IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 16, 2022, accepted March 4, 2022, date of publication March 10, 2022, date of current version March 30, 2022.

*Digital Object Identifier 10.1109/ACCESS.2022.3158667*

# Research on sEMG-Based Gesture Recognition by Dual-View Deep Learning

**YAN ZHANG[1,2], FAN YANG[1,2], QI FAN[1], ANJIE YANG[1], AND XUAN LI[1]**

[1] School of Artificial Intelligence, Hebei University of Technology, Tianjin 300401, China
[2] Engineering Research Center of Intelligent Rehabilitation, Ministry of Education, Hebei University of Technology, Tianjin 300401, China

Corresponding author: Fan Yang (sdezbzr@163.com)

This work was supported in part by the National Science Foundation of China under Grant 61773151 and Grant 61703135, and in part by the Natural Science Foundation of Hebei Province under Grant F2018202279.

**ABSTRACT** In the field of human-machine interaction, gesture recognition using sparse multichannel surface electromyography (sEMG) remains a challenge. Based on the Hilbert filling curve, a dual-view multi-scale convolutional neural network (DVMSCNN) is designed to enhance gesture recognition performance in this paper. The network consists of two parts. In the first part, sEMG is filled using Hilbert filling curve, and the obtained images in the time and electrode domain are used as inputs to the block. In the second part, the depth features learned by block are fused and classified by a ‘‘layer fusion’’ based view aggregation network. The evaluation of the architecture in the four databases of Ninapro-DB1, DB2, DB3 and DB4 shows that DVMSCNN is more than 7% more accurate than other state-of-the-art methods. When validated using a home-grown dataset, DVMSCNN was able to achieve a recognition rate of 0.8848.

**INDEX TERMS** Human–machine interaction, gesture recognition, multi-view learning, Hilbert filling curve, convolutional neural network.

## I. INTRODUCTION

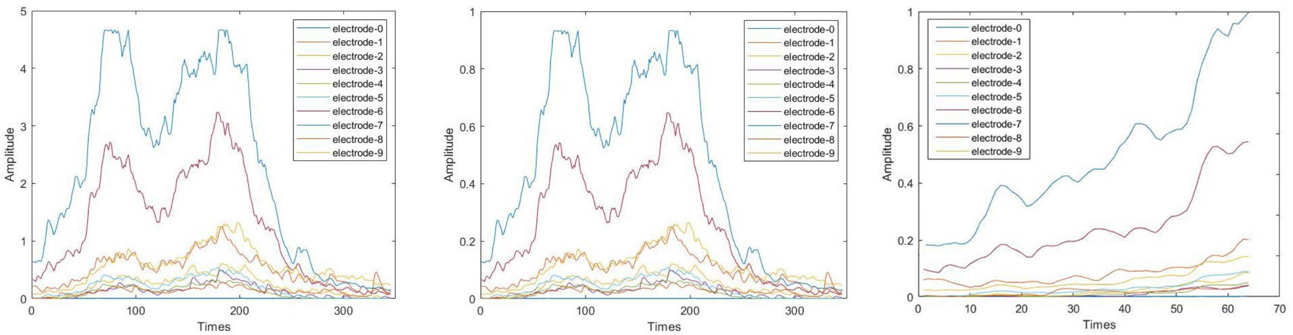
As human-machine interaction playing an important role in modern life, the question of how to interact with computers in an efficient and natural way has become an important research topic. Hand gestures, which are simple and natural, are essential parts of body language. Thus, gesture recognition is also a key technology in human machine interaction [1].

Gesture information collection relies on external sensors and wearable sensors. The former mainly includes conventional cameras [2]–[4], Kinect [5]–[7], and radar [8]–[10], etc. The latter mainly includes inertial measurement units [11]–[13] and sEMG sensors [14]–[16].

In the case of classification, the traditional machine learning (ML) has been widely used for gesture recognition [17], [18]. Lu *et al.* [19] adopted Bayesian linear classifier and an improved dynamic time warping algorithm for classification recognition of 19 gestures. Results suggest that the average accuracy of 89.6% in user-independent testing. Besides, by comparing different sEMG features and classifiers for the classification of 52 gestures from the Ninapro reference dataset. A random forest classifier, a combination of statistical, and frequency domain features, i.e., MAV, histogram, wavelet, and Fourier transform features, yield the best performance [20]–[24]. With the continuous development of deep learning (DL), it has been gradually applied to gesture recognition in recent years [25]–[28]. Panwar *et al.* [29] presents a deep learning framework, Rehab-Net, which can classify the three movements from stroke survivors without using any feature engineering. Finally, the overall accuracy of Rehab-Net achieves 88.87%. Tsagkas *et al.* [31] used short latency dimension reduced sEMG spectrograms as input to convolutional neural network (CNN) and support vector machine (SVM). The results showed that CNN consistently exhibited better performance.

CNN have made breakthrough in feature extraction and image classification tasks in 2D problems. Thus, it makes sense to find a suitable method to convert sEMG into an image that can be used as CNN input [32], [33]. Hilbert filling curves can be applied to organize or compress data by providing a mapping between D-dimensional spaces while retaining

The associate editor coordinating the review of this manuscript and approving it for publication was Long Wang.



(a) (b) (c)

**FIGURE 1.** The sEMG signal before and after data preprocessing of NinaPro-DB1, where (a) is the raw sEMG, (b) is the filtered and normalized sEMG, and (c) is the sEMG after the sliding window method.

locality, such as biomedical signals [34]. Chen *et al.* [35] proposed a feature extraction method based on Hilbert-Huang transform and used extreme learning machine for classification. The experimental results showed that the classification accuracy of this method was 88%. Kurek *et al.* [36] used Hilbert curves to represent mammograms as 1-dimensional vectors and extracted features from them to detect breast cancer, with a final accuracy of 85.83%. In this case, for the input image problem of CNN, filling sEMG with Hilbert curve helps to enhance the classification effect of gesture recognition.

In this paper, we design a dual-view multi-scale convolutional neural network (DVMSCNN) to improve sEMG-based gesture recognition performance. The network consists of two parallel multi-scale CNN and a view aggregation network based on "layer fusion". The two views' inputs are 2D time domain and electrode domain images obtained by filling the sEMG with Hilbert.

The rest of this paper is arranged as follows. Section 2 introduces the process of data collection and completion. Section 3 proposes the classification model based on DVMSCNN. Section 4 presents the experiment results of proposed algorithm. Section 5 summarizes the study and puts forward future work.

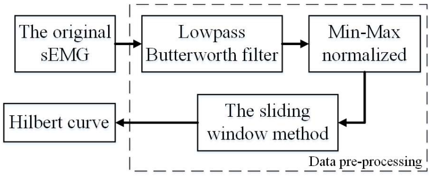


**FIGURE 2.** Flowchart of data processing.

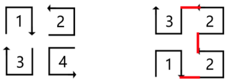


**FIGURE 3.** Hilbert curves of orders 1 and 2.

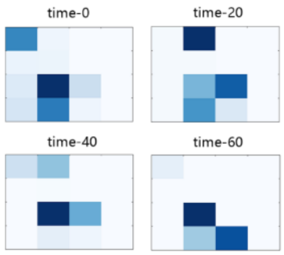


**FIGURE 4.** Image representation of the Hilbert electrode dimension at time (0, 20, 40, 60).

## II. DATASET AND DATA PROCESSING

### A. DATASET AND PREPROCESSING

The evaluations in this work were performed offline using multi-channel sEMG signals from the publicity available NinaPro databases [42]. We chose 4 sub-databases of NinaPro, which the details are as follows:

The first sub-database (denoted as NinaPro-DB1) contains 10-channels sparse multi-channel sEMG signals recorded from 27 intact subjects. Each gesture was recorded with 10 trials at a sampling rate of 100Hz. Each subject was asked to perform 53 gestures, including 12 finger movements, 17 wrist movements and hand postures, 23 grasping and functional movement. Relaxation state between each repetition was resting gesture.

