International Journal of Machine Learning and Cybernetics (2022) 13:1729–1738
https://doi.org/10.1007/s13042-021-01482-7

ORIGINAL ARTICLE

sEMG based hand gesture recognition with deformable convolutional network

Hao Wang[1] · **Yue Zhang**[1] · **Chao Liu**[2] · **Honghai Liu**[3]

Received: 24 March 2021 / Accepted: 15 November 2021 / Published online: 17 January 2022

Abstract
There is a growing interest in human machine interface and their applications using surface electromyography (sEMG). sEMG based gesture recognition plays a crucial role in interfacing with peripheral devices such as prosthetic hands. Give the challenges in the state of the art of sEMG based gesture recognition using deep learning, we propose a deformable convolutional network (DCN) to optimise the conventional convolution kernels with a goal of achieving better performance of sEMG based gesture recognition. The DCN first apply traditional convolutional layer to obtain low-dimensional feature maps, then use deformable convolutional layer to get high-dimensional feature maps. Moreover, we propose and compare two new image representation methods based on traditional feature extraction, which enable deep learning architectures to extract implicit correlations between different channels from the sparse multichannel sEMG signals. The experiments are conducted to evaluate the proposed methods on three groups of different types and numbers of gestures on the Ninapro-DB1 data set, the proposed DCN has an improvement of 1.1%, 2.6%, and 4.9% compared with traditional CNN, respectively. In addition, the results of experiments indicate that the DCN shows robustness and feasibility in both feature extraction and classification recognition for the sEMG based gesture recognition.

Keywords sEMG · Feature extraction · Deformable convolution · Feature representation

1 Introduction

Surface electromyography (sEMG) is an electrodiagnostic medical technique for evaluating and recording the electrical activity produced by skeletal muscles. The sEMG signal detects the electric potential that was generated by muscle cells when these cells are activated electrically or neurologically, and the signal can be obtained through electrodes attached to the skin surface. According to the difference in the muscle area and degree used by different gestures, the acquired the sEMG waveforms are also different. On this account, researchers can achieve the recognition of different gestures. Namely, gesture recognition based on the sEMG signal is realized by using the difference between the sEMG signal generated by muscles in various gesture scenarios. Gesture recognition, which based on the sEMG can realize the control of external peripherals [1], such as prostheses [2, 3], rehabilitation equipment, etc. Thereby it can bring convenience to the lives of the disabled and elder who have mobility impairments. In addition, it can provide a novel way of human-computer interaction (HCI) [4, 5].

The traditional sEMG based gesture recognition is realized by pattern recognition method, which generally consists of the following three steps [6]: signal pre-processing [7], feature extraction [8] and model classification [9–11]. Signal pre-processing generally reserves the effective signal, removes or weakens the invalid signal by filtering technology, thereby improving the signal recognition. Darak et al. [12] shown the effects of different filtering techniques in removing electrocardiogram (ECG) interference in the sEMG. For example, an event-synchronous interference canceller (ESC) obtains a clean sEMG signal, but requires an additional input as a reference input. Using independent component analysis (ICA) has the ability to separate

✉ Hao Wang
haowell.wang@gmail.com

Chao Liu
1121@whut.edu.cn

[1] College of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China

[2] Wuhan University of Techonology, Wuhan 430070, China

[3] Harbin Institute of Technology, Shenzhen 518055, China

statistically independent source signals from a given set of their linear combinations. However, this method requires several recordings of the sEMG signals to attain the maximum possible accuracy. Feature extraction, mainly extracted feature vectors from the original signal to reduce the amount of calculation, and the features were often divided into time-domain features, frequency-domain features, and time-frequency domain features. Phinyomark et al. [13] compared the effects of some different feature extraction methods and their combination under the steady-state of the sEMG signals. Model classification methods in pattern recognition include support vector machines (SVM), linear discriminant analysis (LDA), etc. Li et al. [14] used onset detection method to acquire transient signals for feature extraction, and compared its effect with other classification methods. In the case when the training window length and the test window length are both equal to 150 ms, this paper found the decoding accuracy has reached more than 92% with SVM. Although various methods are used in gesture recognition, the traditional methods are based on hand-crafted features and the data comes from the ideal laboratory environment, which is not universal in common scenarios, resulting in poor robustness and applicability in real scenes.

In recent years, convolutional neural networks (CNN) have become popular in classification and recognition algorithms. Because of avoiding hand-crafted feature generation, it also helps to improve accuracy and achieve even better recognition results. The three most critical elements of CNN are local receptive fields, weight sharing, and downsampling in the pooling layer. CNN was initially used in computer vision, natural language processing, and other fields, but was also used in the field of the sEMG signals, and achieved good results. Tsinganos et al. [15] used Ninapro DB-1 [16] as a data set, and adjusted the structural parameters of the CNN to improve 3% compared with the basic model. Pinzón-Arenas et al. [17] also achieved better results in classifying and recognizing six gestures by using the power spectral density map of the sEMG signal as the input of the CNN. Jia et al. [18] proposed a deep learning model that combined convolutional auto-encoder (CAE) and CNN to classify an sEMG dataset consisting of ten classes of hand gestures. The result achieved the best performance, strongest robustness, and statistical properties compared to other classifiers. Yang et al. [19] used CNN to predict the multi-degree-of-freedom (multi-DOF) of wrist movements based on the sEMG. Chen et al. [20] used the transfer learning method to improve the performance of CNN in the sEMG based gesture recognition. Yang et al. [21] applied large-size window as the input of CNN to improve the classification accuracy in two sEMG public datasets.

