



ARTICLE

A Novel SE-CNN Attention Architecture for sEMG-Based Hand Gesture Recognition

Zhengyuan Xu^{1,2,#}, Junxiao Yu^{1,#}, Wentao Xiang¹, Songsheng Zhu¹, Mubashir Hussain³, Bin Liu^{1,*} and Jianqing Li^{1,*}

¹Key Laboratory of Clinical and Medical Engineering, School of Biomedical Engineering and Informatics, Nanjing Medical University, Nanjing, 211166, China

²Department of Medical Engineering, Wannan Medical College, Wuhu, 241002, China

³Postdoctoral Innovation Practice, Shenzhen Polytechnic, Shenzhen, 518055, China

*Corresponding Authors: Jianqing Li. Email: jqli@njmu.edu.cn; Bin Liu. Email: liubin@njmu.edu.cn

#Zhengyuan Xu and Junxiao Yu contributed equally to this work

Received: 30 October 2021 Accepted: 03 March 2022

ABSTRACT

In this article, to reduce the complexity and improve the generalization ability of current gesture recognition systems, we propose a novel SE-CNN attention architecture for sEMG-based hand gesture recognition. The proposed algorithm introduces a temporal squeeze-and-excite block into a simple CNN architecture and then utilizes it to recalibrate the weights of the feature outputs from the convolutional layer. By enhancing important features while suppressing useless ones, the model realizes gesture recognition efficiently. The last procedure of the proposed algorithm is utilizing a simple attention mechanism to enhance the learned representations of sEMG signals to perform multi-channel sEMG-based gesture recognition tasks. To evaluate the effectiveness and accuracy of the proposed algorithm, we conduct experiments involving multi-gesture datasets Ninapro DB4 and Ninapro DB5 for both inter-session validation and subject-wise cross-validation. After a series of comparisons with the previous models, the proposed algorithm effectively increases the robustness with improved gesture recognition performance and generalization ability.

KEYWORDS

Hand gesture recognition; sEMG; CNN; temporal squeeze-and-excite; attention

1 Introduction

Hand gesture recognition is one of the most important perceptual channels in human-computer interaction. It is widely used in many fields, such as virtual reality, intelligent sign language translation for deaf-mute, rehabilitation therapy and assessment, bionic prosthesis, etc., and shows a broad potential in different applications [1]. However, hands are the most flexible parts of humans, so it is extremely challenging to examine, track, classify, or recognize various hand gestures.



Various research [2–5] has been done so far while hand gesture recognition based on vision is the first to bring up [2,6]. For the vision-based gesture recognition approach, usually, one or more cameras capture hand gesture images and videos for recognition, which is largely affected by the surrounding elements such as obstacles, sunlight, and even complex background [7]. To solve this problem, scholars have come up with hand gesture recognition based on electromyography (EMG). EMG is a physiological signal produced by skeletal muscles and provides sufficient information in muscle activities and function in neuroelectrophysiology. EMG is used for gesture recognition in human-computer interaction without potential interferences (obstacles, sunlight, and complex background, etc.) [8]. There are two main types of EMG recording electrodes [9]: surface electrodes and inserted electrodes. Inserted electrodes collect intramuscular EMG signals, providing more stable data but requiring invasive insertion that is not suitable for human-computer interaction. The non-invasive surface electrodes mainly collect surface EMG (sEMG) signals without harm. Therefore, it quickly catches the attention of scholars and is widely used in many aspects with high demands in gesture recognition. With high precision, easy wearing, and non-invasive properties, sEMG-based gesture recognition has become a hot spot in the research area [10–12].

Early studies of sEMG gesture recognition focus on combining artificially designed EMG signal features and machine learning classification methods. After many years of study, features that properly represent sEMG signals covering time, frequency, and time-frequency domains have been well designed and examined [12–14]. In terms of machine learning classification algorithms for sEMG, models including k-nearest neighbors (KNN), support vector machine (SVM), linear discriminant analysis (LDA), random forest (RF), and artificial neural network (ANN) have been well studied and achieved relatively high classification accuracy [15–18]. Despite the advantages of traditional methods, some useful information is still missing in the process of feature extraction and combination when a large amount of computation is required, which makes it difficult to improve the functionality and capability of gesture recognition based on sEMG [19].

With the development of science and technology, deep learning is well established and equipped with the ability of modeling and feature extraction [20]. It has been applied to and achieved breakthrough outcomes in many fields such as image processing [21] and speech recognition [22]. Recently, sEMG-based gesture recognition by deep learning starts to raise attention. Park et al. [23] firstly come up with an end-to-end CNN model capable of classifying six gesture categories in the Ninapro dataset. Compared to traditional SVM models, the end-to-end convolutional neural network (CNN) model has a significantly higher recognition rate, showing that deep learning has played an efficient role in sEMG-based gesture recognition. Atzori et al. [24] proposed a modified version of LeNet for the classification of multiple Ninapro datasets, each containing 50 hand gestures on average. After comparing with early classification models, this new simple network shows better results than KNN, SVM, and LDA yet has a lower accuracy than that of RF. Based on a simple network model, Wei et al. [25] come up with a new strategy by dividing raw EMG signals into small sections with the same duration (e.g., 200 ms per section) in a multi-stream stage. Then each section is trained with a CNN model and convoluted at the fusion stage to get the final classification results. This method achieves a nearly 85% recognition rate on the NinaPro DB1 dataset. Meanwhile, inspired by Atzori et al. [24], Tsinganios et al. [26] added dropout layers to the simple network model and replace the average pooling with max pooling. The changes finally help them achieve a 3% higher recognition rate than the old one. Ding et al. [19] proposed a parallel multi-scale convolution architecture to extract EMG signal features by different blocks and discriminate the gestures by grouping these features. In this case, they achieve the highest recognition rate of 78.86% on the NinaPro DB2 dataset for all gestures.

From the previous work, gesture recognition based on the integration of deep learning and sEMG signal features including time, frequency, and time-frequency domains is widely proposed by scholars. Hu et al. [27] introduced a CNN-recurrent neural network (RNN) model based on an attention mechanism to extract features from each sEMG channel with the Phinyomark feature. They report that the model obtains 87% and 82.2% of recognition rates on NinaPro DB1 and NinaPro DB2 datasets, respectively. Côté-Allard et al. [28] extended the model of Atzori et al. [24] with a new architecture composed of three combinations: 1) ConvNet with Spectrograms, 2) ConvNet with Continuous Wavelet Transforms, and 3) ConvNet with raw EMG signals. The model is jointly trained with transfer learning to improve the recognition rate. They finally achieve a 68.98% recognition rate in the wrist gestures recognition task using NinaPro DB5. Wei's group proposes a multi-view CNN model by combining 11 classical sEMG feature sets [29]. The multi-CNN model finally achieves a 90% of recognition rate by simply using sEMG signals. Chen et al. [30] came up with a compact Convolution Neural Network (EMGNet) jointly with Continuous Wavelet Transforms (CWT) for hand gesture recognition. They finally obtain 69.62%, 67.42%, and 61.63% of recognition rates for finger gestures, wrist gestures, and functional gestures, respectively.

Despite all the advantages from the previous work, several problems need to be taken into consideration: 1) simple architecture always results in a low recognition rate; 2) algorithms with high recognition rates have relatively poor real-time inference ability and usually require extremely complex architecture; 3) few groups do the “subject-wise [31]” study while those who do only have unsatisfied results. To solve these concerns, Josephs et al. [32] proposed a simple model based on an attention mechanism with a good real-time data analysis. This model uses the 1st, 4th, and 6th samples in every repeated move as the training set, the 3rd sample as the validation set, and the 5th as the testing set. This approach achieves a recognition rate of 87.09% on NinaPro DB5, 74.88% on NinaPro DB4, and 62.96% on NinaPro DB5 after the “subject-wise” cross-validation. Even if Josephs et al. [32] achieved pretty good results, we find that the model shows weak generalization ability.

To solve the mentioned problems as well as increase the recognition rate and generalization ability, we propose a novel squeeze-and-excite CNN (SE-CNN) attention architecture for sEMG-based hand gesture recognition. The main contributions to this paper are as follows:

- The introduction of the temporal squeeze-and-excite block into the CNN frame is to establish the possible relations between channels and enhance important features while suppressing useless ones. Therefore, the robustness and recognition rate of the model can be sufficiently improved. A simple attention mechanism is added to the end of the model to capture space relations between features and enhance the learned representations of multi-channel signals.
- Batch normalization (BN) and parametric rectified linear unit (PReLU) are added to the CNN model to approximate any arbitrary function. This will also accelerate the convergence, prevent gradient explosion and vanishing, avoid overfitting, and enhance further expression of CNN, and therefore, improve the model's generalization ability.
- To the best of our knowledge, the proposed hybrid strategy in combination with SE, CNN, and attention mechanism is the first to be applied to the sEMG-based hand gesture recognition field.

The organization of the rest of the paper is as follows. The details of the hybrid model are fully described in Section 2. More information on the experimental environment and data processing is in Section 3. Section 4 mainly covers the final experimental results, comparisons between previous studies, and discussion. Section 5 comes with the conclusion.

2 Methodology

The proposed SE-CNN attention architecture is composed of three parts as illustrated in Fig. 1: 1) a basic CNN architecture, 2) a temporal Squeeze-and-Excite block, and 3) a simple attention mechanism.