The second sub-database (denoted as NinaPro-DB2) contains 12-channels sparse multichannel sEMG signals recorded from 40 intact subjects. Each gesture was recorded with 6 trials at a sampling rate of 2000Hz. Each subject was asked to perform 50 gestures, including 9 force patterns, 17 wrist movements and hand postures, 23 grasping and functional movement, and the rest movement.

The third sub-database (denoted as NinaPro-DB3) 12-channels sparse multichannel sEMG signals recorded from 11 transradial amputees; other information exactly the same as those in NinaPro-DB2. According to the authors of NinaPro database, three amputated subjects performed only a part of gestures due to fatigue or pain, and in two amputated subjects, the number of electrodes was reduced to ten due

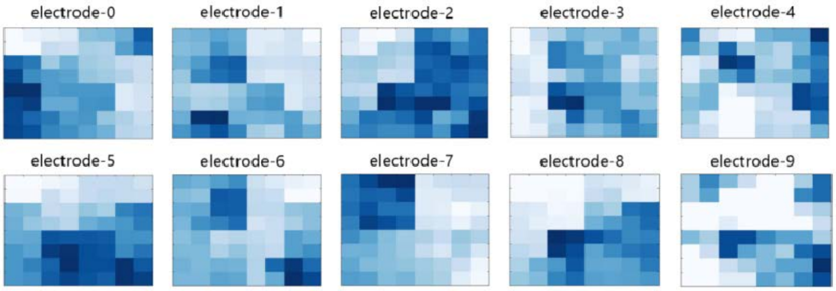


**FIGURE 5.** Image representation of the Hilbert time dimension on the electrode.

to insufficient space. To ensure training and testing of the model can be completed, we omitted data from these subjects following the experimental configuration used by [37].

The fourth sub-database (denoted as NinaPro-DB4) contains 12-channels sparse multichannel sEMG signals recorded from 10 intact subjects. Each gesture was recorded with 6 trials at a sampling rate of 2000Hz. Each subject was asked to perform exactly the same 53 gestures as those in NinaPro-DB1. Because two subjects (i.e., subject 4 and subject 6) did not complete all hand movements, their data was omitted in our experiment.

### B. DATA PROCESSING

Due to memory limitation of the hardware, for experiments on NinaPro DB2-DB4, we down-sampled the sEMG signals from 2000 Hz to 100 Hz following the experimental configuration used in [38]. The data processing is divided into two parts, as shown in Fig. 2. The first part is data pre-processing. Firstly, the sEMG signals use a 1st order 1 Hz lowpass Butterworth filter. Then, the data are Min-Max normalized. As the last step, the data is segmented into overlapping windows of length 640ms with a step of 10ms, using the sliding window method. Fig.1 shows the sEMG signal before and after data preprocessing of NinaPro-DB1.

The second part is to fill the preprocessed data with Hilbert, which is a continuous fractal space-filling curve. Space-filling curves have been widely applied to tasks in data organization and compression. The Hilbert curve is known for being superior in preserving locality compared to alternatives [39], [40], such as the z-order and Peano curves. The rule is to rearrange the D-dimensional space in a recursive manner for another dimension while keeping one dimension of the sequence data unchanged, where D = 2. The sEMG signal is transformed in two ways: (1) across the time dimension, i.e., for each sEMG channel, map the time series into a 2D image, or (2) across the sEMG channels, i.e., for h each time instant, map the values of the channels into a 2D image. If denotes a Hilbert curve of order i, the specific conversion is as follows:

1) $H_0$ is a single point. Second item;
2) $H_1$ consists of four copies of (the point) $H_0$, connected with three straight segments of length h at right angles to each other. Four orientations of this curve, labeled 1, 2, 3, and 4, are shown in Fig. 3.
3) $H_2$ is constructed by connecting four copies of different orientations $H_1$ with three straight segments of length $h/2$. There are four possible directions, and the rules of construction are summarized in Table 1. Fig. 3 shows a 2nd order Hilbert curve, which is oriented #2, i.e., $H_1$ consists of the 1223 direction.
4) $H_n$ is constructed by connecting four copies of different orientations $H_{n-1}$ with three straight segments of length $h/n$. Therefore, Hilbert curves can be generated according to this recursive approach for higher order curves.

**TABLE 1.** Four possible directions for $H_2$.

| Serial No. | Construction rules | | | | | | |
|---|---|---|---|---|---|---|---|
| 1 | 2 | ↑ | 1 | → | 1 | ↓ | 4 |
| 2 | 1 | → | 2 | ↑ | 2 | ← | 3 |
| 3 | 4 | ↓ | 3 | ← | 3 | ↑ | 2 |
| 4 | 3 | ← | 4 | ↓ | 4 | → | 1 |

For a given M × N sEMG signal, M represents the time series and N represents the electrode channels. When mapping in time dimension, for each electrode n and each time step m, Hilbert generates the image coordinates $(i, j)$ $(i = j, m = i \times j)$ for time step m so that the image value at position $(i, j, n)$ is equal to the signal value of electrode n at time-step m. Finally, the maximum size $L \times L \times N$ image is achieved, where $M = L^2$ and L is a power of 2. Similarly, when performing filling in the electrode dimension, for each electrode m and each time step n, Hilbert generates the image coordinates $(i, j)$ $(i = j, n = i \times j)$ for electrode n. The image value at position $(m, i, j)$ is equal to the signal value at time-step m for electrode n. The final maximum size $M \times K \times K$ image is achieved, where $N = K^2$ and K is a power of 2. For example, 64 × 10 sEMG are shown in Fig. 1(c), when it is mapped in the electrode dimension, it can eventually become 64 × 4 × 4 image. Fig. 4 shows that the image representation of Hilbert electrode dimension on time (0,20,40,60). When it is transformed in the time dimension, it can eventually become 8 × 8 × 10 image, as shown in Fig. 5 for the image representation of the Hilbert time dimension on the electrode. Note that when using sequence segments of length less than

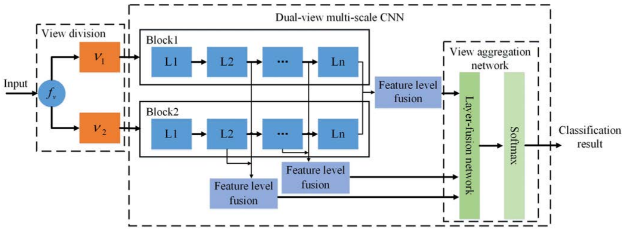


**FIGURE 6.** The dual-view multi-scale convolutional neural network.

or, rows and columns with only zeroes can be deleted(filled) and cropped to the final image.