Although the CNN method has achieved good results in the sEMG based on gesture recognition, the direct application of the CNN method in this field still has shortcomings. The traditional CNN and the sEMG signal are both required to be adjusted to meet the requirements of other's characteristics. For example, CNN was originally designed for images, where the input is a two-dimensional image, but the sEMG is a one-dimensional signal. So Hu et al. [22] proposed a new image representation of traditional features, and get 86.3% accuracy by employing GengNet [23]. CNN extracts common features from a large amount of data, image public datasets such as imageNet can realize big data. But the sEMG data vary from person to person, and the amount of data is limited. So Jiang et al. [24] proposed the signal image (SI) method to expand signal sequences. In addition, the convolution kernel is another very important element in CNN, because the features are extracted through the continuous movement of the convolution kernel to the image. Therefore, there are a lot of improvements that are focused on convolution kernels for CNN, Yu et al. [25] improved the accuracy of image classification on VOC-2012 [26] through dilating receptive fields, Dai et al. [27] improved the accuracy of target recognition in images by offsetting convolution kernels and regions of interest. The feature extraction also has tremendous influences on hand gesture recognition based on the sEMG, so the convolution kernel needs to be adjusted to meet the requirements of the characteristics of the sEMG. Thereby we employ a novel network structure—deformable convolutional network to extract features in this paper. First, we use a traditional convolutional layer to extract the sEMG signal to obtain low-dimensional feature maps, then a deformable convolutional layer was used to deformably convolve the low-dimensional feature maps to get high-dimensional feature maps.

The structure of this paper is as follows. Section 2 has three parts in total, the first two parts briefly introduce the data and our preprocessing methods, and the third part illustrate the CNN, DCN, and the whole network structure. Section 3 presents the results and the analysis are shown. Section 4 makes the conclusions and future work.

2 Materials and methods

2.1 Data

Ninapro-DB1 is a public and widely used dataset for the sEMG based gesture recognition. The data set is obtained by positioning ten electrodes distributed sparely on the upper arm, and the sEMG signals are recorded at a sampling rate of 100 Hz. After data collection, Ninapro-DB1 dataset consists of 52 gestures from 27 intact subjects, and each subject repeated 10 times for each gesture. The 52 gestures are divided into four main classes, including 12 finger movements, 8 hand postures, 9 wrist movements, and 23 grasping and functional movements.

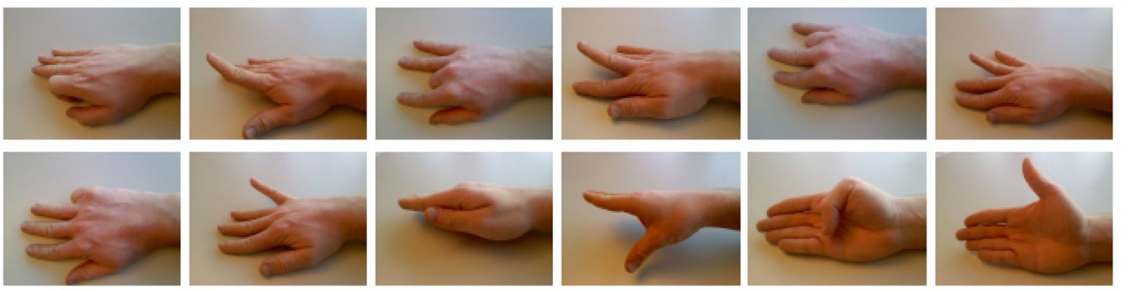
This paper uses the part of Ninapro-DB1 to conduct experiments. This part of the data contains three classes: finger movements, hand postures, and wrist movements [28], a total of 29 gestures, as shown in Fig. 1. For more information about the dataset, see [16, 24, 28].

2.2 Pre-processing

To adapt the input data to the neural network, we first preprocess the data by employing the SI algorithm. On the one hand, the input data are expanded, and on the other hand, the relationship between different channels can be learned by the network. For pre-processing, we convert the original signal into SI [24] to expand its channels. The raw signal is stacked row-by-row into the SI. In this way, every channel has the chance to be adjacent to other channels and only adjacent once, which enables the network to extract hidden correlations between neighboring channels. In short, the implementation is to number the channels first, and then circulates to make channel i adjacent to channel j if channel i and j are not adjacent. And it keeps circulating until all channels are adjacent to other channels and only adjacent once. After processing, the original 10 channels of data will be expanded to 42 channels. The SI algorithm is shown in Table 1. The original signal and its expanded result are shown in Fig. 2.

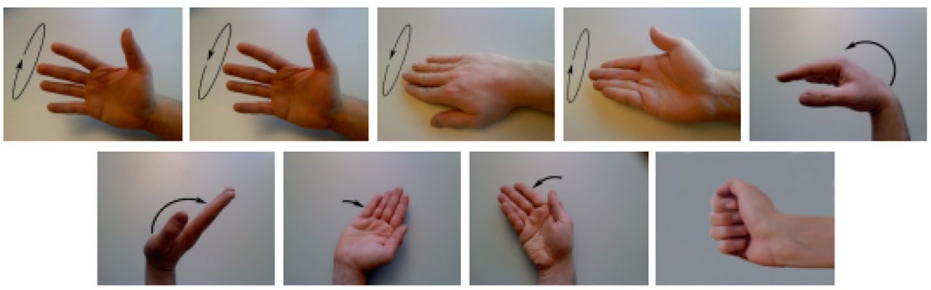
Traditional feature extraction methods are also effective ways to form features, especially to form temporal features. And it can be used as a supplement to the CNN that mainly extracts spatial features. So we combine some traditional feature extraction methods with the above expansion method as a new input to the network. In this paper, we use the mean absolute value (MAV), waveform length (WL), root mean square (RMS) methods to extract features.

