IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 30, 2021, accepted February 23, 2021, date of publication March 2, 2021, date of current version March 10, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3063131

Motion Recognition Based on Sum of the Squared Errors Distribution

WENKANG YANG, QIAOLING DU, JIANCHAO CUI, YANKAI WANG, XINPO LU, CHUNXIAO QI, AND GUIPING ZHANG

State Key Laboratory on Integrated Optoelectronics, College of Electronic Science and Engineering, Jilin University, Changchun 130012, China

Corresponding author: Qiaoling Du (duql@jlu.edu.cn)

ABSTRACT Human motion recognition is playing an increasingly important role in modern society. Direct recognition of complex motions has great limitations, so we usually study basic motions first. The key to the establishment of the basic motion set is how to determine the number of basic motions. Sum of the Squared Errors Distribution (SSED) is therefore proposed to determine the number of clustering classes of Self Organizing Maps (SOM). Secondly, the Weighted Tangent Segmentation (WTS) is also proposed to segment complex motions into simple ones, and the sequence with the time stamp is generated. Finally, Back Propagation with the time stamp (BPTS) is proposed to classify the simple motions according to the basic motion set, and the complex motion is recognized according to the time stamp. The motions of human upper limbs are used to verify the effectiveness of this method. SSED determines that the number of clusters of human upper limb motion is 10. The experimental results show that the correct segmentation rate (CSR) is 98.13%. The recognition rate of basic motions is 98.67% and 99.33% in user independent (UI) and user dependent (UD) experiments, respectively. The recognition rate of complex motions is 96.37%. Experiments on UI and UD recognitions verified the effectiveness of our algorithm. Compared with other recognition algorithms, the complex motion recognition method proposed in this article has better recognition performance.

INDEX TERMS Clustering algorithms, multi-layer neural network, object recognition, object segmentation, self organizing feature maps.

I. INTRODUCTION

Human motion recognition is a key technology in various fields including consumer electronics [1] and health care [2], [3]. For instance, Tewari designed a weak model of gesture recognition based on five postures in 2017 to solve the problem of non-contact interaction in cars [4]. Lu designed a smart wheelchair motion control system based on gesture recognition for people with physical disabilities [5], In recent years, motion recognition technology based on vision has been rapidly developed, such as flow-guided (FG) dynamic maps [6], Residual Network (ResNet) [7], Multiple Batches of Motion History Images (MB-MHIs) [8], skeleton edge motion networks (SEMN) [9],deep convolutional neural network (DCNN) [10], and Random forest (RF) [11]. Most of these methods use image or video information obtained by cameras to perform human motion recognition. However, there are some limitations, such as the camera's field of view, the complexity of target background, the intensity of light, and the privacy of the user. With the development of sensor technology, non-visual motion recognition technology has been developed rapidly, such as Hidden Markov Model (HMM) [12], Support Vector Machine (SVM) [13], Principal component analysis (PCA) [14], K-Nearest Neighbor (KNN) [15], Neural Network (NN), Singular value decomposition (SVD), and Wavelet packet transform (WPT). Most of these algorithms use surface electromyography (SEMG) signal acquisition sensor and inertial sensors in the process of data collection. SEMG refers to the electrical signal collected during muscle contraction. During human movement, the muscle contractions of different movements may be the same or very similar. Therefore, the use of SEMG to recognize human movements has certain limitations. [16] In this article, we use inertial sensors to study human action recognition.

Human motion is complex and diverse, and direct recognition has great difficulties. In order to fully recognize the human posture, it is necessary to segment the complex

The associate editor coordinating the review of this manuscript and approving it for publication was Amir Masoud Rahmani.

postures into simple ones [17]–[19]. The time series segmentation algorithm can be divided into three methods: limiting the number of segments [20], [21], fitting error of a given segment [22], and segmenting by special points [23]. The special point segmentation uses the characteristics of the sequence itself to determine the segmentation point, which can avoid the omission of important information. It can maintain the same shape and characteristics as the original sequence, and has high compression ratio and small fitting error [24]. According to this method, we propose a weighted tangent segmentation (WTS) to segment complex motions into simple unidirectional motions. In this method, the minimum value of weighted tangent is used to determine the segmentation point. Through simple motion recognition, we can recognize arbitrarily complex motions composed of simple motions.

Through the segmentation of complex motions, we have obtained simple motions, but how to classify these motions is still an issue. Common clustering algorithms can be divided into hierarchical method [25]–[28], partition method [29]–[32], density method [33]–[36], grid method [37], [38], statistical method [39]–[42], and neural network method [43], [44]. The hierarchical clustering algorithm has a simple structure, but it is not sensitive to the input order of samples, and the algorithm time complexity is large. The partition-based clustering algorithm has low time complexity, but is sensitive to outliers and noise points. The density-based clustering algorithm has a poor clustering effect when the density is uneven or the clustering distance is very different. The grid-based clustering algorithm has a fast processing speed, but the accuracy is not high. The statistics-based clustering algorithms are generally based on probability models, which are not efficient in execution and require the attributes to be independent of each other. The SOM (Self Organizing Maps) algorithm based on neural network is an unsupervised competitive learning clustering algorithm. The biggest feature of SOM is that it can map any high-dimensional input to a one-dimensional or two-dimensional output layer and keep its topological structure unchanged. This classification method reflects the essential difference between sample sets and greatly reduces the interference of human factors in the principle of consistency [45]. Therefore, we use the SOM to cluster simple motions. However, it is not easy to determine the number of clusters. In this article, SSED (Sum of the Squared Errors Distribution) based on SOM is proposed to determine the optimal number of clusters.