## III. FRAMEWORK AND METHODS

### A. MULTI-VIEW LEARNING

Multi-view learning is an emerging direction in machine learning which refers to the learning of multi-view data or multiple feature sets that can reflect different attributes or views of the data. Compared with single-view learning, multi-view learning can achieve higher performance by making full use of the information in different views of the data. Multi-view CNN is one of the important issues in the practical application of multi-view learning, and it consists of two parts. The first half is a multi-stream CNN composed of multiple branches. Each branch models the data of each view separately to make full use of each view in the learning process; the second half uses a multi-view aggregation network to perform aggregation on the multi-view features learned in the first half of the network.

The data of N views are assumed to be $v_1, v_2, \ldots, v_n$. The multi-view convolutional neural network is modeled in the first half by N convolutional neural network branches $h_{w1}, h_{w2}, \ldots, h_{wn}$, where $w_1, w_2, \ldots, w_n$ are the parameters of these N convolutional neural network branches:

$$H_i = h_{wi}(v_i) \tag{1}$$

where $H_i$ is the feature of the output of the hidden layer specified for the i-th network branch, which can be understood as the feature obtained from the data $v_i$ of the i-th view learned by the convolutional neural network $h_{wi}$.

Then, the multi-view features are aggregated by a multi-view aggregation network $h_w^{agree}$, and the final gesture recognition label y is obtained:

$$y = h_w^{agree}\left(\{H_i\}_{i=1}^{N}\right) \tag{2}$$

### B. PROPOSED DEEP LEARNING FRAMEWORK

Inspired by multi-view learning, propose the dual-view multi-scale convolutional neural network (DVMSCNN) architecture illustrated in Fig. 6. First, two views $v_i$ $(i = 1, 2)$ of the sEMG signal after Hilbert transformation are expressed as follows:

$$v_i = f_v(x) \tag{3}$$

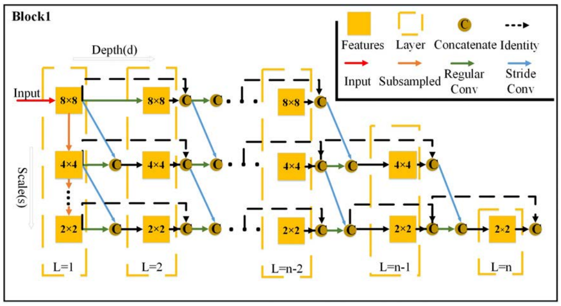


**FIGURE 7.** The specific structure of block1.

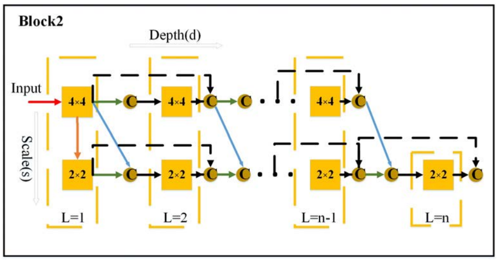


**FIGURE 8.** The specific structure of block2.

where x is denoted as sEMG, $f_v(\cdot)$ denotes Hilbert transform, $v_1$ and $v_2$ are the image representations of the time domain and electrode domain, respectively.

Then, the two views are modeled in parallel by two blocks. This process can be formulated as:

$$H_{\mathrm{i}}^{j} = f(v_i) \tag{4}$$

where $\mathrm{v}_i \in \mathrm{R}^{\mathrm{L}\times\mathrm{W}\times\mathrm{H}}$ is the input, L, W, and H are the dimensions of $v_i$, $f(\cdot)$ denotes the DVMSCNN and $H_{\mathrm{i}}^{j}$ is the output of the jth layer($j = 2, 3, \cdots \mathrm{n}$) of the ith view $(i = 1, 2)$. Finally, the depth features of different layers are fused together by an embedded view aggregation network and fed into the Softmax classifier to output the classification results. This process is written as:

$$H_{layer}^{k} = fuse_F\left(H_i^j\right) \tag{5}$$

$$H_{final} = fuse_S\left(H_{layer}^{k}\right) \tag{6}$$

where $fuse_F(\cdot)$ denotes feature level fusion, $H_{layer}^{k}$, $(k = 2, 3, \cdots n)$ denotes the input to the view aggregation network, $fuse_S(\cdot)$ denotes the view aggregation network, and $H_{final}$ denotes a single output label's final acquisition.

### C. THE BLOCK ARCHITECTURE

Traditional neural networks learn features of fine scale in early layers and coarse scale in later layers (through repeated convolution, pooling, and stride convolution). Coarse scale features in the final layers are important to classify the content of the whole image into a single class. Early layers lack coarse-level features and early-exit classifiers attached to these layers will likely yield unsatisfactory high error rates.

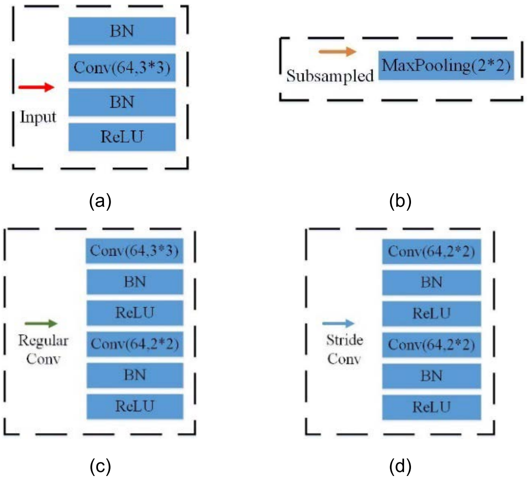


**FIGURE 9.** Detailed steps of input part, down-sampling, regular convolution, and stride convolution. Note: There is zero padding in the regular convolution and the input Conv, but there is no in the second Conv in the stride convolution.

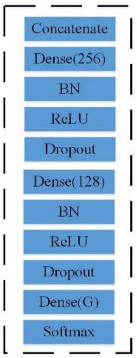


**FIGURE 10.** The view aggregation network. Note: G is the number of categories.

**TABLE 2.** The output of block1 in DVMSCNN as s and L change $x1_L^s$.

| $x1_L^s$ | L=1 | L=2 | $\cdots$ | L=n |
|---|---|---|---|---|
| S=1 | $c_1^1(x1_0^1)$ | $c_2^1(x1_1^1)$ | $\cdots$ | $c_4^1(x1_1^1, x1_2^1, \cdots, x1_n^1)$ |
| S=2 | $k_1^2(x1_1^1)$ | $\begin{bmatrix} k_2^2(x1_1^1) \\ c_2^2(x1_1^2) \end{bmatrix}$ | $\cdots$ | $\begin{bmatrix} k_3^2(x1_1^1, x1_2^1, \cdots, x1_n^1) \\ c_3^2(x1_1^2, x1_2^2, \cdots, x1_n^2) \end{bmatrix}$ |
| S=3 | $k_1^3(x_1^2)$ | $\begin{bmatrix} k_2^3(x1_1^2) \\ c_2^3(x1_1^3) \end{bmatrix}$ | $\cdots$ | $\begin{bmatrix} k_3^3(x1_1^2, x1_2^2, \cdots, x1_n^2) \\ c_3^2(x1_1^3, x1_2^3, \cdots, x1_n^3) \end{bmatrix}$ |