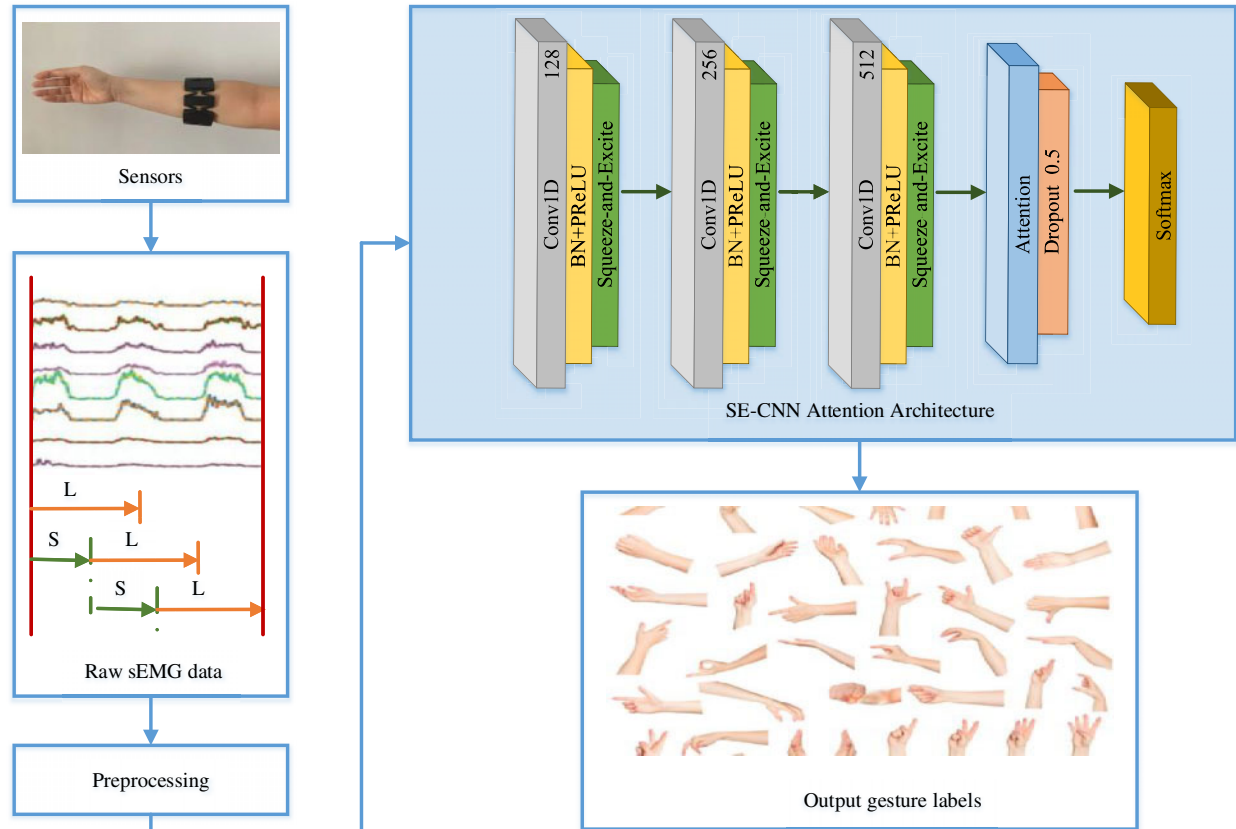


Figure 1: Schematic diagram of the proposed end-to-end framework (SE-CNN attention architecture). It enhances the generalization ability and provides a higher recognition rate for sEMG-based gesture recognition

2.1 The Basic CNN Architecture

The proposed basic CNN architecture consists of three components:

- Three Conv1D layers: The main parameters of the first Conv1D layer are: filter = 128, kernel_size = 8, padding = 'same', and kernel_initializer = 'he_uniform'. The second Conv1D layer's parameters are: filters = 256, kernel_size = 5, padding = 'same', and kernel_initializer = 'he_uniform'. The third Conv1D layer's parameters are: filters = 512, kernel_size = 3, padding = 'same', and kernel_initializer = 'he_uniform'.
- Batch Normalization [33,34]: BN was first introduced by Google in 2015. It is a technique for training deep neural networks. BN accelerates the convergence, and more importantly, alleviates the vanishing gradient problem. It simplifies the neural network model and stabilizes the inputs of a layer for each mini-batch. Therefore, BN has become the standard technology for almost

all CNN models. The training procedure is as follows:

First, we compute the average of a mini_batch with input x : $B = \{x_{1,...,m}\}$ as follows:

$$\mu_B \leftarrow \frac{1}{m} \sum_{i=1}^m x_i. \quad (1)$$

Then, we calculate the variance of the mini-batch with:

$$\delta_B^2 \leftarrow \frac{1}{m} \sum_{i=1}^m (x_i - \mu_B)^2, \quad (2)$$

and followed by normalization:

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\delta_B^2 + \epsilon}}. \quad (3)$$

Then scaling and migration are applied to Eq. (4) as follows:

$$y_i \leftarrow \gamma \hat{x}_i + \beta = BN_{\gamma, \beta}(x_i), \quad (4)$$

where γ and β are the learned parameters. Finally, the normalized network response is returned as $y_i = BN_{\gamma, \beta}(x_i)$.

- Parametric Rectified Linear Unit [35,36]: PReLU was introduced by He et al. in 2015 [35], PReLU. The authors declare that PReLU plays a key role in classification on ImageNet, which is way beyond manual classification. PReLU is an unsaturated activation function that generalizes the traditional rectified unit with a slope of negative values. It shows great advantages in 1) resolving the gradient vanishing problem, and 2) accelerating the conjugation process. The definition of PReLU is as follows:

$$f(y_i) = \begin{cases} y_i, & \text{if } y_i > 0 \\ a_i y_i, & \text{if } y_i \leq 0 \end{cases}, \quad (5)$$

where y_i is the input of the nonlinear activation f on the i -th channel; a_i is the parameter that dominates the partial gradient. The subscript i represents the variance of the nonlinear activation on different channels.

2.2 Temporal Squeeze-and-Excite

Squeeze-and-Excite Network [37] was developed by the auto-driving company Momenta in 2017. It is a novel image recognition architecture and is designed to improve accuracy by modeling the correlation between feature channels as well as enhancing the representation of the key features. SENet is modified as Fig. 2 when dealing with temporal sequence data. The transformation function F_{tr} transforms the input \mathbf{X} to feature \mathbf{U} , where $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_C]$ and \mathbf{u}_c ($c = 1, 2, \dots, C$) is calculated as follows:

$$\mathbf{u}_c = \mathbf{v}_c * \mathbf{X} = \sum_{s=1}^{C'} \mathbf{v}_c^s * \mathbf{X}^s, \quad (6)$$

where \mathbf{v}_c^s denotes the 1D spatial kernel; $*$ denotes the convolution operation. \mathbf{U} passes through two different branches and reaches $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \dots, \tilde{\mathbf{X}}_C]$ as shown in Fig. 2. For the lower branch, \mathbf{U} firstly goes through a squeeze operation to produce a channel descriptor as follows:

$$\mathbf{z}_c = \mathbf{F}_{sq}(\mathbf{u}_c) = \frac{1}{T} \sum_{t=1}^T \mathbf{u}_c(t), \quad (7)$$

where $\mathbf{z} \in \mathbf{R}^C$ and T denotes the temporal dimension. After the excitation operation, each channel is added with produced weights as follows:

$$\mathbf{s} = \mathbf{F}_{ex}(\mathbf{z}, \mathbf{W}) = \sigma(g(\mathbf{z}, \mathbf{W})) = \sigma(\mathbf{W}_2 \delta(\mathbf{W}_1 \mathbf{z})), \quad (8)$$

where \mathbf{F}_{ex} is parameterized as a neural network; σ is the sigmoid activation function; δ is the ReLU activation function. The learnable parameters of \mathbf{F}_{ex} are $\mathbf{W}_1 \in \mathbf{R}^{\frac{C}{r} \times C}$ and $\mathbf{W}_2 \in \mathbf{R}^{\frac{C}{r} \times C}$, where r denotes the reduction ratio.

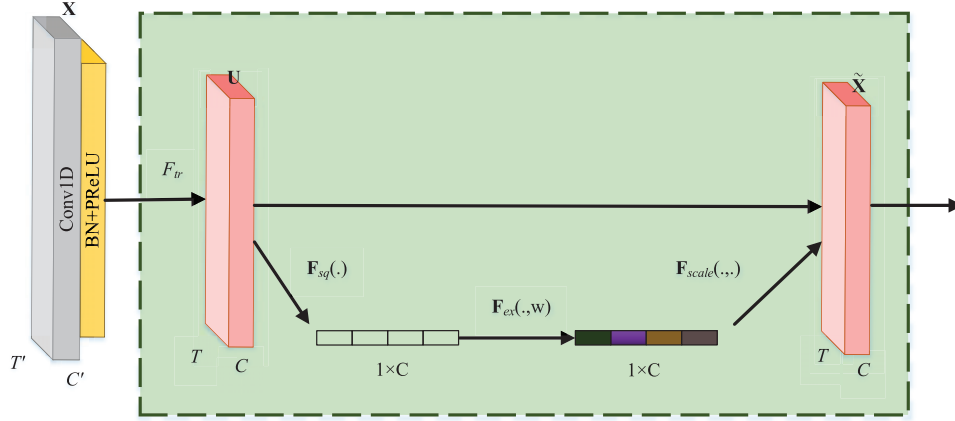


Figure 2: The computation of the temporal squeeze-and-excite block

At last, the resulting weight multiplies with the feature \mathbf{U} from the upper branches to obtain the SE block output as follows:

$$\tilde{\mathbf{X}}_c = \mathbf{F}_{scale}(\mathbf{u}_c, \mathbf{s}_c) = \mathbf{s}_c \cdot \mathbf{u}_c. \quad (9)$$

Then, we get $\tilde{\mathbf{X}} = [\tilde{\mathbf{X}}_1, \tilde{\mathbf{X}}_2, \dots, \tilde{\mathbf{X}}_C]$ as the final result.