Aiming to recognize complex motions, the simple motions formed by segmentation need to be classified and recognized on the basic motion set. Classification algorithms mainly include Naive Bayesian Classifier (NBC) [46], Logistic Regress (LR) [47], Decision Tree algorithm (DT) [48], SVM [49], K-Nearest Neighbor (KNN) [50], and Neural Network (NN) [51]–[53]. NBC is based on the assumption of conditional independence, which will affect the classification effect in practical applications. LR cannot be used to solve nonlinear problems. DT is prone to overfitting, and it is easy

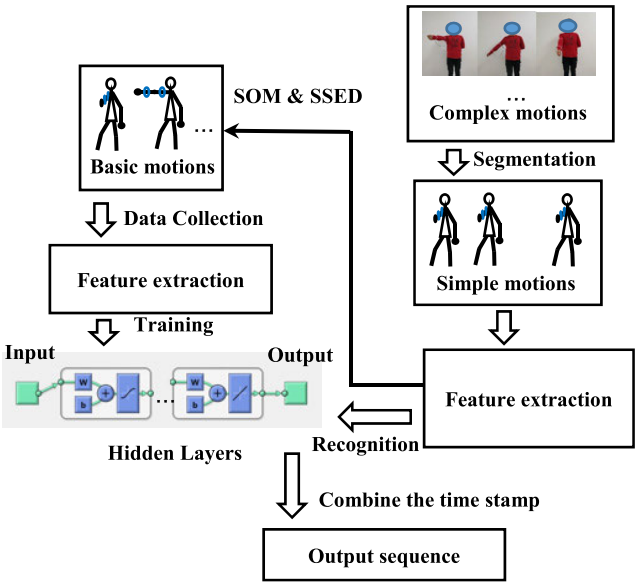


FIGURE 1. Flow chart of human motion recognition system.

to ignore the correlation of attributes in the data set. SVM is very sensitive to missing data and has greater difficulties in solving multi-classification problems. KNN relies heavily on data and has poor tolerance to training data. The neural network does not have these limitations, and it has the ability of self-adaptation and self-organization, and can change the value of the synaptic weight during the learning or training process to meet the needs of the surrounding environment.

Neural networks can be divided into feedforward neural networks (perceptron, back propagation neural network (BPNN), and convolutional neural network (CNN)), feedback neural networks (Elman and Hopfield neural networks), and self-organizing map (SOM) neural networks according to the interconnection of neurons in the network. The perceptron neural network can only handle linear problems [54]. BP neural network adds the number of layers and BP algorithm on the basis of the perceptron, and has strong nonlinear mapping and optimization computing ability [55]. CNN is used for image processing and classification [56]. The feedback network is a dynamic network, and its structure is much more complicated than the feedforward neural network. It needs to meet the stability conditions before it can work normally. Elman neural network mainly used for prediction [57], and Hopfield neural network is mainly used for associative memory and optimization calculation [58]. SOM is an unsupervised neural network and is mainly used for clustering [59], [60]. Therefore, we choose a supervised BPNN for classification in this article.

BPNN uses pre-stored information and learning mechanism for adaptive training, which can recover the original complete information from incomplete information and noise interference, and it has strong recognition and classification capabilities for external input samples.

An overview of our approach to recognize human motion is sketched in Fig. 1. Firstly, WTS is used to divide complex

motions into simple ones. Secondly, eigenvalue extraction is performed on simple motions. Then use SOM based on SSED to determine the number of basic motion clusters and form basic action set. Finally, use BPTS to identify basic motions, and complete the identification of complex motions according to the timestamp. The whole complex motion recognition process mainly includes two aspects. One is the formation of the basic motion set, which can be divided into four steps. ① divide complex motions into basic ones by using WTS. ② extract eigenvalues of the basic motions. ③ use SSED to determine the optimal number of clusters for basic motions. ④ use SOM to cluster the basic motions and form a basic motion set. The other is the recognition of complex motions, which can also be divided into four steps. The first two steps are the same as those of the first module. ③ use BPNN to recognize the basic motions. ④ use time stamps to recognize the complex motions.

This article mainly has the following contributions:

1. SSED is proposed to determine the optimal number of clusters.

2. SOM is used to cluster simple motions and form basic motion set. This method uses machine learning, which can avoid the cumbersome human classification and the influence of subjective factors.

3. Combining WTS, SOM, BPNN, and timestamp together, a method for recognizing complex motions is proposed.

4. Based on the experimental data of human upper limb motions, the feasibility and reliability of complex motion recognition are verified.