**TABLE 3.** The output of block2 in DVMSCNN as s and L change $x2_L^s$.

| $x1_L^s$ | L=1 | L=2 | $\cdots$ | L=n |
|---|---|---|---|---|
| S=1 | $c_1^1(x2_0^1)$ | $c_2^1(x2_1^1)$ | $\cdots$ | $c_4^1(x2_1^1, x2_2^1, \cdots, x2_n^1)$ |
| S=2 | $k_1^2(x2_1^1)$ | $\begin{bmatrix} k_2^2(x2_1^1) \\ c_2^2(x2_1^2) \end{bmatrix}$ | $\cdots$ | $\begin{bmatrix} k_3^2(x2_1^1, x2_2^1, \cdots, x2_n^1) \\ c_3^2(x2_1^2, x2_2^2, \cdots, x2_n^2) \end{bmatrix}$ |

To address this issue, we propose an architecture similar to the Multi-Scale Dense Network (MSDNet) [41], which are shown in Fig. 7 and Fig. 8, respectively. The horizontal direction corresponds to the layer direction (depth) of the network, it can preserve and progress high-resolution information, which facilitates the construction of high quality coarse features in later layers. The vertical direction represents the scale of the feature map and produce coarse features throughout that are amenable to classification. The n × n in the feature represents the size of the feature map and the top right of block1 shows the meaning of each icon and arrow.

The detailed steps of the Block input part are shown in Fig. 9(a). First, batch normalization is required to process the data before the first convolutional layer consisting of 64 3 × 3 filters to prevent overfitting, after which the batch normalization and ReLU activation function are processed as the original input image $xi_0^1$. When $L = 1$, $xi_0^1$ is down-sampled (2∗2 maximum pooling layer) to obtain a coarser-scale feature map, which determines the scale of the whole block. The detailed steps of down-sampling are shown in Fig. 9(b). If the output feature map of the i-th Block, the L-th layer, and the scale s is denoted as $xi_L^s$, then the output feature map of the first layer is $xi_1^s$. When $L > 1$, the output feature map is the fusion of all previous features of the scale s and s-1 after regular convolution $c_L^s(\cdot)$ or stride convolution $k_L^s(\cdot)$. Fig. 9(c)(d) shows the detailed steps of regular convolution, and stride convolution. Among them, regular convolution increases the depth of the architecture along the horizontal direction (d), and stride convolution changes the scale along the vertical path, transfers information from higher to lower resolutions, and learns a more comprehensive range of depth features.

Since the fusion of regular and stride convolution is a cascade along the channel dimension, its output must have feature maps of the same size. Therefore, there is no zero padding in the second convolutional layer in stride convolution, and all other convolutional layers have zero padding. Besides, batch normalization and ReLU activation functions are applied to each layer to prevent overfitting. Finally, the features of the output of layer L in Block1 and Block2 with scale s are shown in Tables 2 and 3.

## D. VIEW AGGREGATION NETWORK

To aggregate depth features and improve gesture recognition accuracy, DVMSCNN embeds a view aggregation network based on "layer fusion", as shown in Fig. 10. The network fuses the depth features from different layers ($L > 1$), which are shown in Table 4. Then, the fused features are passed through a classifier consisting of an FC layer with 256 hidden units, a 128-hidden unit FC layer, and softmax activation, a single output label is finally obtained. Batch normalization and ReLU nonlinearity function are applied to each layer, while Dropout is applied to each FC layer to prevent overfitting.

**TABLE 4.** The output of DVMSCNN in DVMSCNN as L change $x_L^s$.

| | L=1 | L=2 | • • • | L=n |
|---|---|---|---|---|
| $x_L^s$ | $\begin{bmatrix} k_2^2\left(x1_1^1\right) & k_2^2\left(x2_1^1\right) \\ c_2^2\left(x1_1^2\right) & c_2^2\left(x2_1^2\right) \end{bmatrix}$ | $\begin{bmatrix} k_3^2\left(x1_1^1, x2_2^1\right) & k_3^2\left(x2_1^1, x2_2^1\right) \\ c_3^2\left(x1_1^2, x2_2^2\right) & c_3^2\left(x2_1^2, x2_2^2\right) \end{bmatrix}$ | • • • | $\begin{bmatrix} k_3^2\left(x1_1^1, x1_2^1, \cdots, x1_n^1\right) & k_3^2\left(x2_1^1, x2_2^1,, \cdots, x2_n^1\right) \\ c_3^2\left(x1_1^2, x1_2^2,, \cdots, x1_n^2\right) & c_3^2\left(x2_1^2, x2_2^2,, \cdots, x2_n^2\right) \end{bmatrix}$ |

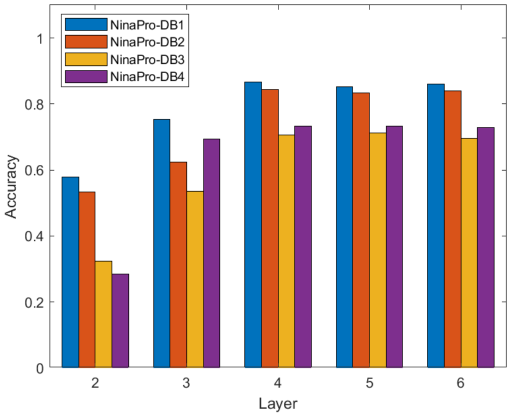


**FIGURE 11.** Performance of DVMSCNN with different number of layers.

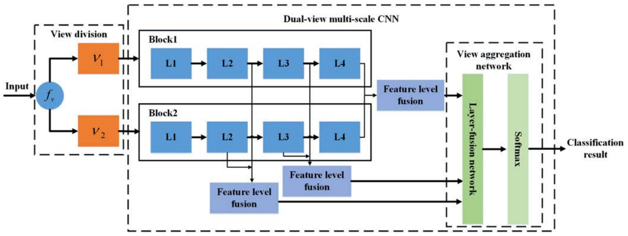


**FIGURE 12.** The 4-layer structured dual-view multi-scale CNN.

**TABLE 5.** Properties evaluation of recognition performance of different view aggregation networks.

| View aggregation network | NinaPro-DB1 | NinaPro-DB2 | NinaPro-DB3 | NinaPro-DB4 |
|---|---|---|---|---|
| $L_2$ , $L_3$ , and $L_4$ | **0.8672** | **0.8329** | **0.7058** | **0.7332** |
| $L_4$ | 0.8293 | 0.7756 | 0.6545 | 0.6824 |
| $L_2$ and $L_4$ | 0.8569 | 0.8963 | 0.6795 | 0.7167 |
| $L_3$ and $L_4$ | 0.8578 | 0.8817 | 0.6813 | 0.7134 |

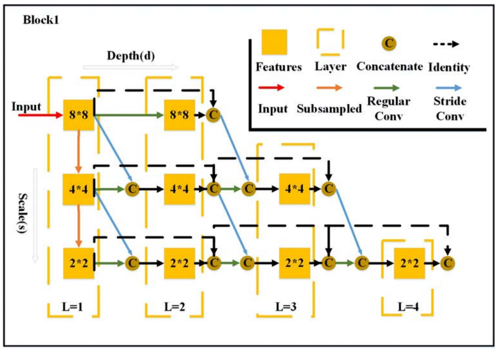


**FIGURE 13.** The specific structure of block1.

**TABLE 6.** Hyper-parameter selection.

| Hyper parameter | Search space | Optimal values |
|---|---|---|
| | | DVMSCNN |
| Pool | {max, average} | Max |
| Dropout | {0.3, 0.25, 0.2} | 0.25 |
| Initial learning rate | {0.05, 0.1, 0.5} | 0.1 |

## IV. RESULTS

### A. PERFORMANCE METRICS

In this paper, the five-fold cross-validations used when experimenting with the NinaPro database. Specifically, for each subject, eight out of ten repetitions are used as training data, and the remaining two are used as testing data. This process is repeated five times and these results are averaged to compute the optimum test performance. Accuracy is used as performance metrics.