For traditional feature extraction methods, the window is 100 ms and the sliding window is set to 50 ms. There are two methods to combine the process of SI expanding and feature extractions. One method is to first expand the channels in SI thus to get 42 channel data, then we conduct different feature extraction methods to get 42 channels features. In this paper, we used three methods and concatenate to 126 channels features. The other method is to conduct feature extraction firstly, by which we get ten



(a) Basic movements of the fingers (flexions and extensions).

(b) Isometric, isotonic hand configurations ("hand postures").



(c) Basic movements of the wrist.

Fig. 1 Gestures of three classes [28]

Table 1 SI Algorithm [24]

Algorithm 1 Original Signal → Signal Image

Notations:
- Signal Image (SI): a 2D array to store permutated original signals.
- Signal Index String (SIS): a string to store signal indices, whose length is N_{SIS}.

Input: N_S signal channels.
Loops:
$i = 1; j = i + 1;$
SI is initialized to be the i-th signal channel;
SIS is initialized to be $'i'; N_{SIS} = 1;$
while $i \neq j$ **do**
  if $j > N_S$ **then**
    $j = 1;$
  else if $'ij' \notin SIS \quad \&\& \quad 'ji' \notin SIS$ **then**
    Append the j-th signal channnel to the bottom of SI;
    Append $'j'$ to SIS; $N_{SIS} = N_{SIS} + 1;$
    $i = j; j = i + 1;$
  else
    $j = j + 1;$
  end if
end while
Output: SI.

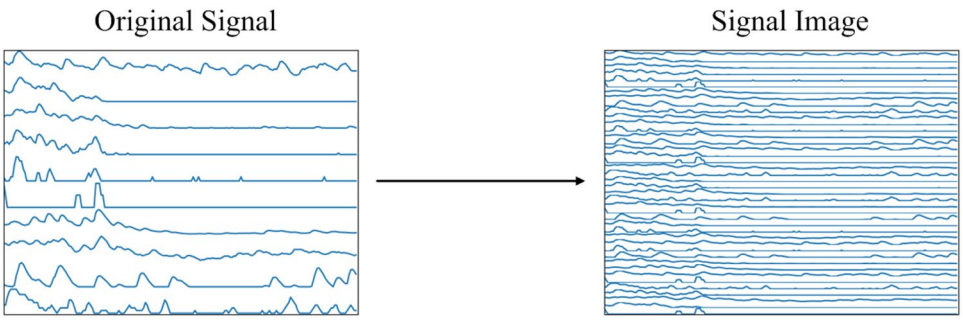


Fig. 2 Original signal and signal image

channels features and concatenated them as a map. This map contained 30 channels and was expanded to 422 channels features. These two methods were called feature-back and feature-for, respectively.

2.3 The deformable convolutional network system

2.3.1 Traditional convolutional network

With the rapid development of deep learning, CNN has been widely used in the field of classification and recognition, especially in computer vision and natural language processing. CNN is a very basic network structure in deep learning, and many well-known and effective networks are improved based on it. CNN can automatically extract effective features during training to realize the end-to-end training process without separating the two steps of feature extraction and classification training like traditional algorithms.

The three most critical elements of CNN are local receptive fields, weight sharing, and downsampling in the pooling layer. The concept of the local receptive field is inspired by the structure of the visual system in neuroscience. Neurons in the visual cortex receive information locally (that is, these neurons only receive signals from a small area). Each neuron does not need to receive all the pixel information of the image, but only needs to perceive the local area, and then we integrate the local information received by these neurons at a higher layer to obtain the global information. Weight sharing can be seen as a feature extraction method. In other words, the convolutional layer has multiple convolution kernels, also called filters, and each convolution kernel corresponds to a feature map after filtering. All pixels in the same feature map come from the same convolution kernel. In the pooling layer, researchers conducted a down-sampling to aggregate the features in different areas of the feature maps, for example, the average value or max value of several features in a certain area could be obtained. These statistical features don't only have a much lower dimensionality, but also enhance the robustness of the features. This kind of aggregation operation is called pooling, and it is usually common to use average pooling or maximum pooling.

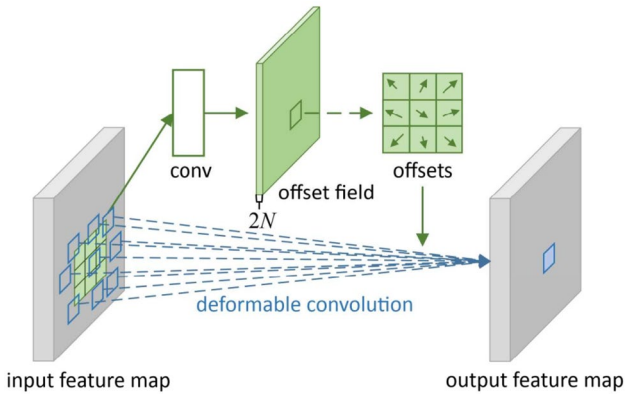


Fig. 3 Illustration of 3 × 3 deformable convolution [27]

2.3.2 Deformable convolutional layer

Deformable convolution is developed based on traditional convolution. The core of DCN lies in offsetting the convolution kernel. The offset acquisition process is shown in Fig. 3. The implementation is to insert an intermediate output layer, and the output results are the convolution kernel offsets.

The traditional 2D convolution is formed from two steps: (1) sampling using a regular grid R over the input feature map x; (2) summation of sampled values weighted by w. The grid R defines the receptive field size. For example, using a 3×3 convolution kernel,

$$R = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\} \tag{1}$$

And the traditional 2D convolution process for each point p_0 on the output feature map y can be expressed as:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n) \tag{2}$$

The DCN convolution process for each point p_0 can be expressed as:

$$y(p_0) = \sum_{p_n \in R} w(p_n) \cdot x(p_0 + p_n + \Delta p_n) \tag{3}$$

where the R represented augmented with offsets $\{\Delta p_n | n = 1,\ldots,N\}$, and $N = |R|$.