II. DEFINITIONS AND FORMULATIONS

This Section provides a unified overview of all the definitions and formulations that appear in this entire article. The specific contents are:

Definition 1: Complex motions are defined as motions which include at least one change of direction in space.

Definition 2: Simple motions are defined as motions which include no change of direction in space.

Definition 3: Basic motion set is a group of relatively homogeneous simple motions.

Definition 4: Basic motion is an element of the basic motion set.

Definition 5: Sum of Squares for Error (SSE) is the sum of the squares of distances between the eigenvalues of $\{a_j\}$ and the nodes in the competitive layer.

Definition 6: We define the relative invariance in complex motions as the interval characteristics.

Definition 7: Reorganization of M_i is to reset the basic motion $\{b_i\}$ to the original complex motion sequence corresponding to the time stamps 1, 2, 3, . . . , p of $\{s_i\}$.

Formulation: Give a motion sequence set $\{D(n)_i\}$, $\forall\varepsilon > 0$, $\exists K \in Z$, Z represents the set of integers. When $k > K$, the SSE converges in distribution, that is SSE $< \varepsilon$.

III. SOM CLUSTERING COMBINED WITH SSED

In order to construct a basic motion set, we first decompose the complex motions into simple motions (see in Section IV.A for details). Secondly, the basic action set is determined by simple actions. The key to determine the basic motion set is the specific number of basic motions. Most people use the artificial definition method to determine the number, which requires cumbersome human classification and is affected by subjective factors. In this section, SSED based on SOM is proposed to determine the number of basic motions, and it uses machine learning to overcome the influence of human factors. Thirdly, SOM is used to cluster and form a basic motion set according to the number by SSED. Finally, the basic motion set is mapped into specific motions according to eigenvalue analysis.

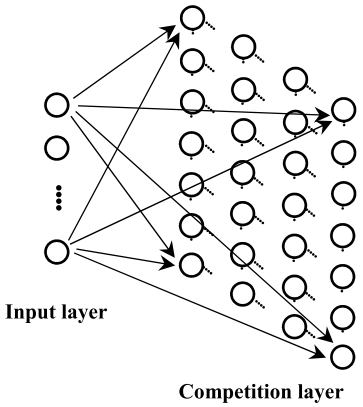


FIGURE 2. SOM neural network structure.

A. DESCRIPTION OF TIME SERIES

In order to represent the sequence of human motion, three-dimensional Euler angle is used in this article. At each moment, the position where the sensor is worn can be described as a point in the three-dimensional space according to Euler angle. The measurements of the sensor in three dimensions at the i_{th} moment can be described as $d(i) = (d(x(i)), d(y(i)), d(z(i)))\ (0 \le i \le n)$, where $d(x(i))$ is the pitch value of Euler angle, $d(y(i))$ is the roll value, and $d(z(i))$ is the yaw value. Let $D(n)$ represent the human body motion sequence.

$$D(n) = \{d(0), d(1), d(2) \ldots d(n)\} \tag{1}$$

where n is the number of sampling points. This time series is used for motion segmentation and feature extraction.

B. SOM CLUSTERING

SOM performs unsupervised learning and clustering. It is a neural network composed of only two layers: an input layer and a competition layer. The nodes in the competition layer and the input layer are fully connected. And each node in the competition layer represents a class to be clustered, as shown in Fig. 2. The training process adopts the ''competitive learning'' mode, and each input finds the optimal matching node in the competition layer, which is the active node. ''Competitive learning'' mode is based on the principle of minimum distance. Let $x = \{x_1, x_2, x_3 \ldots\}$ represent the eigenvalue

of $D(n)$. The distance between x and the j_{th} competition layer node is calculated as shown in (2).

$$d_j(x) = \sum_{i=1}^{m} (x_i - w_{ji})^2 \tag{2}$$

where m is its dimension. w_j is the weight vector of the j_{th} competition layer node, and $d_j(x)$ is the distance from x to the j_{th} node.

w_j is updated according to the stochastic gradient descent method, and the points near the j_{th} competition layer node are also updated appropriately according to their distance. The update process is shown in (3)-(4).

$$T_{j,I(x)} = \exp(-S^2_{j,I(x)}/2\sigma^2) \tag{3}$$

$$\Delta w_{ji} = \eta(t) T_{j,I(x)}(t)(x_i - w_{ji}) \tag{4}$$

where $I(x)$ is the active node, $S_{j,I(x)}$ is the distance from node j to the active node, σ is the step, $T_{j,I(x)}$ is the weight of σ, $\eta(t)$ is the learning rate, and Δw_{ji} is the change value of w_{ji}.

C. SSED

In clustering algorithm, the most difficult problem is how to determine the number of clusters. For example, in K-means and SOM, the number of categories is usually given in advance. When the number of clusters is small, enumeration method combined with various indicators can be used to measure the optimal number [61]. When the number of clusters is large, learning algorithm can be used to modify the number of clusters in the current state with the previous state [62].