2.3 Attention Mechanism

The schematic diagram of the attention mechanism in this article is shown in Fig. 3. The output of the previous neural network is $h \in \mathbf{R}^{C \times T}$, where C represents the number of channels and T is the number of features. h is then transformed into a $T \times C$ matrix. The softmax activation function is applied to get a $T \times C$ attention matrix by inserting each row from h into a standardized fully connected layer as follows:

$$\alpha = \text{Softmax}(\mathbf{W}_{ht} h^C + b_{ct}), \quad (10)$$

where the learned weight is $\mathbf{W}_{ht} \in \mathbf{R}^{C \times C}$ and $b \in \mathbf{R}$ denotes the biases across the channel. Based on these, we can get the learned temporal mask α^C for every feature. Hadamard product is applied to α^C with h afterward to get g as follows:

$$g = \alpha^C \circ h, g \in \mathbf{R}^{C \times T}, \quad (11)$$

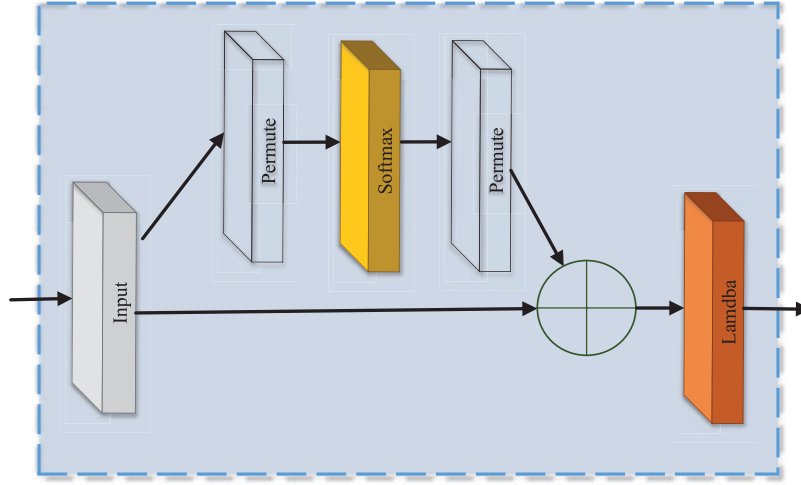


Figure 3: The attention mechanism

Finally, the output $c \in R^{1 \times T}$ of the attention mechanism is calculated through cross-channel summation as follows:

$$c = \sum_{t=1}^c g, c \in R^{1 \times T}. \quad (12)$$

3 Experimental Configurations

3.1 Datasets

For a fair comparison of existing models, we use the same datasets from reference [32] as our test set: the Cometa + Dormo dataset (known as Ninapro DB4) and the Double Myo dataset (known as Ninapro DB5) [38].

The Ninapro DB4 dataset has collected signals from muscle activities through 12 active single-differential wireless electrodes of Cometa. Corresponding to the brachioradialis joint, 8 electrodes are placed around the forearm and evenly distributed. Another 2 electrodes are placed at the joint's main movement points of the finger flexor and extensor muscles. The last 2 are placed on the main movement points of the biceps and triceps. 10 testers with 52 gesture movements, including an additional rest condition, are included in the Ninapro DB4 dataset. The sampling frequency of surface EMG is 2 kHz during data collection. Each movement is repeated 6 times.

The Ninapro DB5 dataset has collected muscle activity signals through two Thalmic Myo armbands which contain 16 active single-differential wireless electrodes. The first Myo armband is placed close to the elbow with the first transducer placed on the humeral joint. Closer to the hand, the second Myo armband is placed after the first one with a 22.5-degree angle. The sampling frequency is 200 Hz when collecting surface EMG signals. 10 testers with 52 gesture movements, including an additional rest condition, are included in the dataset. Each movement is repeated 6 times.

3.2 Preprocessing

Based on the algorithm by Josephs et al. [32], the raw data is separated into 260 ms windows with 235 ms of overlap at first. Then, a 20 Hz high-pass Butterworth filter is applied to get rid of the artifacts [39]. Finally, a moving average filter smooths the signal. These approaches reduce the windows' time step T from 52 to 38 for Ninapro DB5 and from 520 to 371 for Ninapro DB4.

3.3 Training Environment and Details

- Training Environment: All the experiments are implemented in Python 3.7, Tensorflow_gpu 2.4, and Keras 2.4 on WIN 64 with Intel® Core™ i7-8700H CPU @3.2 GHz 3.19 GHz, 64G RAM, NVIDIA GeForce GTX 1080Ti 11 GB, CUDA 10.1.243, and CuDNN v7.6.3.30.
- Training Details: The batch size is set to 128 and the learning rate anneals from 10^{-3} to 10^{-5} during the training process. As the accuracy decreases after 80–100 epochs with no further improvement as shown in Fig. 4, we stop early at 100 epochs. The Ranger optimizer is implemented in the proposed algorithm for network optimization. It combines two newly developed variants, RAdam and Lookahead [40], into a single optimizer. RAdam provides the best starting point for the optimizer at the early stage of training. It uses a dynamic rectifier to adjust Adam's adaptive momentum according to the variance and then comes up with an effective automatic warm-up procedure for the current dataset to ensure a steady training start. LookAhead is inspired by the recent understanding of the deep neural network's lost surface. It helps achieve a stable and effective exploration during the training process. Both RAdam and LookAhead provide great breakthroughs in different aspects of deep learning optimization, so the combination is highly synergistic.

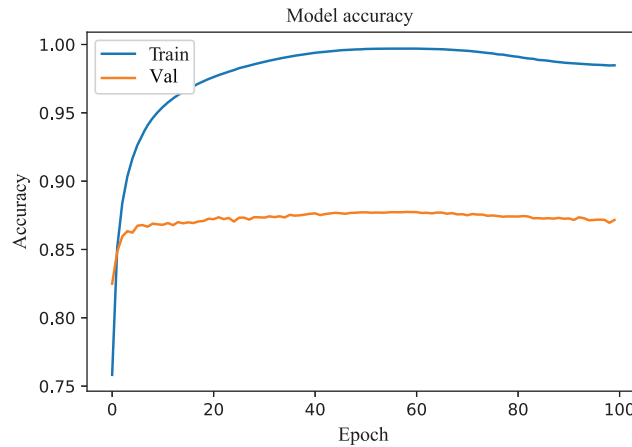


Figure 4: The model learning curve

4 Results and Discussion

4.1 Intra-Session Validation

The details of experimental specifications for Ninapro DB4 and Ninapro DB5 datasets are presented in Table 1. For comparison, referring to Josephs et al. [32], this article takes each gesture movement's 1st, 2nd, 4th, and 6th repetitions as the training set, the 3rd repetitions as the validation set, and the 5th as testing set since the 5th repetitions are commonly used for testing [24,29,32,38]. We report the performance of our model in recognition of all hand gestures in Ninapro DB4 and Ninapro

DB5 datasets, wrist and functional gestures in Ninapro DB5, and just wrist gestures in Ninapro DB5. The results and comparisons with previous studies are shown in [Table 2](#).