Since the introduction of offset may get the point beyond the original image, so bilinear interpolation is used for this as:

$$x(p) = \sum_{q} G(p,q) \cdot x(q) \tag{4}$$

where $x(q)$ represents the point existing in the original image, q enumerates all integral spatial locations in the input feature map x, and $x(p)$ is an arbitrary point obtained by the offsets, namely $p = p_0 + p_n + \Delta p_n$. In addition, the $G(\cdot,\cdot)$ represents the bilinear interpolation kernel, it can be separated into two one dimensional kernels as

$$G(p,q) = g(p_x,q_x) \cdot g(p_y,q_y) \tag{5}$$

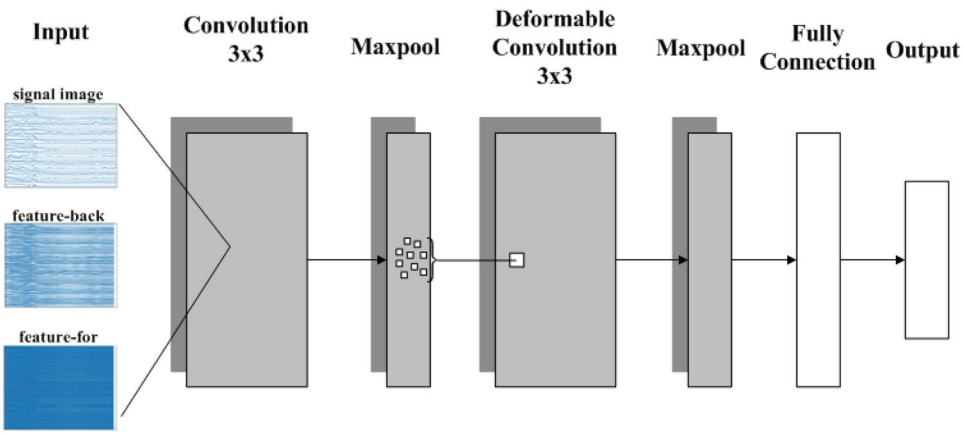
where $g(a,b) = \max(0, 1 - |a-b|)$.

The final offsets can be obtained by back propagation through the above formula. For more information about DCN, see [27].

2.3.3 Network architecture

To make good use of the relationship between the different channels of the input, we first use the traditional convolutional layer to extract low-dimensional feature maps, and then employ deformable convolution to extract high-dimensional abstract feature maps. The traditional 2D convolution is a good feature extraction method, even if the input is a one-dimensional signal, such as the sEMG signals, the effect is not bad. And DCN uses bilinear interpolation to obtain the value of the pixels that do not exist originally, but the data we used is a multi-channel sEMG map, which greatly deviates the bilinear interpolation relationship between

Fig. 4 Network architecture



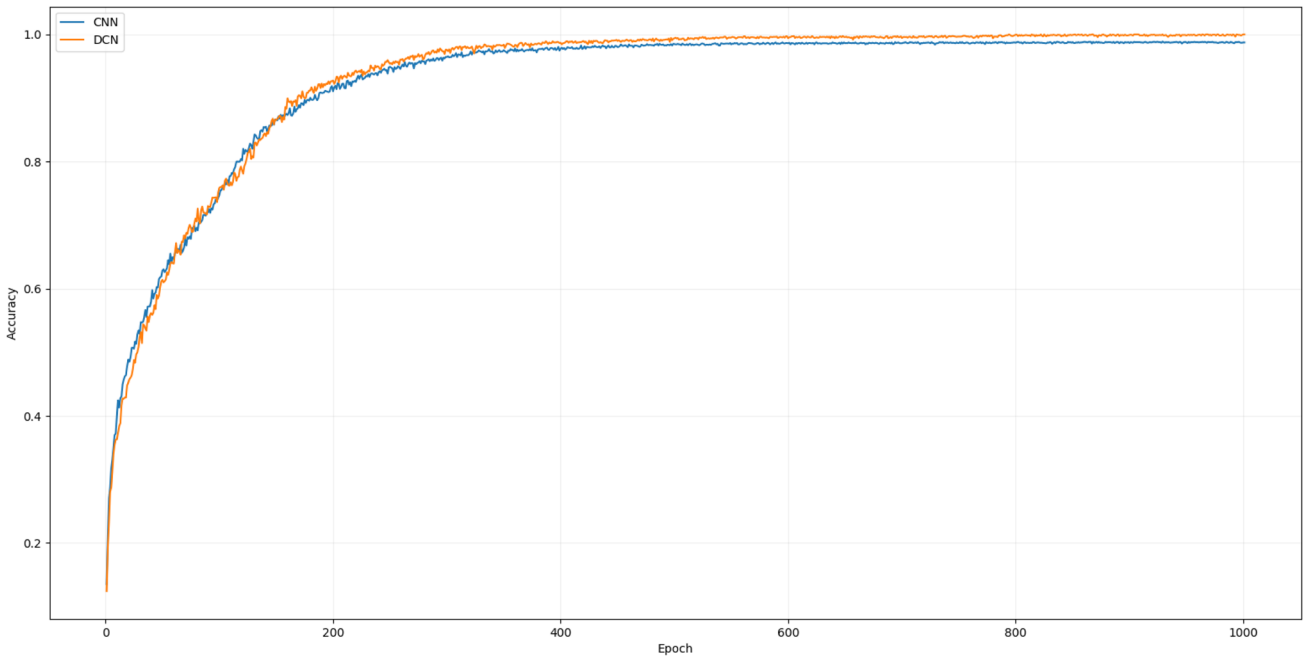


Fig. 5 The accuracy in training

different channels. Therefore, we first used a traditional convolutional layer to extract low-dimensional features of the original sEMG signal data or handcrafted feature image to obtain the low-dimensional feature maps. And then we used deformable convolution to obtain a broader and adaptive receptive field on the low-dimensional feature maps. Finally, we obtained the better high-dimensional feature maps for classification and recognition.