Common methods to determine the optimal number of clusters include NbClust [63], Calinsky criterion, Gap Statistic [64], [65], Silhouette Coefficient [66], [67], Sum of the Squared Errors (SSE) [68]. However, the computational complexity of NbClust, Calinsky criterion and Gap Statistic is too high, and they are not suitable for high-dimensional data [64]. Silhouette Coefficient has a large uncertainty and calculation amount in a complex data set, and the results of multiple calculations have large deviations [67]. In this article, we extract 12 eigenvalues of upper limb movement in three-dimensional space. The data dimensions are high, and the movement data are complex and diverse. Therefore, methods such as gap statistics and contour coefficients are not suitable for our data. In this article, the SOM-based SSED is proposed to determine the optimal number of clusters. The calculation formula of SSE is as follows.

$$SSE = \sum_{j=1}^{k} \sum_{x \in a_j} dist(x - w_j)^2 \tag{5}$$

where k is the number of clusters, a_j stands for class j of SOM output, and x describes the eigenvalue of each motion sequence of a_j.

SSE reflects the similarity between the data objects in the same category and the neurons in the competition layer. The SSE values reflect the similarity and clustering performance.

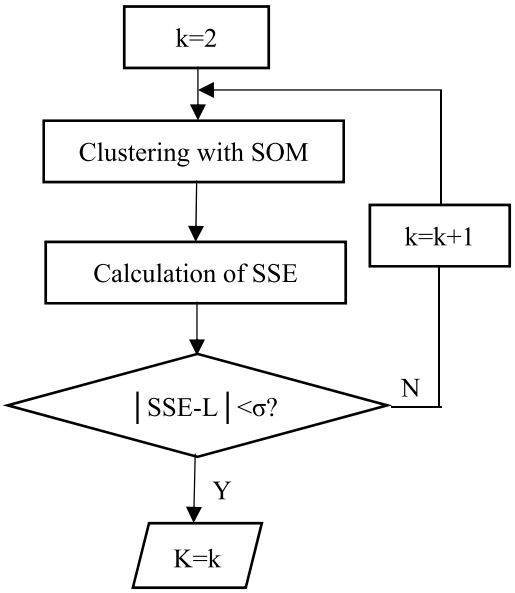


FIGURE 3. Determining process of the optimal number of K.

Its small value represents a large similarity and good clustering effect.

$$\lim_{k \to \infty} SSE - \varepsilon < 0 \tag{6}$$

Set $L = \lim_{k \to \infty} SSE$. L is determined by calculating the SSED. The calculation steps of SSED are as follows.

1. Firstly segment complex motions into simple ones.
2. Set *step*.
3. k is enumerated by *step*.
4. Calculate SSE_k based on SOM according to (5).
5. SSED can be obtained by fitting SSE_k.

step is the step size of k, and is set according to the specific classification object. If the number of clusters is small, *step* is set to small; if the number of clusters is large, *step* is set to large. The calculation process of K is shown in Fig. 3.

In this article, the upper limb movements of human body are sampled as tested sequences. WTS is used to segment complex motions into simple motions. Set $step = 1$, and k is from 2 to 20. Then for each value of k we use SOM to cluster on the test set to get the network structure, and calculate the SSE on the validation set under this structure. The test set include 200 simple motions, and the verification set include 100 simple motions that is divided into 5 groups $\{g_1, g_2, g_3, g_4, g_5\}$.

The SSE curve is shown in Fig. 4, where k is the number of clusters, and No. 1 – No. 5 are the curves obtained on the five groups of validation sets. It can be found that No. 1 – No. 5 have similar trends. We calculate the average of No. 1 – No. 5, and get the curve shown in Fig. 5. By fitting this curve, we get the fitting formula shown in (7). The limit of SSE is calculated in (8).

$$SSE = 6.44 + \frac{13.51}{1 + e^{\frac{k-3.64}{1.68}}} \tag{7}$$

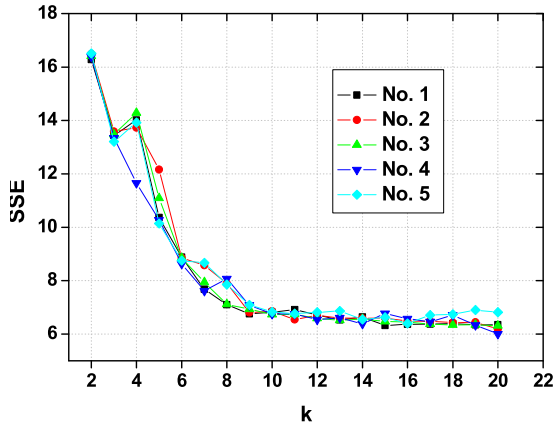


FIGURE 4. SSE value under different number of clusters.