### B. EXPERIMENTAL RESULTS OF DVMSCNN

The influence of different network layers ($L = 2, 3, 4, 5, 6$) on gesture recognition are shown in Fig. 11, which show that the performance grows in a stepwise manner as the number of layers increases ($L = 2, 3, 4$). However, when $L > 4$, the number of layers does not bring better performance optimization. Thus, the number of layers of DVMSCNN is set to a four-layer structure as shown in Fig. 12, where block 1 and block 2 are shown in Fig. 13 and Fig. 14.

According to the number of layers of DVMSCNN $L = 4$, four layers fusion can be obtained, which are only $L_4$ fusion, only $L_2$ and $L_4$ for layer fusion, only $L_3$ and $L_4$ for layer fusion and $L_2$, $L_3$, and $L_4$ fusion. The recognition performance of the fusion of different layers is shown in Table 5, where bold indicates the best. The proposed $L_4$ fusion achieved the gesture recognition accuracy of 0.8293, 0.7756, 0.6545 and 0.6824 on NinaPro-DB1, DB2, DB3, and DB4, respectively. However, the accuracy of DVMSCNN obtained after $L_2$, $L_3$, and $L_4$ fusion can be improved to 0.8672, 0.8329, 0.7058 and 0.7332. Therefore, DVMSCNN uses this fusion method of view aggregation network.

### C. HYPER-PARAMETER SELECTION

Hyper-parameter is an important concept in deep learning, referring to the parameter that need to be set artificially before

**TABLE 7.** Hyper-parameter selection.

| Initial learning rate | Dropout | NinaPro-DB1 | | NinaPro-DB2 | | NinaPro-DB3 | | NinaPro-DB4 | |
|---|---|---|---|---|---|---|---|---|---|
| | | max | average | max | average | max | average | max | average |
| 0.05 | 0.3 | 0.7569<br>(0.0246) | 0.7462<br>(0.0378) | 0.7486<br>(0.0314) | 0.7324<br>(0.0314) | 0.6879<br>(0.0312) | 0.6756<br>(0.0362) | 0.7123<br>(0.0198) | 0.7214<br>(0.0248) |
| | 0.25 | 0.7823<br>(0.0548) | 0.7689<br>(0.0649) | 0.7746<br>(0.0348) | 0.7564<br>(0.0387) | 0.6745<br>(0.0379) | 0.6654<br>(0.0243) | 0.7169<br>(0.0255) | 0.7236<br>(0.0159) |
| | 0.2 | 0.7656<br>(0.0387) | 0.7523<br>(0.04365) | 0.7654<br>(0.0434) | 0.7587<br>(0.0298) | 0.6458<br>(0.0234) | 0.6536<br>(0.0302) | 0.7032<br>(0.0343) | 0.6872<br>(0.0148) |
| 0.1 | 0.3 | 0.8026<br>(0.0258) | 0.7754<br>(0.0412) | 0.7826<br>(0.0236) | 0.7659<br>(0.0314) | 0.6632<br>(0.0164) | 0.6456<br>(0.0123) | 0.7268<br>(0.0281) | 0.7231<br>(0.0300) |
| | 0.25 | **0.8672**<br>**(0.0316)** | 0.8283<br>(0.0356) | **0.8329**<br>**(0.0289)** | 0.8139<br>(0.0301) | **0.7058**<br>**(0.145)** | 0.7865<br>(0.0327) | **0.7332**<br>**(0.0367)** | 0.7135<br>(0.0214) |
| | 0.2 | 0.8312<br>(0.0387) | 0.7856<br>(0.0478) | 0.7897<br>(0.0378) | 0.7765<br>(0.0324) | 0.6875<br>(0.0202) | 0.6789<br>(0.0413) | 0.7191<br>(0.0247) | 0.7069<br>(0.0254) |
| 0.25 | 0.3 | 0.7726<br>(0.0347) | 0.7624<br>(0.0247) | 0.7612<br>(0.0364) | 0.7642<br>(0.0378) | 0.6743<br>(0.0362) | 0.6459<br>(0.0385) | 0.6987<br>(0.0315) | 0.6987<br>(0.0316) |
| | 0.25 | 0.7824<br>(0.0312) | 0.7646<br>(0.0345) | 0.8034<br>(0.0211) | 0.7853<br>(0.0287) | 0.6952<br>(0.0347) | 0.6875<br>(0.0298) | 0.7236<br>(0.0316) | 0.7234<br>(0.0317) |
| | 0.2 | 0.7554<br>(0.0147) | 0.7478<br>(0.0479) | 0.7625<br>(0.0425) | 0.7564<br>(0.0245) | 0.6823<br>(0.0316) | 0.6475<br>(0.0341) | 0.7031<br>(0.0432) | 0.6869<br>(0.0198) |

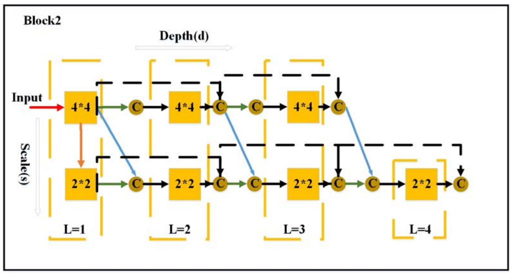


**FIGURE 14.** The specific structure of block2.

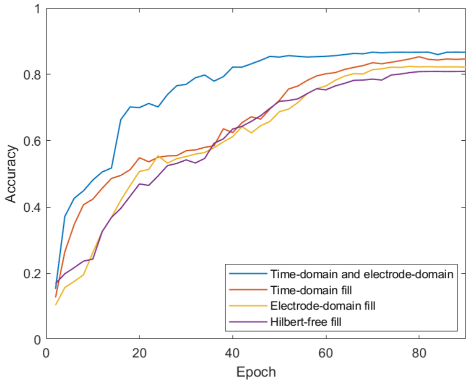


**FIGURE 15.** The accuracy of DVMSCNN on the NinaPro-DB1.

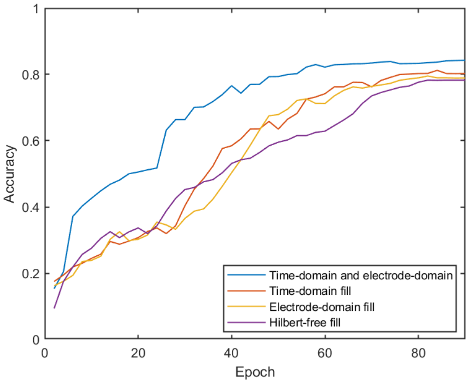


**FIGURE 16.** The accuracy of DVMSCNN on the NinaPro-DB2.

starting training on a model. The setting of hyper-parameter has a great impact on the performance of a network model, so finding a suitable set of hyper-parameters is one of the key steps in building a deep model. For the problem of choosing hyper-parameters for Pool, Dropout, and Initial learning rate, Table 6 lists the search space and the selected values. DVMSCNN trained the network for 90 periods using stochastic gradient descent (SGD) to validate the accuracy of different parameter combinations for NinaPro-DB1, NinaPro-DB2, NinaPro-DB3 and NinaPro-DB4. The results are shown in Table 7, where bold indicates the best, and standard deviation are in parentheses.

