the test accuracy in Figure 6(a) increases gradually with the increase of iterations. Compared with CNN and DNN, the accuracy of methods based on capsule network lags behind at first, but it gets higher after a certain number of iterations. Among TDACAPS, CAPS + FF, and FFiCAPS, FFiCAPS reaches the highest accuracy in the end.

### 3.3.2. Recognizing sEMG from Different People

To recognize sEMG signals from different people, it is required that the classifier should be able to mine general features from sEMG and find out the link between these features and gestures from subjects. The experiment was set with a batch size of 16, and Adam algorithm was used to optimize the loss. Different methods are tested to see whether the proposed method works. The results are displayed in Table 4.

It is demonstrated that traditional machine learning methods perform well in this experiment with overall accuracy higher than 70%. RF even achieves the highest accuracy in distinguishing action WE, though its accuracy for action WF is relatively low. As for SVM, it performs well in action WF but does not get high accuracy in action HO. Therefore, the machine learning approach has some advantages when it comes to robustness of the algorithm or mining general characteristics of sEMG signal. However, the recognition accuracy still needs to be improved, which does not reach 80%.

The accuracy of CNN with sEMG signal as input for different subjects is only 69.27%, the lowest of all methods,

<table>
<thead>
<tr>
<th>Method</th>
<th>HC</th>
<th>HO</th>
<th>RF</th>
<th>WE</th>
<th>WF</th>
<th>Overall accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM + FE [17]</td>
<td>36.41 ± 0.00</td>
<td>3.08 ± 0.00</td>
<td>86.15 ± 0.00</td>
<td><strong>100 ± 0.00</strong></td>
<td><strong>100 ± 0.00</strong></td>
<td>65.13 ± 0.00</td>
</tr>
<tr>
<td>RF + FE [18]</td>
<td>65.64 ± 0.00</td>
<td>14.36 ± 0.00</td>
<td>84.10 ± 0.00</td>
<td><strong>100 ± 0.00</strong></td>
<td><strong>100 ± 0.00</strong></td>
<td>78.36 ± 0.00</td>
</tr>
<tr>
<td>CNN [19]</td>
<td><strong>98.72 ± 1.67</strong></td>
<td>11.42 ± 8.18</td>
<td>14 ± 9.55</td>
<td>97.79 ± 1.38</td>
<td>90 ± 3.49</td>
<td>63.36 ± 2.31</td>
</tr>
<tr>
<td>CNN + FE</td>
<td>81.38 ± 4.24</td>
<td>1.64 ± 1.16</td>
<td>89.64 ± 3.60</td>
<td>99.90 ± 0.31</td>
<td>91.90 ± 0.45</td>
<td>74.03 ± 0.76</td>
</tr>
<tr>
<td>CNN + FF</td>
<td>94.56 ± 4.44</td>
<td>35.33 ± 15.32</td>
<td>74.87 ± 10.10</td>
<td>97.85 ± 1.39</td>
<td>85.18 ± 4.42</td>
<td>78.77 ± 1.26</td>
</tr>
<tr>
<td>DNN + FE [20]</td>
<td>84.67 ± 3.89</td>
<td>0.50 ± 0.63</td>
<td>92.92 ± 4.15</td>
<td>99.48 ± 0.92</td>
<td>92.31 ± 0.00</td>
<td>75.18 ± 1.25</td>
</tr>
<tr>
<td>TDACAPS [21]</td>
<td>97.59 ± 2.62</td>
<td>74.87 ± 15.71</td>
<td><strong>95.18 ± 4.42</strong></td>
<td>98.56 ± 1.82</td>
<td>55.39 ± 16.87</td>
<td>84.77 ± 0.82</td>
</tr>
<tr>
<td>CAPS + FF</td>
<td>98.41 ± 4.27</td>
<td>44.15 ± 9.82</td>
<td>92.51 ± 7.58</td>
<td>99.74 ± 0.77</td>
<td>78.92 ± 6.71</td>
<td>84.04 ± 2.54</td>
</tr>
<tr>
<td>*FFiCAPS</td>
<td>97.74 ± 4.26</td>
<td><strong>80.61 ± 8.53</strong></td>
<td>93.02 ± 5.78</td>
<td>99.74 ± 0.77</td>
<td>55.26 ± 16.92</td>
<td><strong>86.58 ± 2.70</strong></td>
</tr>
</tbody>
</table>
while the accuracy with extracted features or fused features as input is above 75%. Moreover, the accuracy of the fused features is 0.27% higher than that of extracted features, which illustrates the effectiveness of the feature fusion method proposed in this paper. Although the overall improvement seems small, it does help to discover the valid information for action WF compared with CNN with extracted features. It can be seen that CNN with sEMG signal as input reaches the highest accuracy in recognizing action WF and CNN with extracted features only get 19.91%. For CNN with fused features as input, it is 33.66% higher than that of CNN with extracted features as input, which proves that the proposed method enriches the input information. DNN achieves the best accuracy on action HC and WE but does not perform well on action HO and RF, resulting in its total accuracy lower than 80%.