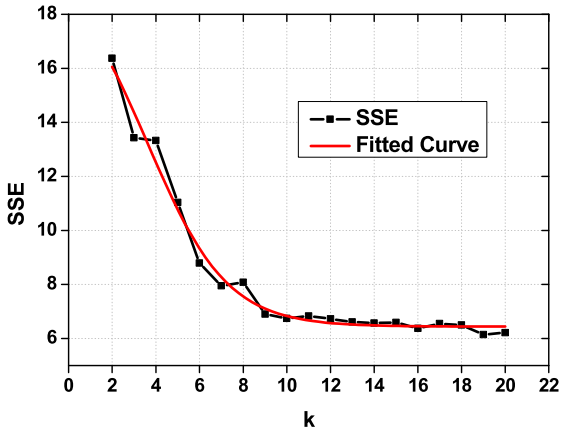


FIGURE 5. Average curve and fitting curve of the five sets of SSE data.

$$L = \lim_{k\to\infty}(6.44 + \frac{13.51}{1+e^{\frac{k-3.64}{1.68}}}) = 6.44 \tag{8}$$

For $\forall\varepsilon > 0$, when $k > 1.68 * \ln(\frac{13.51}{\varepsilon} - 1) + 3.64$, $\left|6.44 + \frac{13.51}{1+e^{\frac{k-3.64}{1.68}}} - 6.44\right| < \epsilon$. It can be proved that the SSE curve converges to 6.44.

Let σ represent the threshold. $\exists K \in Z$, when $k > K$, (9) is satisfied. Then K is the optimal number of clusters. ΔSSE on $\{g_1, g_2, g_3, g_4, g_5\}$ are shown in Table 1.

$$\Delta SSE = SSE - 6.44 < \sigma \tag{9}$$

Set σ be 5% of L. Then $\sigma = 0.32$, and we get $K = 10$ according to (7) and (9). From the experimental results in Table 1, we can obtain that when k $\geq$ 10, (9) is satisfied.

Therefore, we can determine the optimal number of clusters according to SSED.

D. MAPPING CLUSTERING TO DISTINCT ACTIVITIES

SOM is adopted to cluster simple motions according to $K = 10$ and get $A = \{a_1, a_2, a_3 \ldots, a_{10}\}$. a_i is a class in the clustering result of SOM. However, the specific motions represented by $\{a_i\}$ are not clear.

We select 20 groups in each cluster, and then calculate the average of their eigenvalues, as shown in Table 2. The

TABLE 1. The absolute value of the difference between SSE and L on No. 1 – No. 5.

k	No.1	No.2	No.3	No.4	No.5	Average
2	9.84	10.05	9.97	9.95	10.07	9.98
3	7.00	7.15	7.04	6.90	6.77	6.97
4	7.58	7.29	7.85	5.21	7.48	7.08
5	3.92	5.72	4.65	3.83	3.70	4.37
6	2.44	2.42	2.41	2.19	2.31	2.35
7	1.25	2.14	1.50	2.18	2.23	1.86
8	0.86	1.46	1.07	1.63	1.42	1.29
9	0.56	0.90	0.69	0.85	1.08	0.82
10	0.05	0.31	0.29	0.32	0.25	0.24
11	0.17	0.10	0.20	0.32	0.13	0.18
12	0.28	0.27	0.15	0.10	0.38	0.23
13	0.09	0.17	0.08	0.17	0.42	0.19
14	0.20	0.16	0.12	-0.06	0.11	0.11
15	0.02	0.17	0.03	0.34	0.18	0.14
16	-0.02	0.05	0.01	0.14	-0.04	0.04
17	0.04	0.04	0.04	0.03	0.27	0.08
18	-0.04	-0.01	-0.09	0.27	0.30	0.08
19	0.00	0.01	0.02	-0.10	0.46	0.08
20	0.01	-0.25	-0.13	-0.03	0.38	0.00

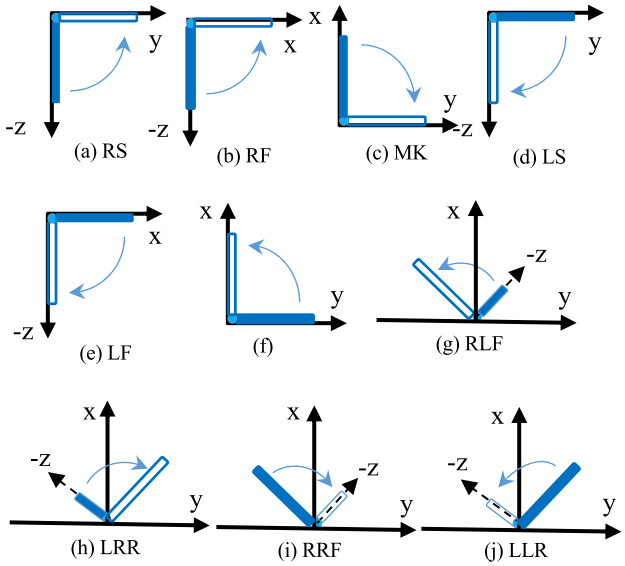


FIGURE 6. Schematic diagram of 10 basic motions, the x-axis direction is the front of the human body, the y-axis direction is the right side of the human body, the z-axis direction is pointing to the ground, and the -z-axis direction is pointing to the sky.