Table 1: Specifications of the sEMG datasets

| Dataset | Number of gestures | Number of finger gestures | Number of wrist gestures | Number of functional (grasping) gestures | Intact subjects | Number of sEMG channels | Number of trials | Trials for training | Trials for validation | Trials for testing | Sampling rate |
|-------------|--------------------|---------------------------|--------------------------|------------------------------------------|-----------------|-------------------------|------------------|---------------------|-----------------------|--------------------|---------------|
| Ninapro DB4 | 53 | 12 | 17 | 23 | 10 | 12 | 6 | 1, 2, 4, 6 | 3 | 5 | 2000 Hz |
| Ninapro DB5 | 53 | 12 | 17 | 23 | 10 | 16 | 6 | 1, 2, 4, 6 | 3 | 5 | 200 Hz |

Table 2: The comparison between the proposed model and baseline models

| Database | Method | Accuracy |
|--------------------------------------------|-----------------------|---------------|
| Ninapro DB5, all gestures | Proposed method | 87.42% |
| | Josephs et al. [32] | 87.09% |
| | Shen et al. [41] | 72.09% |
| Ninapro DB5, wrist and functional gestures | Proposed method | 87.80% |
| | Josephs et al. [32] | 87.13% |
| | Wei et al. [29] | 90% |
| Ninapro DB5, just wrist gestures | Proposed method | 89.54% |
| | Josephs et al. [32] | 89.18% |
| | Chen et al. [30] | 67.42% |
| | Wu et al. [42] | 62% |
| Ninapro DB4, all gestures | Proposed method | 77.61% |
| | Josephs et al. [32] | 74.88% |
| | Pizzolato et al. [38] | 69.01% |
| | Wei et al. [29] | 60% |

From [Table 2](#), we can see that the proposed model achieves an accuracy of 87.42% for classifying all gestures in the Ninapro DB5 dataset. It is 0.33% higher than the best performance demonstrated by Josephs et al. [32] and 15.33% higher than Shen et al. [41]. The proposed model achieves a recognition rate of 87.80% in the wrist and functional gestures in the Ninapro DB5 dataset, only 2.2% lower than Wei et al. [29]. Yet compared with Josephs et al. [32], the recognition rate of our model increases by 0.67%. Also, it is not difficult to find that the proposed model achieves the highest recognition rate of 89.54% among all the related algorithms on just wrist gestures in the Ninapro DB5 dataset, which is 0.36% higher than Josephs et al. [32], 22.12% higher than Chen et al. [30], and 27.54% higher than Wu et al. [42]. [Table 2](#) also indicates that the proposed model achieves the best performance among all models and has a recognition rate of 77.61% for all gestures in the Ninapro DB4 dataset, which is

2.73% higher than Josephs et al. [32], 8.6% higher than Pizzolato et al. [38], and 17.61% higher than Wei et al. [29].

Accuracy is the most straightforward and efficient metric to evaluate a model. However, sometimes, accuracy can be misleading, especially for imbalanced datasets. As shown in Fig. 5 and Table 3, Category 0 has more samples than others. Therefore, accuracy may cause unsatisfied results even if it is high. The trained model is meaningless if the prediction is always 0 regardless of inputs. To avoid this, additional evaluation metrics need to be involved, such as precision, recall, and F1-score. Precision evaluates the percentage of true positive examples in all predicted positive examples. Recall tells the possibility of how many positive examples are successfully retrieved. F1-score is the combination of precision and recall. In this case, higher precision and recall (higher F1-score) indicate better performance.

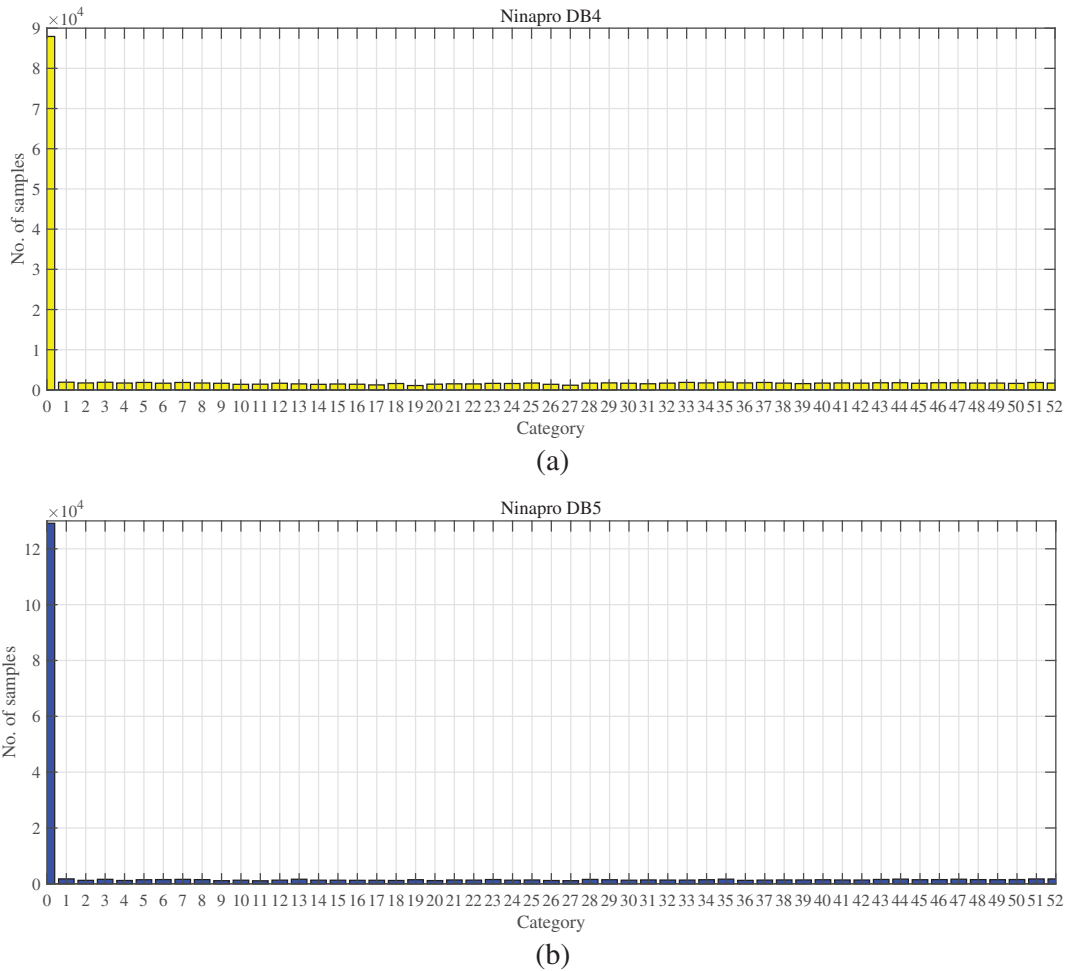
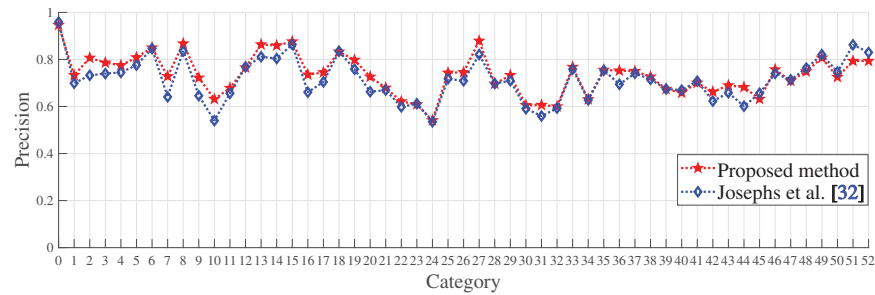


Figure 5: The distribution of experiment datasets: (a) The numbers of test samples in different categories in the Ninapro DB4 dataset; and (b) The numbers of test samples in different categories in the Ninapro DB5 dataset

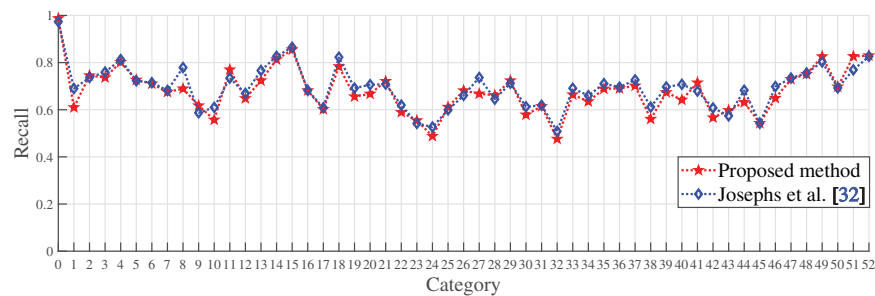
Table 3: The number of test samples per category in Ninapro DB4 and Ninapro DB5 datasets

| | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|-------------|--------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|------|
| Category | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | |
| Ninapro DB4 | 87936 | 1916 | 1733 | 1905 | 1712 | 1859 | 1669 | 1855 | 1718 | 1650 | 1398 | 1426 | 1660 | 1502 | 1391 | 1471 | 1416 | 1243 | 1589 | 1089 | 1431 | 1518 | 1495 | 1631 | 1596 | 1711 | |
| Ninapro DB5 | 129146 | 1763 | 1238 | 1606 | 1190 | 1458 | 1515 | 1591 | 1507 | 1119 | 1289 | 1087 | 1310 | 1640 | 1298 | 1290 | 1254 | 1271 | 1197 | 1461 | 1160 | 1377 | 1322 | 1528 | 1297 | 1374 | |
| Category | 26 | 27 | 28 | 29 | 30 | 31 | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 |
| Ninapro DB4 | 1396 | 1170 | 1685 | 1746 | 1660 | 1535 | 1692 | 1868 | 1745 | 1933 | 1742 | 1853 | 1697 | 1574 | 1700 | 1731 | 1687 | 1777 | 1794 | 1652 | 1789 | 1782 | 1711 | 1700 | 1626 | 1848 | 1699 |
| Ninapro DB5 | 1177 | 1120 | 1573 | 1465 | 1296 | 1385 | 1349 | 1340 | 1466 | 1660 | 1240 | 1337 | 1388 | 1382 | 1449 | 1374 | 1357 | 1553 | 1679 | 1466 | 1508 | 1648 | 1507 | 1476 | 1527 | 1741 | 1745 |

Fig. 6 shows the comparison of precision, recall, and F1-score between the proposed algorithm and Josephs et al. [32] (with the best performance so far) for all gestures in the Ninapro DB5 dataset. We find that both algorithms have close performance. Also, they both show poor performance in gesture categories 22, 23, and 24 (known as the 10th, 11th, and 12th categories in wrist gestures), as well as 30, 31, and 32 (known as the 1st, 2nd, and 3rd categories in functional gestures), resulting in relatively low precision, recall, and F1-score. Further experiments indicate that the 10th, 11th, and 12th gesture categories are wrist pronation (axis: middle), wrist supination (axis: little finger), and wrist pronation (axis: little finger). They are all related to muscle activities and cause poor performance in gesture recognition. There is also a strong correlation between muscle activity in the 9th (Wrist supination, axis: middle finger) and 10th (Wrist supination, axis: middle finger) gesture categories. Similarly, the 1st, 2nd, and 3rd categories in functional gestures, known as the large diameter grasp, the small diameter grasp (power grip), and the fixed hook grasp, are also somehow affected by muscle functions, and therefore, causing incorrect prediction. This may explain the reason why the precision, recall, and F1-score are low when testing the proposed model.