The accuracy of capsule networks was generally higher than that of CNNs, indicating that capsule networks can reduce the loss of information and facilitate the mining of general features of sEMG signal. Among the three methods based on capsule network, TDACAPS achieves the highest accuracy on action HO. FFiCAPS achieves the highest overall accuracy, which is 6.2% higher than CAPS and a bit higher than TDACAPS. Moreover, the performance of FFiCAPS is the best on gesture HC, RF, WE, and WF among three methods based on capsule network, and it reaches the top accuracy on gesture RF among all methods.

Besides, the performance of all methods in accuracy and loss is compared in this section. In Figure 7(a), it can be seen that accuracy of all methods gradually increases in spite of some ups and downs. Compared with CNN and DNN, methods based on capsule network are left behind when iteration is less than 10 but catch up with other methods later. As for loss in Figure 7(b), all methods decrease gradually, and there is no obvious lag in methods based on capsule network.

### 3.3.3. Testing for Different Squash Functions

Squash function is a crucial component of capsule network. So as to test e-Squash, various squash functions are compared in this section. The squash function equations and curves are shown in Table 5 and Figure 8, respectively. The effectiveness of the proposed e-Squash function is verified by experiments and the results are shown in Table 6.

In the case of electrode displacement, all methods using e-Squash obtain a higher accuracy than those using the original Squash function. As for FFiCAPS, four different

### Table 4: Recognition accuracy under different people.

<table>
<thead>
<tr>
<th>Method</th>
<th>Gesture recognition accuracy (%)</th>
<th>Overall accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HC</td>
<td>HO</td>
</tr>
<tr>
<td>SVM + FE [17]</td>
<td>93.85 ± 0.00</td>
<td>54.19 ± 0.00</td>
</tr>
<tr>
<td>RF + FE [18]</td>
<td>93.76 ± 0.00</td>
<td>81.45 ± 0.00</td>
</tr>
<tr>
<td>CNN [19]</td>
<td>68.42 ± 2.55</td>
<td>50.78 ± 0.56</td>
</tr>
<tr>
<td>CNN + FE</td>
<td>99.94 ± 0.09</td>
<td>75.38 ± 14.97</td>
</tr>
<tr>
<td>CNN + FF</td>
<td>88.63 ± 0.17</td>
<td>76.83 ± 3.88</td>
</tr>
<tr>
<td>DNN + FE [20]</td>
<td>99.98 ± 0.05</td>
<td>75.89 ± 0.68</td>
</tr>
<tr>
<td>TDACAPS [21]</td>
<td>98.25 ± 3.89</td>
<td>84.98 ± 7.12</td>
</tr>
<tr>
<td>CAPS + FF</td>
<td>98.68 ± 2.29</td>
<td>69.86 ± 4.78</td>
</tr>
<tr>
<td>* FFiCAPS</td>
<td>98.78 ± 0.56</td>
<td>78.70 ± 5.40</td>
</tr>
</tbody>
</table>

FE means feature extraction, and FF means feature fusion. Bolded fonts stand for best performance in each column.
Squash functions show an improvement in overall accuracy compared with the original Squash function. e-Squash proposed in this paper shows the largest improvement of 4.84% followed by strict-Squash, indicating that e-Squash function is beneficial for improving accuracy.

In the case of different subjects, CAPS with e-Squash reaches a higher accuracy, while TDACAPS does not fit well with e-Squash and perform worse than original model. Perhaps, the combination of spatial attention and e-Squash fails to focus on valid information of abstract features and tend to cause overfitting. For FFiCAPS, the differences in overall accuracy are not significant. HSquash is slightly better than the original Squash function. Squash-4 and strict-Squash perform worse than the original Squash function. And e-Squash has the best result, with a 4.05% improvement compared with the original Squash.

3.3.4. Computational Resources Consumed. In this section, training time and recognition time for all methods under the case of electrode displacement and different subjects are compared in Table 7.