In this paper, we use the deformed CNN–DCN, which is divided into six layers in total, the first layer is a traditional convolution layer, and its convolution kernel size is 3×3, followed by a maximum pooling layer and normalization processing, the third layer is the second convolutional layer, the convolutional layer is deformable convolution layer, and its convolution kernel size is also 3×3, followed by a maximum pooling layer and normalization processing, the fifth layer is a fully connected layer with a dropout rate of 0.5 [29], and the rest is the output layer with a softmax activation function. In addition, the ReLU function was used in the convolution layer and pooling layer. The network architecture is briefly demonstrated in Fig. 4.

3 Results and discussion

In this experiment, we used the traditional CNN and the DCN for comparison, both are based on a deep learning framework: MXNET [30]. Firstly, we took the sEMG data (200×42) directly as the images as the network input, and use the stochastic gradient descent (SGD) method during training [31]. The training epoch was set to 1000. The initialization of the parameters was achieved by a fixed random seed, and the learning rate is set to 0.03. Besides, we set the batch size to 100 due to the performance of memory. And the network is built in the framework of MXNET [30]. In addition, the experimental environment of CPU core is i5-4200H, and the memory size is 8 GB.

We chose 70% of the data as the training set and 30% of the data as the test set. And the data was divided into three groups, the first group contains 12 finger movements; the

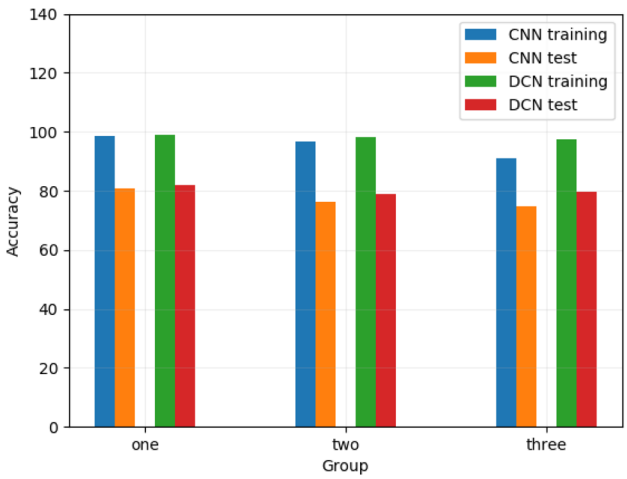


Fig. 6 The classification accuracy for different groups

Table 2 The classification accuracy for different groups

Group	CNN training (%)	CNN test (%)	DCN training (%)	DCN test (%)
One	98.7391	80.7000	99.0000	81.8000
Two	96.7368	76.2941	98.1316	78.9412
Three	90.9455	74.6250	97.3273	79.5417

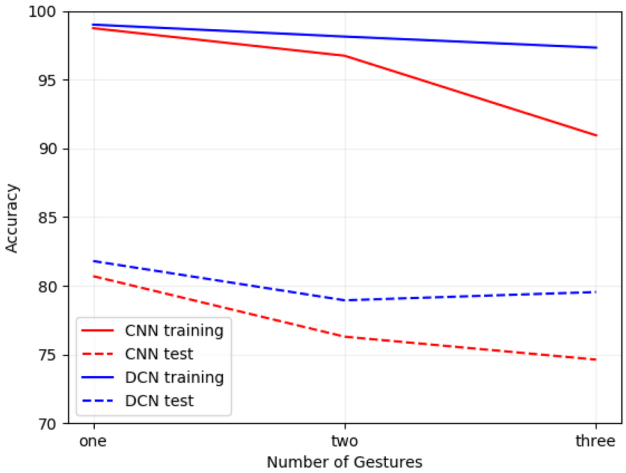


Fig. 7 The accuracy with different number of gestures

second group contains finger movements and hand postures, including 20 gestures; the third group contains all gestures, including 29 gestures.

The accuracy in training is illustrated in Fig. 5. The experimental results are shown in Fig. 6 and Table 2. The confusion matrices of group one are shown in Fig. 8, and other groups' results are demonstrated on section Supplementary Materials. For all groups, the accuracy of DCN is better than CNN in the training set and test set. As the number of gestures increases, the overall trends of the classification accuracy of the two networks are decreasing. But the accuracy of DCN changes more insignificant than CNN as the number of gestures increases. That is, the DCN is more robust, and Fig. 7 shows more intuitive. Note that the accuracy of the DCN test set in Group three is 79.54%, even better than the accuracy of the DCN test set in Group two. This shows that as the number of data increases, DCN can extract more universal features, thereby improving classification accuracy.

To a certain extent, the behavior of offsetting convolution kernel in the DCN can be regarded as an extension of the receptive field. Therefore, in the following experiment, the receptive field is further expanded by the method of dilated convolution. To exam the classification effects of the DCN network under the different dilation values. The dilation value of the dilated convolution can be simply regarded as the distance between two adjacent convolution points in the horizontal or vertical direction. The conventional convolution can be regarded as the dilated convolution with the dilation value of (1,1). The result of transforming from regular convolution to dilated convolution with the dilation value of (2,2) is shown in Fig. 9. The results are shown in Fig. 10.