(a)



(b)

Figure 6: (Continued)

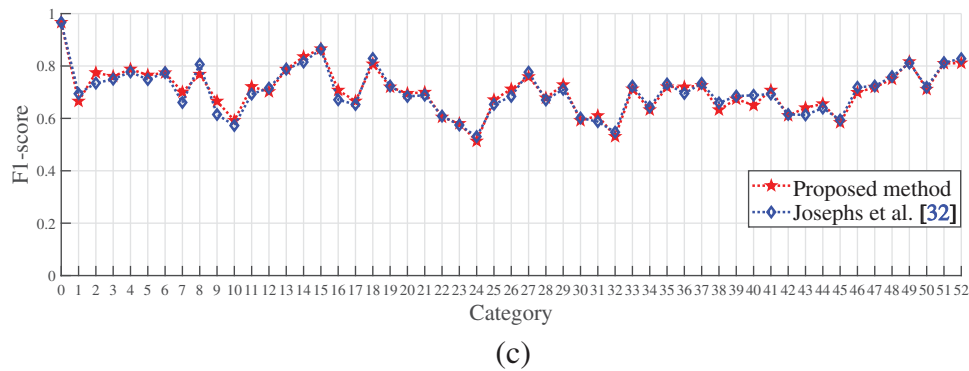


Figure 6: The performance of the algorithms for all gestures from this article and Josephs et al. [32] on the Ninapro DB5 dataset: (a) Precision; (b) Recall; (c) F1-score

Fig. 7 shows the comparison of the performance between the algorithm from this article and Josephs et al. [32] in the wrist and functional gestures in the Ninapro DB5 dataset. We analyze the precision, recall, and F1-score for both algorithms, and find that the performance of the proposed model is superior to Josephs et al. [32]. We also notice that both algorithms show poor performance in wrist gestures in terms of the 10th, 11th, and 12th categories, and functional gestures in terms of the 1st, 2nd, and 3rd categories based on their precision, recall, and F1-score.

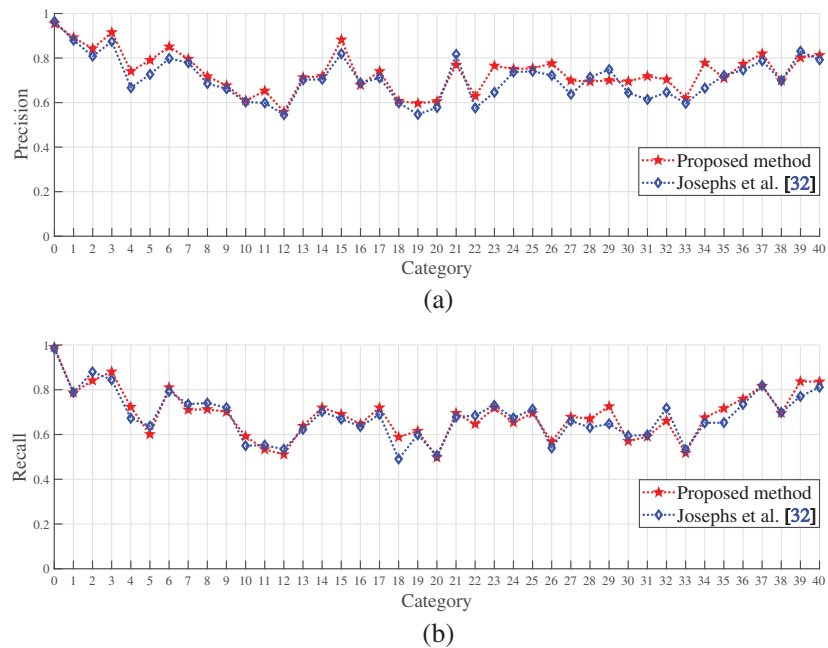


Figure 7: (Continued)

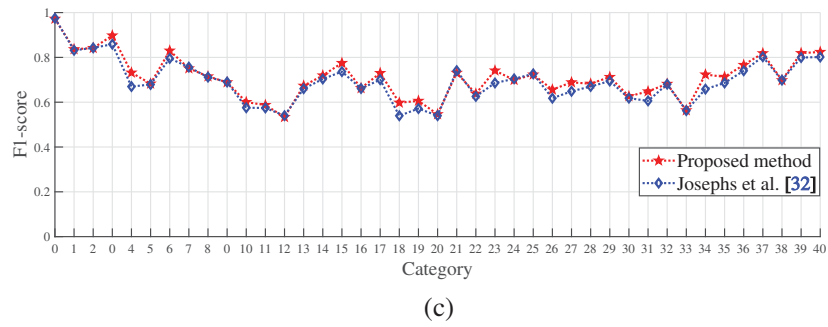


Figure 7: The performance of the algorithms for wrist and functional gestures from this article and Josephs et al. [32] in the Ninapro DB5 dataset: (a) Precision; (b) Recall; (c) F1-score

Fig. 8 shows the normalized confusion matrix of the algorithms from this article and Josephs et al. [32] in just wrist gestures in the Ninapro DB5 dataset. The proposed algorithm has better performance than Josephs et al. [32]. Similarly, we notice that it is highly possible to misidentify the 10th gesture as the 9th by both algorithms. The 11th gesture is more likely to be misidentified as the 10th, and the 12th gesture has the highest probability of being misidentified as the 11th.

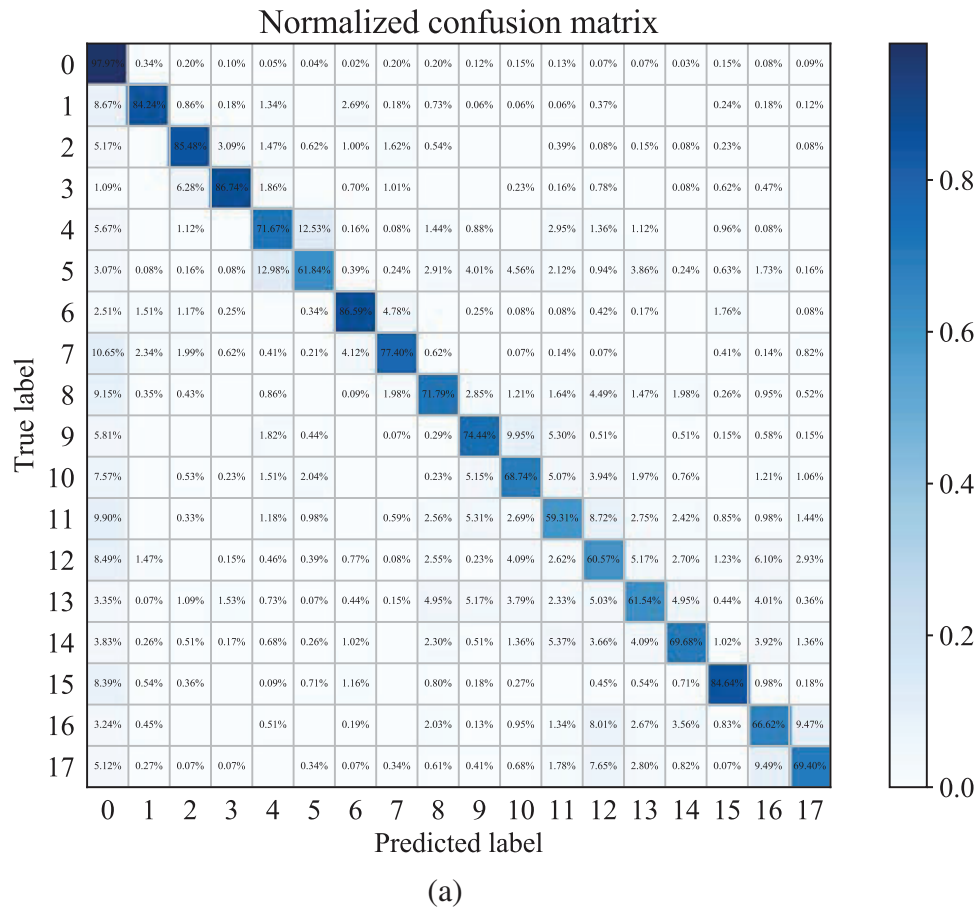


Figure 8: (Continued)