eigenvalue can reflect the position information of the motion. According to *MAX* and *MIN*, the starting and ending positions of the motion can be obtained. *AVG* reflects the middle position of the motion, and *SD* can reflect the fluctuation of the motion. The eigenvalue changes of a_1 and a_2 in Table 2 are mainly reflected in the pitch, as shown in Fig. 6.a and 6.d. According to the position information contained in the eigenvalue, a_1 and a_2 are mapped into activities as shown in Fig. 7.a and 7.d. The eigenvalue changes of a_3 and a_4 in Table 2 are mainly reflected in Roll, as shown in Fig. 6.b and 6.e, and the mapped activities are shown in Fig. 7.b and 7.e. The eigenvalue changes of a_5 and a_6 in Table 2 are mainly reflected in Yaw, as shown in Fig. 6.c and 6.f, and the mapped activities are shown in Fig. 7.c and 7.f. The eigenvalue changes of

TABLE 2. Feature values of each cluster set.

Simple Motion	Pitch				Roll				Yaw				Basic motion
	MIN	MAX	AVG	SD	MIN	MAX	AVG	SD	MIN	MAX	AVG	SD	
a_1	-73.9	0.02	-35.17	25.71	-5.94	1.32	-2.86	1.53	-8.02	2.5	-3.71	1.82	RS
a_2	-0.21	72.5	40.42	25.36	-7.74	8.58	0.59	1.56	-4.40	9.92	3.39	2.84	LS
a_3	-1.63	4.93	0.87	1.51	-78.6	1.59	-35.4	28.57	-3.49	2.16	0.29	1.13	RF
a_4	0.05	7.74	3.43	3.68	-5.18	76.7	40.33	34.07	-6.28	3.35	-1.63	2.86	LF
a_5	-3.14	5.93	2.24	2.09	-2.24	7.61	2.94	3.77	-70.96	5.17	-38.19	31.66	MR
a_6	-0.96	3.55	1.37	5.11	-4.89	2.27	-0.94	3.55	-1.14	78.63	41.46	34.32	ML
a_7	-0.06	8.74	4.37	2.91	0.67	86.3	44.13	30.73	2.75	43.51	22.61	15.78	RLF
a_8	-6.33	0.18	-2.59	1.08	-87.91	1.28	-47.8	36.86	-54.31	0.84	-23.76	19.48	LRR
a_9	-7.28	3.72	-3.25	3.51	-0.21	84.5	36.37	31.29	-43.98	0.42	-19.68	13.64	RRF
a_{10}	-6.91	1.79	-4.86	4.77	-83.36	0.08	-45.5	35.04	-0.96	49.46	20.75	18.62	LLR

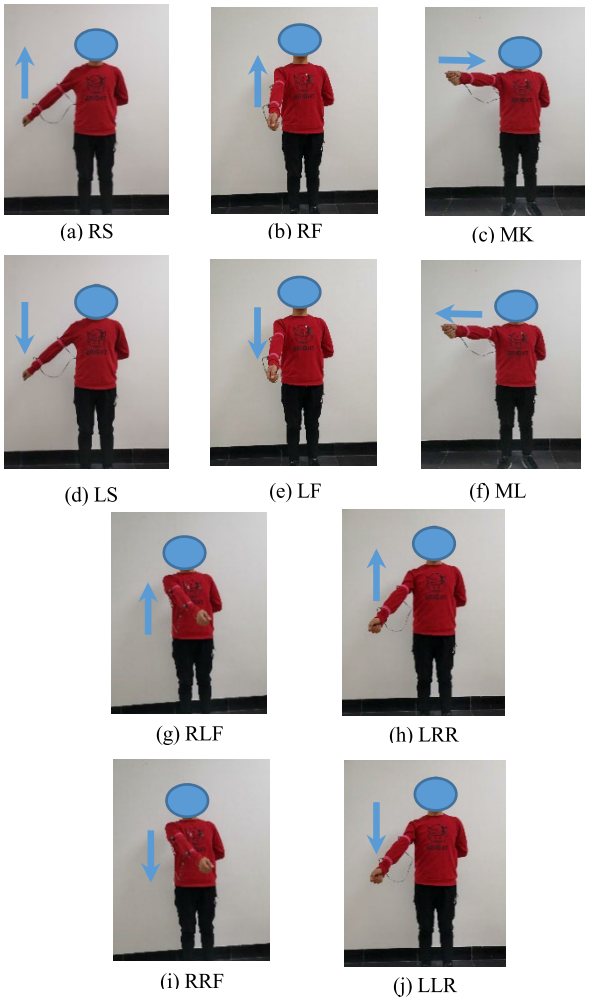


FIGURE 7. The mapped activities of 10 basic motions.

$a_7 - a_{10}$ in Table 2 are mainly reflected in Roll and Yaw, as shown in Fig. 6.g-6.j, and the mapped activities are shown in Fig. 7.g-7.j. According to the mapping results of simple motions, we get 10 basic motions $\{b_1, b_2, b_3 \ldots, b_{10}\}$ and basic motion set.