Within a certain range, the expansion of the receptive field can enhance the network to integrate the information, thereby

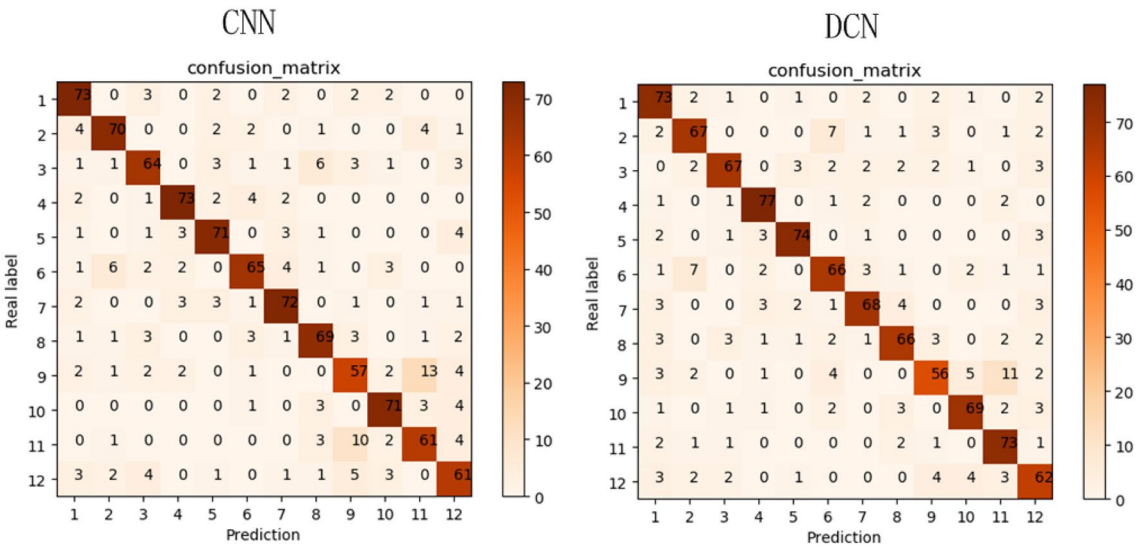


Fig. 8 Confusion matrices of group one with CNN and DCN

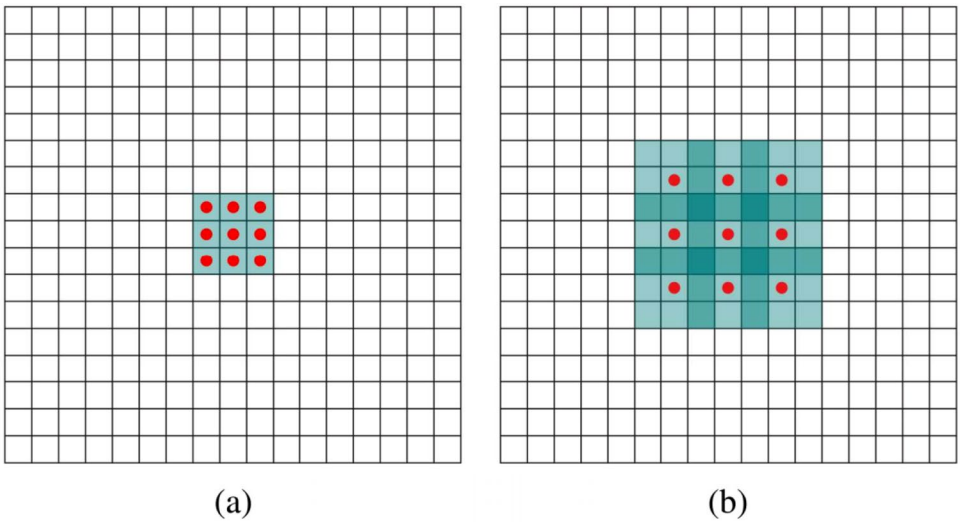


Fig. 9 Illustration of 3 × 3 dilated convolution. **a** Dilation value is (1,1). **b** Dilation value is (2,2) [25]

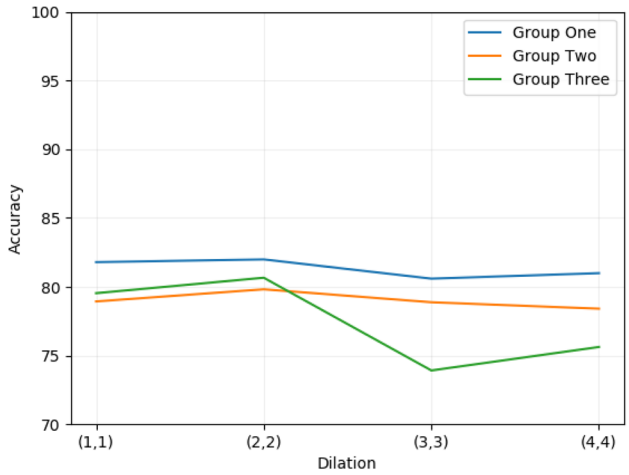


Fig. 10 The classification rate with dilation convolution for different groups

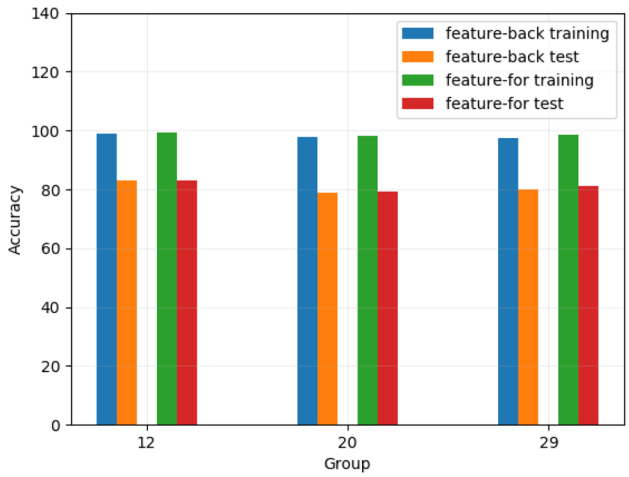


Fig. 11 The classification rate with represent methods for different groups

Table 3 The classification accuracy with represent methods for different groups

Group	Feature-back Training (%)	Feature-back Test (%)	Feature-for Training (%)	Feature-for Test (%)
One	98.7826	83.0000	99.3478	83.1000
Two	97.9211	79.0000	98.2368	79.2941
Three	97.5263	79.9412	98.4474	81.2353