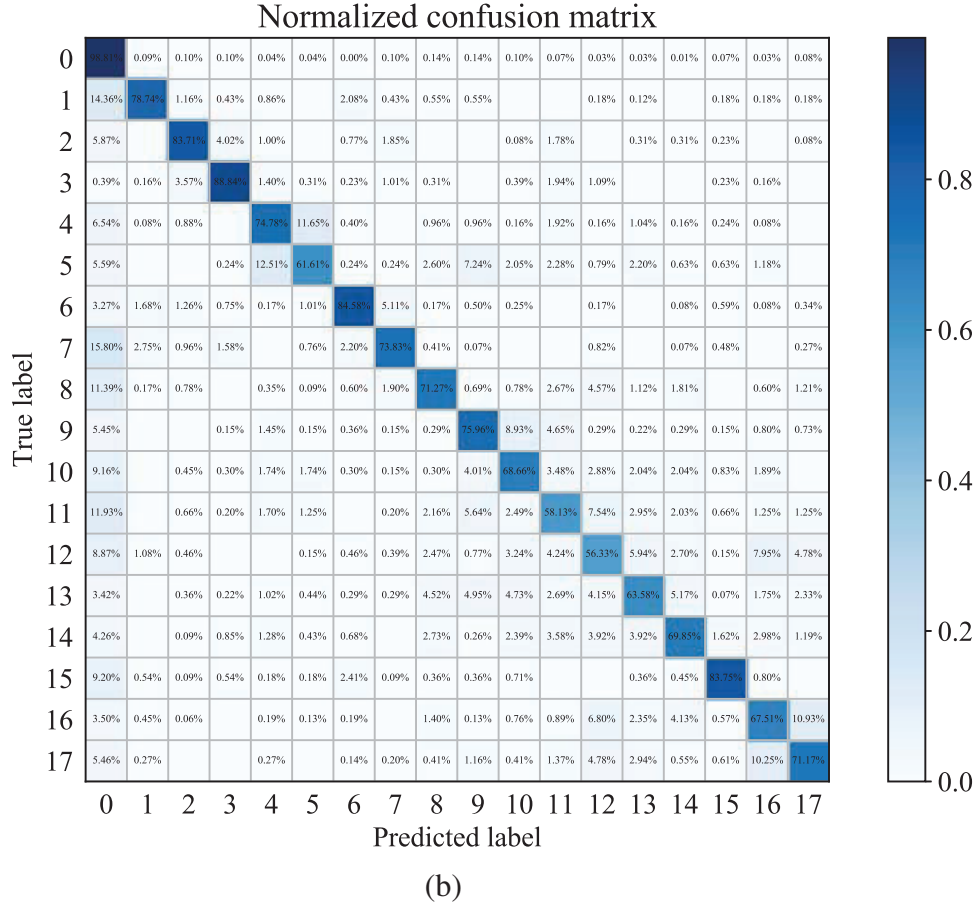


Figure 8: The performance of the algorithms for just wrist gestures from this article and Josephs et al. [32] in the Ninapro DB5 dataset: (a) the normalized confusion matrix of Josephs et al. [32]; (b) the proposed method's normalized confusion matrix

Fig. 9 shows the comparison of precision, recall, and F1-score between the algorithm from this article and Josephs et al. [32] in all gestures in the Ninapro DB4 dataset. The proposed algorithm has better precision, recall, and F1-score compared to Josephs et al. [32].

Based on Table 2 and Figs. 6–9, we can conclude that:

- The proposed algorithm achieves a relatively high recognition rate in all experiments. The recognition rate for all gestures in Ninapro DB5 is 87.42%, for wrist and functional gestures in Ninapro DB5 is 87.80%, for just wrist gestures in Ninapro DB5 is 89.54%, and for all gestures in Ninapro DB4 is 77.61%.
- Compared to the algorithm proposed by Josephs et al. [32], the proposed algorithm has a better performance in precision, recall, and F1-score. Also, we find that relative muscle activities affect the model recognition rate. The 9th, 10th, 11th, and 12th categories in wrist gestures, known as wrist supination (axis: middle finger), wrist pronation (axis: middle), wrist supination (axis: little finger), and wrist pronation (axis: little finger), have similarities in muscle activities. The 1st, 2nd, and 3rd categories in functional gestures, known as large diameter grasp, small

diameter grasp (power grip), and fixed hook grasp, also share similarities in muscle movements. Therefore, it leads to a certain decrease in the recognition rate of the above gestures.

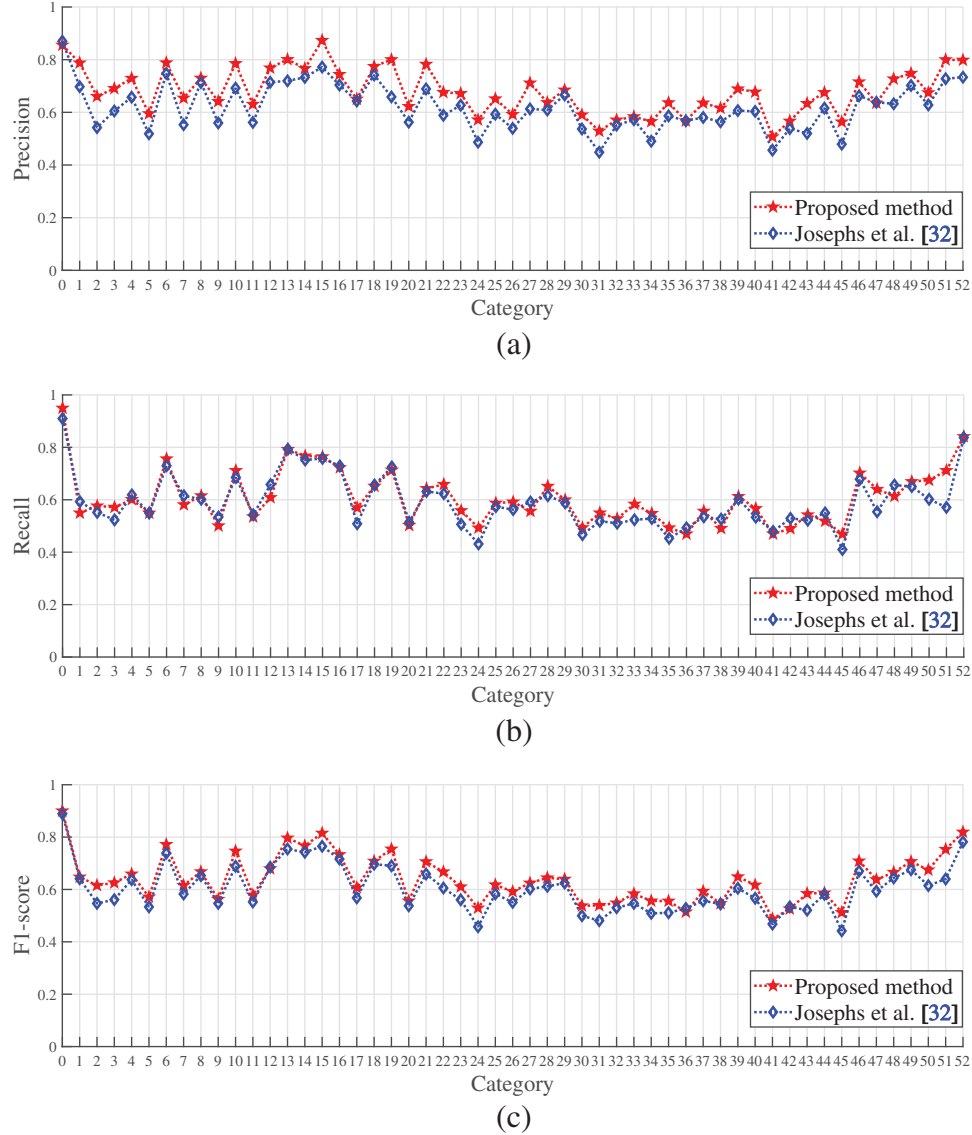


Figure 9: The performance of the algorithms for all gestures from this article and Josephs et al. [32] in the Ninapro DB4 dataset: (a) Precision; (b) Recall; (c) F1-score

4.2 Subject-Wise Cross-Validation

Poor generalization has always been an obstacle to the development of gesture recognition techniques. To further improve the generalization ability of the proposed algorithm, we include data from different subjects for training and testing. The details are shown in Fig. 10. First, ten subjects of the Ninapro DB5 dataset are divided into five groups by ordinal number. Then five-fold cross-validation is applied to verify the performance of the proposed algorithm.