IV. COMPLEX MOTION RECOGNITION BASED ON WTS-BPTS

The recognition of complex motions based on the basic motion set is divided into the following steps. Firstly, complex motions are collected using the human posture acquisition device (HPAD-I). Secondly, the collected data are preprocessed. Thirdly, the complex motion M_i is decomposed into simple motion $\{s_i\}$ by WTS. $\{s_{i1}, s_{i2}, s_{i3} \ldots, s_{ip}\}$ represents the time series of simple motions corresponding to M_i, and the serial number of the subscript 1, 2, 3, . . . , p are used as the time stamps. Fourthly, the eigenvalues of simple motions are extracted. Fifthly, the BPNN is used to identify the simple motions based on $A = \{a_1, a_2, a_3 \ldots, a_K\}$. Finally, the basic motion corresponding to the simple motion s_i is reset to the sequence $\{b_i\}$ according to the time stamp.

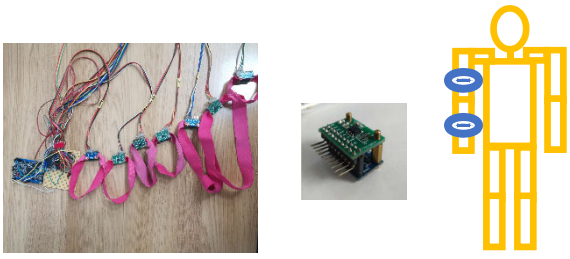


FIGURE 8. HPAD-I, attitude acquisition node and its wearing position.

A. DATA COLLECTION

Our laboratory has developed HPAD-I, where the nine-axis sensor MPU9250 is used to collect motion data, as shown in Fig. 8a. MPU9250 is composed of a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The signals output by the three modules meet the complementary relationship of frequency, and the data are fused to make the output more accurate and reliable. The frequently used communication protocol of MPU9250 is the IIC protocol, which converts analog signals into digital signals for output through its internal 16-bit ADC. However, the IIC protocol is restricted by the capacitance on the bus. If the transmission distance is too long, the capacitance on the wire will exceed the maximum allowable value, which will cause the data transmission to fail. In this article, we choose an IIC to UART module to change the communication mode between the data processing node and the data acquisition node, which can increase the transmission distance. STM32F103VET6 was

selected as the core chip to collect and process data, and two analog switches of CD4067BE were used to select and control the two sensors. HPAD-I has 7 attitude acquisition nodes, and we used two of them in this article. Attitude acquisition node of HPAD-I has a length of 2.5 cm, a width of 1.8 cm, and a height of 1.5 cm, as shown in Fig 8b. The wearing positions are shown in Fig 8c, and the X axis of HPAD-I should point to the left side of the human body, the Y axis should point to the front, and the Z axis should point vertically to the ground. The data sampling rate is 13 Sps.

Through MPU9250, we can collect quaternion Q. Compared with quaternion, Euler angle data have fewer dimensions and are more intuitive, which are convenient to analyze and process. Although Euler angles have the phenomenon of gimbal lock, the movement space of the human arm itself has some limitations, and it is difficult to trigger the gimbal lock within its movement range. (10)-(13) are used to convert the quaternion into Euler angles.

$$Q = q_0 + q_1\mathrm{i} + q_2\mathrm{j} + q_3\mathrm{k} \tag{10}$$

$$x_s = \arctan\left(\frac{2(q_1q_2 - q_0q_3)}{q_0^2+q_1^2-q_2^2-q_3^2}\right) \tag{11}$$

$$y_s = \arcsin(-2(q_0q_2 + q_1q_3)) \tag{12}$$

$$z_s = \arctan\left(\frac{2(q_2q_3 - q_0q_1)}{q_0^2-q_1^2-q_2^2+q_3^2}\right) \tag{13}$$

where q_0, q_1, q_2, and q_3 are the four parameters corresponding to the quaternion, i, j, and k represent the orthonormal basis of three-dimensional space, and x_s, y_s, and z_s are the Yaw, Pitch, and Roll calculated from the quaternion respectively.

B. DATA PREPROCESSING

Let $Ds\,(n)$ represent the raw of human motion.

$$Ds\,(n) = \{ds\,(0)\,, ds\,(1)\,, ds\,(2)\ldots ds\,(n)\} \tag{14}$$

where $ds\,(i) = (x_s(i), y_s(i), z_s(i))\ (0 \leq i \leq n)$. Two kinds of errors are encountered during data collection:

1. Abnormal data are different from the surrounding data, which seriously affect the extraction of *MAX* and *MIN*.

2. Data loss causes null values. Data loss not only affects *AVG* and *SD*, but also affects the selection of segmentation points in motion segmentation algorithm.