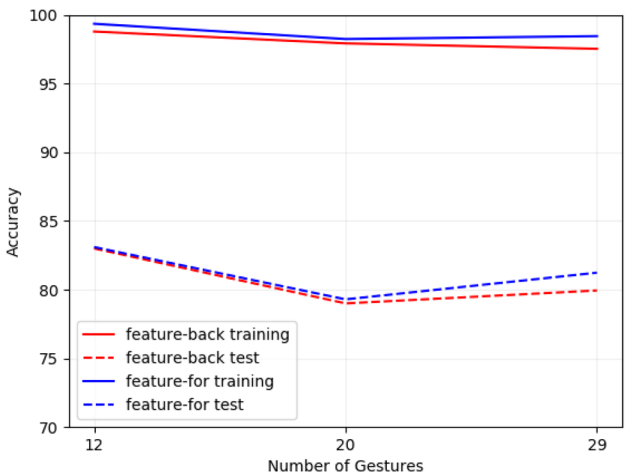


Fig. 12 The accuracy of represent methods with different number of gestures

enhancing the feature extraction capabilities. But when the receptive field is too large, it will lead to the omission of information, which has a great impact on the accuracy of classification and recognition. The expansion of the receptive field by deformable convolution is learned independently through the

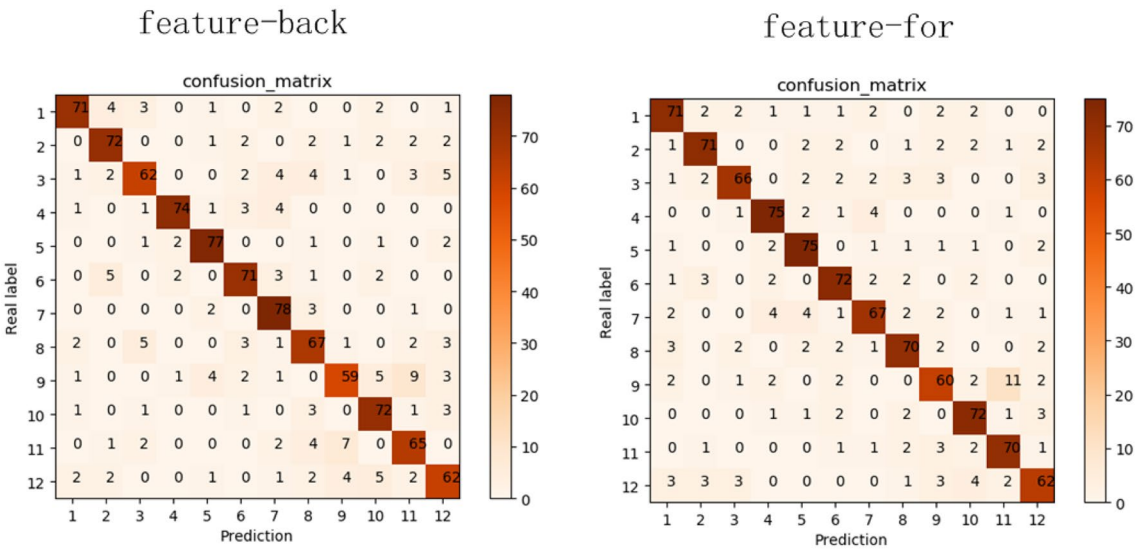


Fig. 13 Confusion matrices of group one with feature-back and feature-for

learning ability of the network in a small range, so the classification accuracy can be improved.

Thirdly, we took the feature-back data and feature-for data as the input to the DCN. Other settings are the same as above mentioned experiment. The experimental results are shown in Fig. 11 and Table 3. The confusion matrices of group one are shown in Fig. 13. The feature-back method only used three traditional feature extraction methods, so the input data of feature-back (39×126) is even smaller than the original signal (200×42). And the input data of feature-for (39×422) is larger than the original signal and the representation of the feature-back. In all test sets, the results of both representation methods are better than the DCN. The average accuracy of the feature-for representation method is better than that of the feature-back representation method. According to the number of gestures from small to large, feature-for represent 0.10%, 0.29%, 1.29% higher average accuracy than feature-back respectively. The results are shown in Fig. 12. One possible explanation is that the comparison of the number of different gesture experiments used in this paper is based on the gesture category. That is, the number of gestures is increased by adding the same kind of gestures, while the different kinds of gestures have a more complicated spatio-temporal relationship. So the feature-for method enables the convolution kernel to extract the implicit relevance between different features in different areas, as well as increases the classification accuracy.

4 Conclusions

In this paper, we used DCN to extract features and conducted classification. And the performance of DCN is better than traditional CNN in the different numbers of gestures. Furthermore, we tested the influence of the receptive field in the neural network on the feature extraction of the sEMG signal by experimenting with DCN under different dilation values. Besides, we compare the performances of two feature representation methods, which we called feature-back and feature-for. The result of feature-for is a little better than the feature-back in the experiment.

In the future, we can change the interpolation relationship of DCN to make it more in line with the characteristics of the multi-channel sEMG signals, thereby extracting better features to improve the classification accuracy. Also, we can adjust deformable convolution to make it more suitable for the sEMG signal. In this paper, the feature-back method only uses three traditional feature extraction methods, so the input data of feature-back (39×126) is even smaller than the original signal (200×42). Maybe we can use a more traditional feature extraction method to improve the results of the feature-back representation method. And for both feature-back and feature-for methods, we can take some ways to mitigate overfitting.