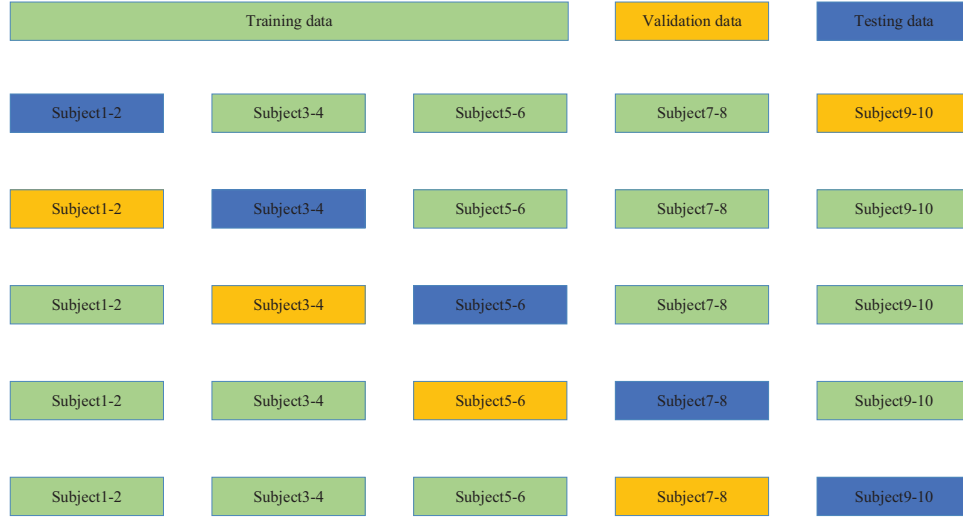


Figure 10: Division of training (green boxes), validation (yellow boxes), and testing sets (blue boxes) for five-fold cross-validation across individual subjects

From Table 4, we can find that the proposed algorithm has an average recognition rate of 65.74% for all gestures in the Ninapro DB5 dataset, 2.78% higher than Josephs et al. [32]. The average recognition rate of the proposed algorithm for wrist and functional gestures in Ninapro DB5 is 66.67%, 2.24% higher than Josephs et al. [32]. The average recognition rate of the proposed algorithm for just wrist gestures in Ninapro DB5 is 69.47%, 3.11% higher than Josephs et al. [32]. Fig. 11 gives the results of five-fold cross-validation. In contrast to Josephs et al. [32], the proposed algorithm has a higher average recognition rate with less recognition dispersion in five-fold cross-validation. In other words, the proposed algorithm has better generalization ability.

Table 4: Performance of the proposed method

| Dataset | Method | Accuracy |
|--------------------------------------------|---------------------|---------------|
| Ninapro DB5, all gestures | Proposed method | 65.74% |
| | Josephs et al. [32] | 62.96% |
| Ninapro DB5, wrist and functional gestures | Proposed method | 66.67% |
| | Josephs et al. [32] | 64.43% |
| Ninapro DB5, just wrist gestures | Proposed method | 69.47% |
| | Josephs et al. [32] | 66.36% |

4.3 Discussion

This article introduces a novel SE-CNN attention architecture for hand gesture recognition based on surface EMG signals. Adding the temporal SE module to the output of convolutional layers can efficiently highlight the expression of significant features, establish relationships among different channels, as well as illustrate the model's ability to extract features. Additionally, the attention mechanism greatly assists in capturing spatial relationships between features and enhancing

the learning ability in terms of multi-channel expression, which further boosts its robustness and recognition rates. The results of the experiments indicate that the proposed algorithm is valuable in gesture recognition systems. Based on the results above, our algorithm shows good performance in various datasets, including all gestures in Ninapro DB5 (recognition rate: 87.42%), wrist and functional gestures in Ninapro DB5 (recognition rate: 87.80%), just wrist gestures in Ninapro DB5 (recognition rate: 89.54%), and all gestures in Ninapro DB4 (recognition rate: 77.61%). The proposed algorithm has achieved the best performance compared to previous studies on Ninapro DB4, Ninapro DB5 and Ninapro DB4 datasets are two datasets from completely different sources in both data acquisition methods and sampling frequencies, but our algorithm still achieves relatively good outcomes on them. Moreover, the proposed algorithm performs well in evaluation metrics including precision, recall, and F1-score, which proves that our model has better adaptability when applying to different datasets, more relaxed requirements for data collection programs, and relatively fine classification performance when dealing with imbalanced data, ensuring the success in developing gesture recognition systems.

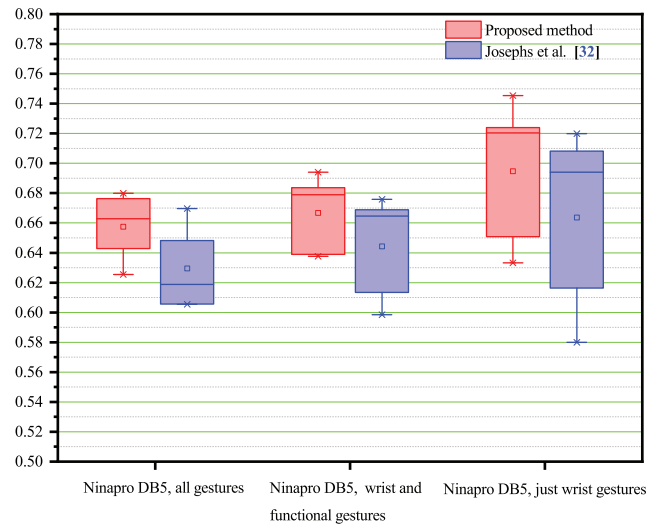


Figure 11: The performance comparison between the algorithms proposed by this article and Josephs et al. [32] in inter-subject five-fold cross-validation

Poor generalization is one of the important factors in slowing down the development of recognition systems. Herein, we conduct an inter-subject validation on all gestures, wrist and functional gestures, and just wrist gestures from Ninapro DB5 respectively. 10 subjects from either dataset are collected and divided into independent training, testing, and validation sets by five-fold cross-validation method with corresponding recognition rates of 65.74% (53 gestures), 66.67% (41 gestures), and 69.47% (18 gestures). Compared with Josephs et al. [32], our algorithm performs better in various aspects, proving its excellent generalization ability.

Meanwhile, we interestingly find that the gestures involving similar muscle activities lead to unsatisfied performance and confusion classification results, such as wrist pronation (axis: middle finger) in wrist gestures, wrist supination (axis: little finger), and wrist pronation (axis: little finger). More experiments need to be done in the future, including further analysis on sEMG signals from gestures that share similar muscle activities. Also, the neural network needs to be further optimized for better classification in gestures involving the same muscle movement by investigating the characteristics of corresponding sEMG signals. In inter-subject validation, though the proposed algorithm has achieved

a better result compared to previous studies, the recognition rate is still not very high. Therefore, further investigation is required to achieve better inter-subject validation performance.

5 Conclusion

In this article, we propose a novel SE-CNN attention architecture for sEMG-based hand gesture recognition. It consists of a basic CNN architecture, a temporal squeeze-and-excite architecture, and an attention mechanism. The main advantage of the proposed algorithm is the introduction of the temporal squeeze-and-excite block into a CNN-based gesture recognition architecture using sEMG signals. The temporal squeeze-and-excite block enhances important features by suppressing meaningless ones. Therefore, the model's feature extraction is improved. Also, an attention mechanism is added at the end of the model to enhance the capability of representation learning of multi-channel signals. We conduct two experimental paradigms: 1) intro-session validation, and 2) subject-wise cross-validation. The results from intra-session validation using Ninapro DB5 and Ninapro DB4 datasets indicate that the proposed algorithm has improved the system robustness in hand gesture recognition tasks with a better recognition rate. Further analysis using subject-wise cross-validation in all gestures, wrist and functional gestures, and just wrist gestures in the Ninapro DB5 dataset shows that the proposed algorithm has better stability and generalization ability in multi-gesture recognition, facilitating the promotion of intelligent interactive systems based on surface EMGs in practical applications. However, the algorithm proposed in this article still has limitations, especially when classifying gestures with similar muscle activities, which demotes the application of EMG gesture recognition. Therefore, the proposed algorithm still has a lot of room for improvement. Further investigation is required in the future in terms of individual myoelectricity differences during gesture recognition.

Acknowledgement: The authors would like to thank the editor and reviewers for the valuable comments and suggestions.

Funding Statement: This document is the results of the research project funded by the National Key Research and Development Program of China (2017YFB1303200), NSFC (81871444, 62071241, 62075098, and 62001240), Leading-Edge Technology and Basic Research Program of Jiangsu (BK20192004D), Jiangsu Graduate Scientific Research Innovation Programme (KYCX20_1391, KYCX21_1557).

Conflicts of Interest: The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

References

1. Sharma, R. P., Verma, G. K. (2015). Human computer interaction using hand gesture. *Procedia Computer Science*, 54, 721–727. DOI 10.1016/j.procs.2015.06.085.
2. Rautaray, S. S., Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: A survey. *Artificial Intelligence Review*, 43(1), 1–54. DOI 10.1007/s10462-012-9356-9.
3. Ren, Z., Meng, J., Yuan, J., Zhang, Z. (2011). Robust hand gesture recognition with kinect sensor. *Proceedings of the 19th ACM International Conference on Multimedia*, Scottsdale, Arizona, USA.
4. Li, G., Tang, H., Sun, Y., Kong, J., Jiang, G. et al. (2019). Hand gesture recognition based on convolution neural network. *Cluster Computing*, 22(2), 2719–2729. DOI 10.1007/s10586-017-1435-x.