In summary, the network was trained using SGD for 90 epochs with an initial learning rate of 0.1, halved every 10 epochs, and a batch size of 1024. Dropout layers were appended after convolutional layers with a forget rate of 0.25 to avoid overfitting the networks caused by the small training set. Besides, weight decay regularization with a value of $l_2 = 0.0005$ was applied to all convolutional layers.

### *D. EFFECT OF HILBERT ON EXPERIMENTAL RESULTS*

This section mainly verifies Hilbert can help to enhance gesture recognition performance or not. DVMSCNN inputs sEMG without Hilbert, sEMG only with time-domain filling, and sEMG with only electrode-domain mapping as the comparison. According to Fig. 15-18, it can be concluded that Hilbert filling can significantly improve the performance of gesture recognition. In addition, the combination of time domain and electrode domain is better than the combination of single domain.

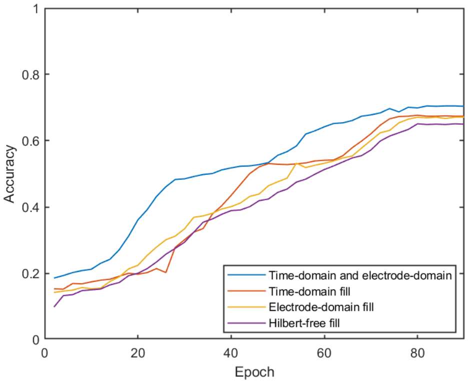


**FIGURE 17.** The accuracy of DVMSCNN on the NinaPro-DB3.

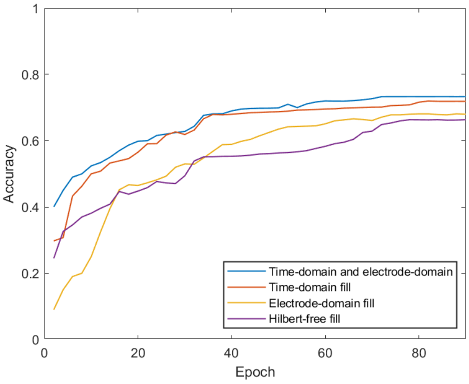


**FIGURE 18.** The accuracy of DVMSCNN on the NinaPro-DB4.

**TABLE 8.** Comparison results with the state-of-the-art gesture recognition approaches.

| Method | NinaPro-DB1 | NinaPro-DB2 | NinaPro-DB3 | NinaPro-DB4 |
|---|---|---|---|---|
| DVMSCNN | **0.8672** | **0.8329** | **0.7058** | **0.7332** |
| VGGNet [41] | 0.8112 | 0.8242 | 0.6842 | 0.7054 |
| MSCNet [42] | 0.8324 | 0.8294 | 0.6915 | 0.7171 |
| MyoCNN [43] | 0.7825 | 0.7915 | 0.6724 | 0.6812 |
| SVM [44] | 0.6945 | 0.6549 | 0.6042 | 0.6182 |
| Random forests | 0.7536 | 0.7325 | 0.6453 | 0.6452 |

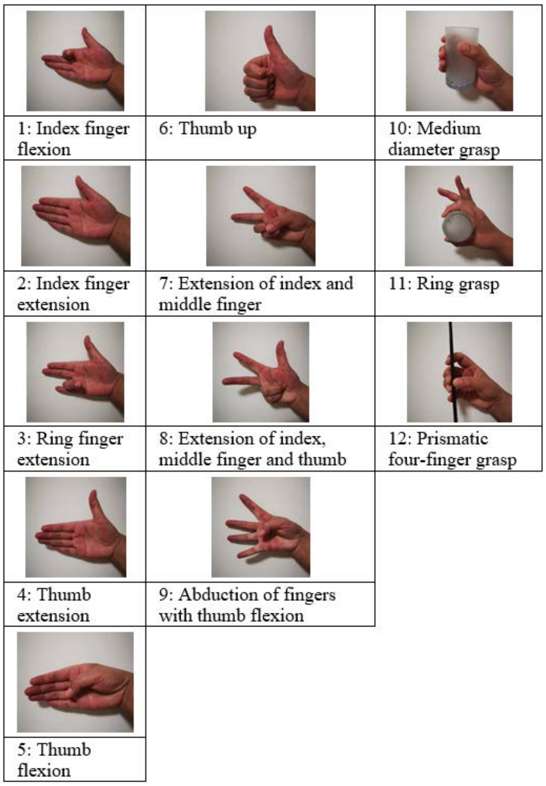

1: Index finger flexion
6: Thumb up
10: Medium diameter grasp
2: Index finger extension
7: Extension of index and middle finger
11: Ring grasp
3: Ring finger extension
8: Extension of index, middle finger and thumb
12: Prismatic four-finger grasp
4: Thumb extension
9: Abduction of fingers with thumb flexion
5: Thumb flexion

**FIGURE 19.** The specific gesture movements.

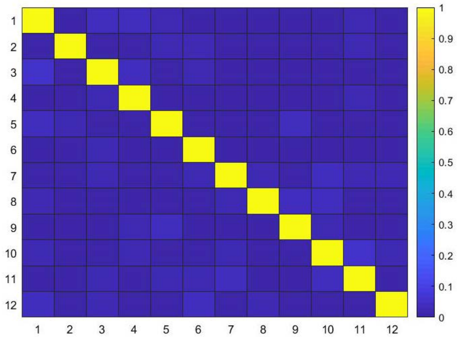


**FIGURE 20.** The confusion matrix for the actual gesture data.

### E. COMPARISON WITH THE STATE-OF-THE-ART GESTURE RECOGNITION APPROACHES

To evaluate the performance of the DVMSCNN, comparative study is conducted with other state-of-the-art sEMG-based models. The introduction and processing are described in section II, and the performance metrics and hyper parameter selection are described from sections IV.A to IV.B. Notably, the training was performed according to the original paper's process since there is a manual extraction of features in the data processing using Random forests and SVM. Table 8 shows that the performance metrics of DVMSCNN implemented on the Ninapro database are 0.8672, 0.8329, 0.7058 and 0.7332, respectively, all higher than the values of other methods.