In terms of computing resources, SVM and RF only consume little time to train and recognize. For models based on CNN or DNN, their consumption of time is more than machine learning methods, but their accuracy is higher than that of SVM and RF. The models that have the largest time consumption are models based on capsule network, as they

### Table 5: The equations for different squash functions.

<table>
<thead>
<tr>
<th>Squash function</th>
<th>Equation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Squash [14]</td>
<td>( v_j = \frac{(|s_j|/4)^2/(1 + |s_j|^2)}{(|s_j|^2)} )</td>
</tr>
<tr>
<td>HSquash [22]</td>
<td>( v_j = \frac{(|s_j|^2 + |s_j|^2)}{|s_j|^2} )</td>
</tr>
<tr>
<td>Strict-squash [23]</td>
<td>( v_j = 0.69|s_j|^2 + 2^{-0.6}|s_j|^{-1.115} )</td>
</tr>
<tr>
<td>Squash-4 [24]</td>
<td>( v_j = |s_j|^2/0.5 + |s_j|^2|s_j|^2/|s_j|^2 )</td>
</tr>
<tr>
<td>e-Squash</td>
<td>( v_j = (1 - e^{-|s_j|^2})|s_j|^2/|s_j|^2 )</td>
</tr>
</tbody>
</table>

### Table 6: Recognition accuracy of different squash functions.

<table>
<thead>
<tr>
<th>Methods</th>
<th>Squash function</th>
<th>Under electrode displacement</th>
<th>Under different people</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAPS + FF</td>
<td>Squash [14]</td>
<td>84.04 ± 2.54</td>
<td>75.92 ± 1.38</td>
</tr>
<tr>
<td>CAPS + FF</td>
<td>e-Squash</td>
<td>84.68 ± 1.41</td>
<td>76.06 ± 0.61</td>
</tr>
<tr>
<td>TDACAPS [25] + FE</td>
<td>Squash [14]</td>
<td>84.77 ± 0.82</td>
<td>81.90 ± 1.09</td>
</tr>
<tr>
<td>TDACAPS [25] + FE</td>
<td>e-Squash</td>
<td>85.46 ± 0.92</td>
<td>77.23 ± 0.41</td>
</tr>
<tr>
<td>FFiCAPS + FF</td>
<td>Squash [14]</td>
<td>81.74 ± 1.53</td>
<td>78.07 ± 0.92</td>
</tr>
<tr>
<td>FFiCAPS + FF</td>
<td>HSquash [22]</td>
<td>82.06 ± 3.51</td>
<td>78.15 ± 1.17</td>
</tr>
<tr>
<td>FFiCAPS + FF</td>
<td>Strict-squash [23]</td>
<td>82.71 ± 1.66</td>
<td>75.95 ± 0.87</td>
</tr>
<tr>
<td>FFiCAPS + FF</td>
<td>Squash-4 [24]</td>
<td>82.39 ± 1.29</td>
<td>77.52 ± 0.86</td>
</tr>
<tr>
<td>FFiCAPS + FF</td>
<td>e-Squash</td>
<td><strong>86.58 ± 2.70</strong></td>
<td><strong>82.12 ± 0.90</strong></td>
</tr>
</tbody>
</table>

Bolded fonts stand for best performance in each column.

---

**Figure 8:** The squash function curves of five different squash functions.
do not apply pooling operations and dynamic routing requires a large number of training parameters. But they make improvement on overall performance and the recognition time is still acceptable, which meets the requirement for time delay.

4. Conclusions

In this paper, we propose a framework named FFiCAPS, which consists of a new method of generating fused features and capsule network with modifications. Our method is able to capture the correlation among extracted features and decrease the information loss.

The effectiveness of the proposed method is verified by two experiments that are under the case of electrode displacement and under different subjects. In electrode displacement, the accuracy for hand open is particularly poor. Among all competing methods, our method had the highest accuracy of 80.61% for hand open, which is much higher than that of models based on convolution neural network. Besides, FFiCAPS also achieved the best overall accuracy in electrode displacement, which illustrates the effectiveness of proposed method. When it comes to different subjects, the proposed method achieved the highest accuracy, though overall improvement compared with the second best accuracy is small. However, it is true that our method achieved the highest accuracy on four gestures out of five among three models based on capsule network and performed the best on radial flexion. This proves that in robustness our method also has some advantages.

The method proposed in this paper can meet the realistic requirements of both time delay and accuracy, but it still needs to be improved in terms of computational efficiency. Dynamic routing consumes a large amount of computational resources, so the next goal is to optimize the dynamic routing mechanism, improve the computational efficiency, and further enhance the performance of gesture recognition.

Data Availability

The data used to support the findings of this article are available from the corresponding author upon reasonable request.

Conflicts of Interest

The authors declare no conflicts of interest.

Acknowledgments

This work was supported by the Natural Science Foundation of China (nos. 51875524 and 61873240).

References