5. Qi, J., Jiang, G., Li, G., Sun, Y., Tao, B. (2020). Surface emg hand gesture recognition system based on pca and grnn. *Neural Computing and Applications*, 32(10), 6343–6351. DOI 10.1007/s00521-019-04142-8.
6. Oyedotun, O. K., Khashman, A. (2017). Deep learning in vision-based static hand gesture recognition. *Neural Computing and Applications*, 28(12), 3941–3951. DOI 10.1007/s00521-016-2294-8.
7. He, J., Jiang, N. (2020). Biometric from surface electromyogram (SEMG): Feasibility of user verification and identification based on gesture recognition. *Frontiers in Bioengineering and Biotechnology*, 8, 00058. DOI 10.3389/fbioe.2020.00058.
8. Heloir, A., Nunnari, F., Bachynskyi, M. (2019). Ergonomics for the design of multimodal interfaces. In: *The handbook of multimodal-multisensor interfaces: Language processing, software, commercialization, and emerging directions*, vol. 3, pp. 263–304. San Rafael: Association for Computing Machinery and Morgan & Claypool.
9. Ginn, K. A., Cools, A., Halaki, M. (2020). Do surface electrodes validly represent lower trapezius activation patterns during shoulder tasks? *Journal of Electromyography and Kinesiology*, 53, 102427. DOI 10.1016/j.jelekin.2020.102427.
10. Ovrur, S. E., Zhou, X., Qi, W., Zhang, L., Hu, Y. et al. (2021). A novel autonomous learning framework to enhance semg-based hand gesture recognition using depth information. *Biomedical Signal Processing and Control*, 66, 102444. DOI 10.1016/j.bspc.2021.102444.
11. Tam, S., Boukadoum, M., Campeau-Lecours, A., Gosselin, B. (2019). A fully embedded adaptive real-time hand gesture classifier leveraging hd-semg and deep learning. *IEEE Transactions on Biomedical Circuits and Systems*, 14(2), 232–243. DOI 10.1109/TBCAS.4156126.
12. Junior, J. J. A. M., Freitas, M. L., Siqueira, H. V., Lazzaretti, A. E., Pichorim, S. F. et al. (2020). Feature selection and dimensionality reduction: An extensive comparison in hand gesture classification by semg in eight channels armband approach. *Biomedical Signal Processing and Control*, 59, 101920. DOI 10.1016/j.bspc.2020.101920.
13. Arozi, M., Caesarendra, W., Ariyanto, M., Munadi, M., Setiawan, J. D. et al. (2020). Pattern recognition of single-channel semg signal using pca and ann method to classify nine hand movements. *Symmetry*, 12(4), 541. DOI 10.3390/sym12040541.
14. Khezri, M., Jahed, M. (2010). A neuro-fuzzy inference system for semg-based identification of hand motion commands. *IEEE Transactions on Industrial Electronics*, 58(5), 1952–1960. DOI 10.1109/TIE.2010.2053334.
15. Xue, Y., Ji, X., Zhou, D., Li, J., Ju, Z. (2019). Semg-based human in-hand motion recognition using nonlinear time series analysis and random forest. *IEEE Access*, 7, 176448–176457. DOI 10.1109/Access.6287639.
16. Wang, S. H., Jiang, X., Zhang, Y. D. (2021). Multiple sclerosis recognition by biorthogonal wavelet features and fitness-scaled adaptive genetic algorithm. *Frontiers in Neuroscience*, 15, 737785. DOI 10.3389/fnins.2021.737785.
17. Chen, W., Zhang, Z. (2019). Hand gesture recognition using semg signals based on support vector machine. *2019 IEEE 8th Joint International Information Technology and Artificial Intelligence Conference (ITAIC)*, IEEE, Chongqing, China.
18. Gijssberts, A., Atzori, M., Castellini, C., Müller, H., Caputo, B. (2014). Movement error rate for evaluation of machine learning methods for semg-based hand movement classification. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 22(4), 735–744. DOI 10.1109/TNSRE.7333.
19. Ding, Z., Yang, C., Tian, Z., Yi, C., Fu, Y. et al. (2018). Semg-based gesture recognition with convolution neural networks. *Sustainability*, 10(6), 1865. DOI 10.3390/su10061865.
20. Wang, S. H., Khan, M. A., Zhang, Y. D. (2022). Vispnn: Vgg-inspired stochastic pooling neural network. *Computers, Materials & Continua*, 70(2), 3081–3097. DOI 10.32604/cmc.2022.019447.
21. Wang, S. H., Zhang, X., Zhang, Y. D. (2021). Dssae: Deep stacked sparse autoencoder analytical model for covid-19 diagnosis by fractional Fourier entropy. *ACM Transactions on Management Information System*, 13(1), 1–20. DOI 10.1145/3451357.

22. Zhang, Z., Geiger, J., Pohjalainen, J., Mousa, A. E. D., Jin, W. et al. (2018). Deep learning for environmentally robust speech recognition: An overview of recent developments. *ACM Transactions on Intelligent Systems and Technology*, 9(5), 1–28. DOI 10.1145/3178115.
23. Park, K. H., Lee, S. W. (2016). Movement intention decoding based on deep learning for multiuser myoelectric interfaces. *International Winter Conference on Brain-Computer Interface*, IEEE, Gangwon, Korea (South).
24. Atzori, M., Cognolato, M., Müller, H. (2016). Deep learning with convolutional neural networks applied to electromyography data: A resource for the classification of movements for prosthetic hands. *Frontiers in Neurobotics*, 10, 9. DOI 10.3389/fnbot.2016.00009.
25. Wei, W., Wong, Y., Du, Y., Hu, Y., Kankanhalli, M. et al. (2019). A multi-stream convolutional neural network for semg-based gesture recognition in muscle-computer interface. *Pattern Recognition Letters*, 119, 131–138. DOI 10.1016/j.patrec.2017.12.005.
26. Tsinganos, P., Cornelis, B., Cornelis, J., Jansen, B., Skodras, A. (2018). Deep learning in emg-based gesture recognition. *5th International Conference on Physiological Computing Systems*.
27. Hu, Y., Wong, Y., Wei, W., Du, Y., Kankanhalli, M. et al. (2018). A novel attention-based hybrid cnn-rnn architecture for semg-based gesture recognition. *PLoS One*, 13(10), e0206049. DOI 10.1371/journal.pone.0206049.
28. Côté-Allard, U., Fall, C. L., Drouin, A., Campeau-Lecours, A., Gosselin, C. et al. (2019). Deep learning for electromyographic hand gesture signal classification using transfer learning. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(4), 760–771. DOI 10.1109/TNSRE.7333.
29. Wei, W., Dai, Q., Wong, Y., Hu, Y., Kankanhalli, M. et al. (2019). Surface-electromyography-based gesture recognition by multi-view deep learning. *IEEE Transactions on Biomedical Engineering*, 66(10), 2964–2973. DOI 10.1109/TBME.10.
30. Chen, L., Fu, J., Wu, Y., Li, H., Zheng, B. (2020). Hand gesture recognition using compact CNN via surface electromyography signals. *Sensors*, 20(3), 672. DOI 10.3390/s20030672.
31. Little, M. A., Varoquaux, G., Saeb, S., Lonini, L., Jayaraman, A. et al. (2017). Using and understanding cross-validation strategies. Perspectives on Saeb et al. *GigaScience*, 6(5), 1–6. DOI 10.1093/giga-science/gix020.
32. Josephs, D., Drake, C., Heroy, A., Santerre, J. (2020). Semg gesture recognition with a simple model of attention. *Proceedings of Machine Learning Research*, Virtual.
33. Chen, L., Fei, H., Xiao, Y., He, J., Li, H. (2017). Why batch normalization works? A buckling perspective. *2017 IEEE International Conference on Information and Automation (ICIA)*, IEEE, Macao, China.
34. Zhang, Y. D., Dong, Z. C., Wang, S. H., Yu, X., Yao, X. J. et al. (2020). Advances in multimodal data fusion in neuroimaging: Overview, challenges, and novel orientation. *Information Fusion*, 64, 149–187. DOI 10.1016/j.inffus.2020.07.006.
35. He, K., Zhang, X., Ren, S., Sun, J. (2015). Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision*, Santiago, Chile.
36. Wang, S. H., Chen, Y. (2020). Fruit category classification via an eight-layer convolutional neural network with parametric rectified linear unit and dropout technique. *Multimedia Tools and Applications*, 79(21), 15117–15133. DOI 10.1007/s11042-018-6661-6.
37. Hu, J., Shen, L., Sun, G. (2018). Squeeze-and-excitation networks. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, Utah.
38. Pizzolato, S., Tagliapietra, L., Cognolato, M., Reggiani, M., Müller, H. et al. (2017). Comparison of six electromyography acquisition setups on hand movement classification tasks. *PLoS One*, 12(10), e0186132. DOI 10.1371/journal.pone.0186132.

39. Hua, W. S., Celebi, M. E., Dong, Z. Y., Xiang, Y., Yuan, L. S. et al. (2021). Advances in data preprocessing for biomedical data fusion: An overview of the methods, challenges, and prospects. *Information Fusion*, 76, 376–421. DOI 10.1016/j.inffus.2021.07.001.
40. Bao, Y., An, Y., Cheng, Z., Jiao, R., Zhu, C. et al. (2020). Named entity recognition in aircraft design field based on deep learning. *International Conference on Web Information Systems and Applications*, Springer, Guangzhou, China.
41. Shen, S., Gu, K., Chen, X. R., Yang, M., Wang, R. C. (2019). Movements classification of multi-channel semg based on cnn and stacking ensemble learning. *IEEE Access*, 7, 137489–137500. DOI 10.1109/Access.6287639.
42. Wu, Y., Zheng, B., Zhao, Y. (2018). Dynamic gesture recognition based on LSTM-CNN. *2018 Chinese Automation Congress (CAC)*, IEEE, Xi'an, China.