### F. THE COMPARISON RESULTS BETWEEN THE ACTUAL GESTURE AND THE STANDARD DATA SET

This section adds the results of comparing the actual gestures with the standard dataset. A total of 8 channels of sparse multichannel sEMG signals recorded from 8 healthy subjects (3 females, 5 males) were acquired using the Delsys Trigno wireless acquisition system in this paper. Each gesture was recorded for 5 experiments at a sampling rate of 200 Hz with a 10 second interruption in between each experiment to avoid muscle fatigue. Each subject was asked to perform 12 gestures, including 5 basic finger movements, 4 isotonic and isometric hand configurations, and 3 grasping hand-gestures. The specific gesture movements are shown in Fig. 19.

The actual gesture dataset uses the same data processing as the Ninapro database. The table 9 shows the final accuracy results and the Fig. 20 shows the confusion matrix for the actual gesture data, where the numbers 1-12 represent the gestures in Fig. 19, respectively. Compared to the standard database, the home-grown dataset in this paper is deficient in terms of the number of subjects and gesture actions, but in general, it verifies that DVMSCNN has good recognition performance for the actual gesture map dataset as well.

**TABLE 9. The comparison results between the actual gesture and the standard data set.**

| NinaPro-DB1 | NinaPro-DB2 | NinaPro-DB3 | NinaPro-DB4 | Home-grown dataset |
|---|---|---|---|---|
| 0.8672 | 0.8329 | 0.7058 | 0.7332 | 0.8848 |

## V. CONCLUSION

This paper designs a dual-view multi-scale Hilbert convolutional neural network for effectively classifying hand gesture from Ninapro-DB1, Ninapro-DB2, Ninapro-DB3 and Ninapro-DB4. Firstly, the sEMG is partitioned into the time-domain and electrode-domain based datasets using the Hilbert filling curve's property. Secondly, to improve the classification effect, two views are used as the input to the network, and a view aggregation network based on "layer fusion" is embedded into the network to aggregate the features from each layer of the two views. In conclusion, the framework we designed for DVMSCNN achieved accuracies of 0.8672, 0.8329, 0.7058, and 0.7332. When validated with a home-grown dataset, the recognition rate can reach 0.8848. In addition, better overall performance is reported compared to the state-of-the-art model. In the future, following recent progress in multi-view classification, we will expand current work to larger datasets (not only the sEMG dataset) to build more discriminative multi-view representations. Furthermore, since sEMG signals are essentially sequences of temporal data, novel temporal models, such as sequentially supervised long and short-term memory, should be employed to explore more sophisticated fusion algorithms for gesture recognition.

## REFERENCES

[1] B. Hudgins, P. Parker, and R. N. Scott, "A new strategy for multifunction myoelectric control," *IEEE Trans. Biomed. Eng.*, vol. 40, no. 1, pp. 82–94, Jan. 1993.

[2] S. Poularakis and I. Katsavounidis, "Low-complexity hand gesture recognition system for continuous streams of digits and letters," *IEEE Trans. Cybern.*, vol. 46, no. 9, pp. 2094–2108, Sep. 2016.

[3] J. Wan, G. Guo, and S. Z. Li, "Explore efficient local features from RGB-D data for one-shot learning gesture recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 8, pp. 1626–1639, Aug. 2016.

[4] Z. Li, K. Gavrilyuk, E. Gavves, M. Jain, and C. G. M. Snoek, "VideoLSTM convolves, attends and flows for action recognition," *Comput. Vis. Image Understand.*, vol. 166, pp. 41–50, Nov. 2018.

[5] J. Liu, G. Wang, L.-Y. Duan, K. Abdiyeva, and A. C. Kot, "Skeleton-based human action recognition with global context-aware attention LSTM networks," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1586–1599, Apr. 2017.

[6] Y. Sun, C. Li, and G. Li, "Gesture recognition based on Kinect and sEMG signal fusion," *Mobile Netw. Appl.*, vol. 23, no. 4, pp. 797–805, Aug. 2018.

[7] J. Liu, A. Shahroudy, D. Xu, A. K. Chichung, and G. Wang, "Skeleton-based action recognition using spatio-temporal LSTM network with trust gates," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 3007–3021, Dec. 2017.

[8] T. Fan, C. Ma, and Z. Gu, "Wireless hand gesture recognition based on continuous-wave Doppler radar sensors," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 11, pp. 4012–4020, Nov. 2016.

[9] Z. Zhou, Z. Cao, and Y. Pi, "Dynamic gesture recognition with a terahertz radar based on range profile sequences and Doppler signatures," *Sensors*, vol. 18, no. 1, pp. 10–25, Jan. 2018.

[10] J.-W. Choi, S.-J. Ryu, and J.-H. Kim, "Short-range radar based real-time hand gesture recognition using LSTM encoder," *IEEE Access*, vol. 7, pp. 33610–33618, 2019.

[11] D. Iyer, F. Mohammad, and Y. Guo, "Generalized hand gesture recognition for wearable devices in IoT: Application and implementation challenges," *Springer Int. Publishing*, vol. 23, no. 4, pp. 346–355, Jul. 2016.

[12] S. Jiang, B. Lv, W. Guo, C. Zhang, H. Wang, X. Sheng, and P. B. Shull, "Feasibility of wrist-worn, real-time hand, and surface gesture recognition via sEMG and IMU sensing," *IEEE Trans. Ind. Informat.*, vol. 14, no. 8, pp. 3376–3385, Aug. 2018.

[13] B. Fang, F. Sun, H. Liu, and C. Liu, "3D human gesture capturing and recognition by the IMMU-based data glove," *Neurocomputing*, vol. 277, pp. 198–207, Feb. 2018.

[14] X. Li, O. W. Samuel, X. Zhang, H. Wang, P. Fang, and G. Li, "A motion-classification strategy based on sEMG-EEG signal combination for upper-limb amputees," *J. Neuroeng. Rehabil.*, vol. 14, no. 1, pp. 1–15, Jan. 2017.

[15] M. A. Oskoei and H. Hu, "Support vector machine-based classification scheme for myoelectric control applied to upper limb," *IEEE Trans. Biomed. Eng.*, vol. 55, no. 8, pp. 1956–1965, Aug. 2008.

[16] P. Kumar, H. Gauba, P. P. Roy, and D. P. Dogra, "Coupled HMM-based multi-sensor data fusion for sign language recognition," *Pattern Recognit. Lett.*, vol. 86, pp. 1–8, Jan. 2017.

[17] X. Shi, P. Qin, J. Zhu, M. Zhai, and W. Shi, "Feature extraction and classification of lower limb motion based on sEMG signals," *IEEE Access*, vol. 8, pp. 132882–132892, 2020.

[18] E. J. Scheme and K. Englehart, "Electromyogram pattern recognition for control of powered upper-limb prostheses: State of the art and challenges for clinical use," *J. Rehabil. Res. Develop.*, vol. 48, no. 6, pp. 643–659, Jul. 2011.

[19] Z. Lu, X. Chen, Q. Li, X. Zhang, and P. Zhou, "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices," *IEEE Trans. Human-Mach. Syst.*, vol. 44, no. 2, pp. 293–299, Apr. 2014.

[20] F. Muri, C. Carbajal, A. M. Echenique, H. Fernández, and N. M. López, "Virtual reality upper limb model controlled by EMG signals," *J. Phys., Conf. Ser.*, vol. 477, Dec. 2013, Art. no. 012041.