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s13042-021-01482-7.

References

1. Ahsan MR, Ibrahimy MI, Khalifa OO (2012) Optimization of neural network for efficient EMG signal classification. In: International symposium on mechatronics and its applications
2. Li G (2011) Electromyography pattern-recognition-based control of powered multifunctional upper-limb prostheses. Adv Appl Electromyogr 6:99–116

3. Wang R, Huang C, Li B (1996) Discussion on various methods of EMG processing for the control of prostheses. In: Proceedings of international conference on biomedical engineering, pp 341–344
4. Hu OH (2007) Myoelectric control systems-a survey. Biomed Signal Process Control 2:275–294
5. Xing K, Yang P, Huang J, Wang Y, Zhu Q (2014) A real-time EMG pattern recognition method for virtual myoelectric hand control. Neurocomputing 136(JUL.20):345–355
6. Faust O, Hagiwara Y, Hong TJ, Lih OS, Acharya UR (2018) Deep learning for healthcare applications based on physiological signals: a review. Comput Methods Progr Biomed 161:1–13
7. Rao R, Derakhshani R (2005) A comparison of EEG preprocessing methods using time delay neural networks. In: International IEEE Embs conference on neural engineering
8. Ron Kohavi A, George B, John H (1997) Wrappers for feature subset selection. Artif Intell 97(1–2):273–324
9. Lam HK, Ekong U, Xiao B, Ouyang G, Liu H, Chan KY, Ho Ling S (2015) Variable weight neural networks and their applications on material surface and epilepsy seizure phase classifications. Neurocomputing 149:1177–1187
10. Ekong U, Lam HK, Xiao B, Ouyang G, Hongbin L (2016) Classification of epilepsy seizure phase using interval type-2 fuzzy support vector machines. Neurocomputing 199:66–76
11. Alty SR, Lam HK, Prada J (2012) On the applications of heart disease risk classification and hand-written character recognition using support vector machines. In: Computational intelligence and its applications: evolutionary computation, Fuzzy Logic, Neural Network and Support Vector Machine Techniques, pp 213–253
12. Darak BS, Hambarde SM (2015) A review of techniques for extraction of cardiac artifacts in surface EMG signals and results for simulation of ECG–EMG mixture signal. In: 2015 International conference on pervasive computing (ICPC)
13. Phinyomark A, Quaine F, Charbonnier S, Serviere C, Tarpin-Bernard F, Laurillau Y (2013) EMG feature evaluation for improving myoelectric pattern recognition robustness. Expert Syst Appl 40:4832–4840
14. Li Y, Zhang Q, Zeng N, Chen J, Zhang Q (2019) Discrete hand motion intention decoding based on transient myoelectric signals. IEEE Access 7:81630–81639
15. Tsinganos P, Cornelis B, Cornelis J, Jansen B, Skodras A (2018) Deep learning in EMG-based gesture recognition. In: 5th International conference on physiological computing systems
16. Atzori M, Gijsberts A, Castellini C, Caputo B, Müller H (2014) Electromyography data for non-invasive naturally-controlled robotic hand prostheses. Sci Data 1:1–13
17. Pinzón-Arenas JO, Jiménez-Moreno R, Herrera-Benavides JE (2019) Convolutional neural network for hand gesture recognition using 8 different EMG signals. In: Symposium on image, signal processing and artificial vision
18. Jia G, Lam HK, Liao J, Wang R (2020) Classification of electromyographic hand gesture signals using machine learning techniques. Neurocomputing 401:236–248
19. Yang W, Yang D, Liu Y, Liu H (2019) Decoding simultaneous multi-DOF wrist movements from raw EMG signals using a convolutional neural network. IEEE Trans Hum Mach Syst 49(5):1–10
20. Chen X, Li Y, Hu R, Zhang X (2020) Hand gesture recognition based on surface electromyography using convolutional neural network with transfer learning method. IEEE J Biomed Health Inform 25(4):1
21. Yang W, Yang D, Liu Y, Liu H (2019) Emg pattern recognition using convolutional neural network with different scale signal-spectra input. Int J Hum Robot 16(04):305–345
22. Yongkang Wong, Wentao Wei, Mohan Kankanhalli, Weidong Geng (2018) A novel attention-based hybrid CNN–RNN architecture for SEMG-based gesture recognition. PloS One 13:e0206049
23. Geng W, Du Y, Jin W, Wei W, Li J (2016) Gesture recognition by instantaneous surface EMG images. Sci Rep 6:36571
24. Jiang W, Yin Z (2015) Human activity recognition using wearable sensors by deep convolutional neural networks. In: ACM international conference on multimedia, pp 1307–1310
25. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122
26. Everingham M, Gool LV, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vis 88(2):303–338
27. Dai J, Qi H, Xiong Y, Li Y, Zhang G, Hu H, Wei Y (2017) Deformable convolutional networks. In: Proceedings of the IEEE international conference on computer vision, pp 764–773
28. Atzori M, Gijsberts A, Heynen S, Mittaz-Hager AG, Mueller H (2012) Building the ninapro database: a resource for the biorobotics community. In: 2012 4th IEEE RAS and EMBS international conference on biomedical robotics and biomechatronics (BioRob)
29. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R (2014) Dropout: a simple way to prevent neural networks from overfitting. J Mach Learn Res 15(1):1929–1958
30. Chen T, Li M, Li Y, Lin M, Wang N, Wang M, Xiao T, Xu B, Zhang C, Zhang Z (2015) Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. arXiv preprint arXiv:1512.01274
31. Sutskever I, Martens J, Dahl G, Hinton G (2013) On the importance of initialization and momentum in deep learning. In: International conference on machine learning, PMLR, pp 1139–1147