[21] J. Cheng, X. Chen, Z. Lu, K. Wang, and M. Shen, "Key-press gestures recognition and interaction based on SEMG signals," in *Proc. Int. Conf. Multimodal Interfaces Workshop Mach. Learn. Multimodal Interact. (ICMI-MLMI)*, 2010, pp. 1–4.

[22] M. Simao, P. Neto, and O. Gibaru, "Natural control of an industrial robot using hand gesture recognition with neural networks," in *Proc. 42nd Annu. Conf. IEEE Ind. Electron. Soc. (IECON)*, Oct. 2016, pp. 5322–5327.

[23] Y. Y. Huang, K. H. Low, and H. B. Lim, "Objective and quantitative assessment methodology of hand functions for rehabilitation," in *Proc. IEEE Int. Conf. Robot. Biomimetics*, Feb. 2009, pp. 846–851.

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

[25] W. Geng, Y. Du, W. Jin, W. Wei, Y. Hu, and J. Li, "Gesture recognition by instantaneous surface EMG images," *Sci. Rep.*, vol. 6, no. 1, pp. 1–8, Nov. 2016.

[26] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 1929–1958, Jul. 2014.

[27] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Adv. Neural Inf. Process. Syst. (NIPS)*, May 2017, pp: 1945-1953.

[28] S. Ji, W. Xu, M. Yang, and K. Yu, "3D convolutional neural networks for human action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 221–231, Aug. 2012.

[29] M. Panwar, D. Biswas, H. Bajaj, M. Jobges, R. Turk, K. Maharatna, and A. Acharyya, "Rehab-net: Deep learning framework for arm movement classification using wearable sensors for stroke rehabilitation," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 11, pp. 3026–3037, Nov. 2019.

[30] X. Zhai, B. Jelfs, R. H. M. Chan, and C. Tin, "Self-recalibrating surface EMG pattern recognition for neuroprosthesis control based on convolutional neural network," *Frontiers Neurosci.*, vol. 11, pp. 379–386, Jul. 2017.

[31] N. Tsagkas, P. Tsinganos, and A. Skodras, "On the use of deeper CNNs in hand gesture recognition based on sEMG signals," in *Proc. 10th Int. Conf. Inf., Intell., Syst. Appl. (IISA)*, Jul. 2019, pp. 1–4.

[32] G. Peano, "Sur une courbe, qui remplit toute une aire plane," *Mathematische Annalen*, vol. 36, no. 1, pp. 157–160, Mar. 1890.

[33] P. Tsinganos, B. Cornelis, J. Cornelis, B. Jansen, and A. Skodras, "A Hilbert curve based representation of sEMG signals for gesture recognition," in *Proc. Int. Conf. Syst., Signals Image Process. (IWSSIP)*, Jun. 2019, pp. 201–206.

[34] L. Chen and Y. Hao, "Feature extraction and classification of EHG between pregnancy and labour group using Hilbert-Huang transform and extreme learning machine," *Comput. Math. Method Med*, vol. 9, pp. 1–10, Jul. 2017.

[35] J. Kurek, B. Swiderski, and S. Osowski, "Deep learning versus classical neural approach to mammogram recognition," *Bull. Pol. Acad. Sci.-Tech. Sci*, vol. 66, no. 6, pp. 831–840, Jun. 2018.

[36] W. Wei, H. Hong, and X. Wu, "A hierarchical view pooling network for multichannel surface electromyography-based gesture recognition," *Comput. Intell. Neurosci.*, vol. 65, no. 3, p. 13, Jun. 2021.

[37] W. Wei, Q. Dai, Y. Wong, Y. Hu, M. Kankanhalli, and W. Geng, "Surface-electromyography-based gesture recognition by multi-view deep learning," *IEEE Trans. Biomed. Eng.*, vol. 66, no. 10, pp. 2964–2973, Oct. 2019.

[38] C. Gotsman and M. Lindenbaum, "On the metric properties of discrete space-filling curves," *IEEE Trans. Image Process.*, vol. 5, no. 5, pp. 794–797, Feb. 1996.

[39] B. Moon, H. V. Jagadish, C. Faloutsos, and J. H. Saltz, "Analysis of the clustering properties of the Hilbert space-filling curve," *IEEE Trans. Knowl. Data Eng.*, vol. 13, no. 1, pp. 124–141, Jan. 2001.

[40] G. Huang, D. Chen, T. Li, F. Wu, L. van der Maaten, and K. Q. Weinberger, "Multi-scale dense networks for resource efficient image classification," 2017, *arXiv:1703.09844*.

[41] M. Atzori, A. Gijsberts, C. Castellini, B. Caputo, A.-G.-M. Hager, S. Elsig, G. Giatsidis, F. Bassetto, and H. Müller, "Electromyography data for non-invasive naturally-controlled robotic hand prostheses," *Sci. Data*, vol. 1, no. 1, pp. 1–13, Dec. 2014.

[42] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *Comput. Sci. Data*, vol. 63, no. 3, pp. 1–15, Sep. 2015.

[43] W. Wei, Y. Wong, Y. Du, Y. Hu, M. Kankanhalli, and W. Geng, "A multi-stream convolutional neural network for sEMG-based gesture recognition in muscle-computer interface," *Pattern Recognit. Lett.*, vol. 119, pp. 131–138, Mar. 2019.

[44] S. Pizzolato, L. Tagliapietra, and M. Cognolato, "Comparison of six electromyography acquisition setups on hand movement classification tasks," *PLoS One*, vol. 12, no. 10, pp. 1–17, Oct. 2017.

**YAN ZHANG** was born in Shijiazhuang, Hebei, China, in 1975. She received the Ph.D. degree in control theory and control engineering from Nankai University, Tianjin, in 2004. From 1999 to 2020, she worked with the Hebei University of Technology. Since 2012, she has been a Professor. Her research interests include nonlinear control intelligent rehabilitation technical aids, pattern recognition, and intelligent control.

**FAN YANG** was born in Ulanqab, Inner Mongolia, China, in 1997. He received the B.S. degree from Chang'an University. He is currently pursuing the M.S. degree in control theory and control engineering with the Hebei University of Technology, Tianjin, China. His research interests include human–computer interaction and upper limb exoskeleton.

**QI FAN** was born in Zhangjiakou, Hebei, China, in 1997. He received the B.S. degree from the Hebei University of Science and Technology. He is currently pursuing the M.S. degree in control theory and control engineering with the Hebei University of Technology, Tianjin, China. His research interests include human–computer interaction and lower extremity mobility aids.

**ANJIE YANG** was born in Lu'an, Anhui, China, in 1997. He received the bachelor's degree in electrical engineering and intelligent control engineering from the Anhui University of Science and Technology, in 2020. He is currently pursuing the master's degree with the School of Artificial Intelligence and Data Science, Hebei University of Technology. His research interests include upper limb exoskeleton and corresponding control algorithms.

**XUAN LI** was born in Xingtai, Hebei, China, in 1997. She received the bachelor's degree in measurement and control technology and instrument from the Hebei University of Science and Technology, in 2019. She is currently pursuing the master's degree with the School of Artificial Intelligence and Data Science, Hebei University of Technology. Her research interests include lower limb exoskeleton and corresponding control algorithms.