We set the thresholds to solve abnormal data. By comparing the difference between adjacent data, abnormal data are found and deleted. If any of (15)-(17) is satisfied, $ds\,(i)$ is regarded as abnormal.

$$x_s\,(i) - x_s\,(i-1) > \sigma_x \tag{15}$$

$$y_s\,(i) - y_s\,(i-1) > \sigma_y \tag{16}$$

$$z_s\,(i) - z_s\,(i-1) > \sigma_z \tag{17}$$

where σ_x is the threshold of pitch, σ_y is the threshold of roll, and σ_z is the threshold of yaw. We use the average value of $ds\,(i-1)$ and $ds\,(i+1)$ instead of $ds\,(i)$, as shown in (26).

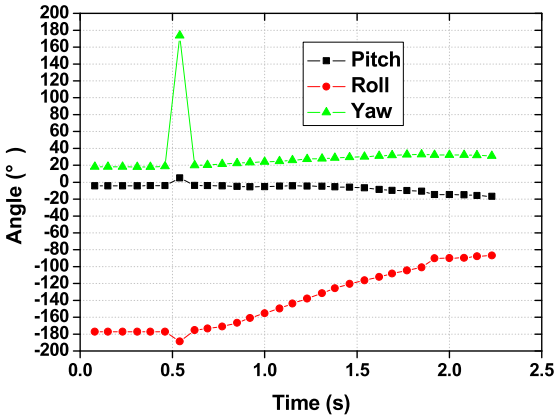


FIGURE 9. Sampling points before preprocessing operation.

Through the analysis of experimental data, set $\sigma_x = \sigma_y = \sigma_z = 50$.

$$ds\,(i) = \frac{1}{2}\,(ds\,(i-1) + ds\,(i+1)) \tag{18}$$

The null values $ds\,(i)$ usually appear alone. We use the average value of $ds\,(i-1)$ and $ds\,(i+1)$ instead of $ds\,(i)$ to ensure the accuracy and continuity of the data according to (18). If there are q consecutive null values $\{ds\,(i)\,, ds\,(i+1)\,, \cdots, ds\,(i+q-1)\}$, the data compensation method is shown in (19).

$$\begin{cases} ds\,(i+j) = ds\,(i-1) + j * ds\,(*) \\ ds(*) = \dfrac{1}{q+1}(ds\,(i+q) - ds(i-1)) \end{cases} \tag{19}$$

where $0 \leq j \leq q$, $ds\,(i+j)$ is the j_{th} null value and $ds(*)$ represents the difference between two null values.

In the process of motion collection, the arm sways to a certain extent under the influence of gravity and inertial force, resulting in the collected data floating up and down. To solve this problem, we design a moving average filter to process the data as shown in (20).

$$dsf\,(i) = \frac{1}{2m+1}\sum_{j=-m}^{m} ds\,(i+j) \tag{20}$$

where n is the number of samples, m is the filter window size, and $m \leq i \leq (n-m)$. By removing abnormal data, null compensation and filtering, the original data need to be preprocessed for subsequent segmentation, shown in (21).

$$d\,(i) = dsf\,(i) - dsf\,(0)\ \ (0 \leq i \leq n) \tag{21}$$

The original data are shown in Fig. 9, and the processed data are shown in Fig. 10.

C. SEGMENTATION ALGORITHM

Complex motions usually consist of two or more simple motions, and it is not effective to recognize complex motion directly. On the contrary, random combinations of simple motions can form various complex motions. In order to recognize, it is necessary to decompose complex motions into

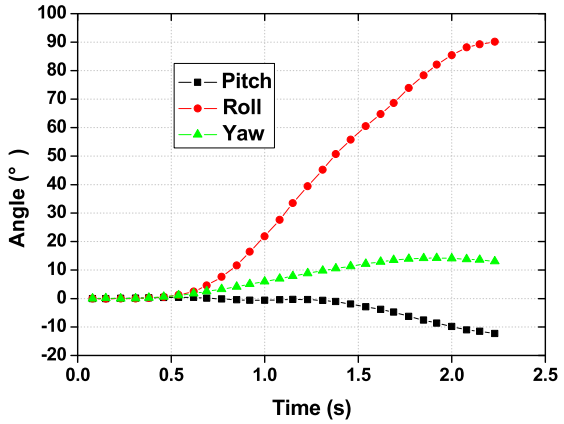
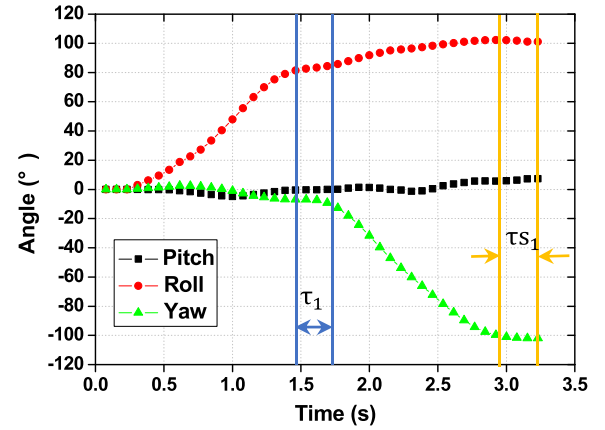
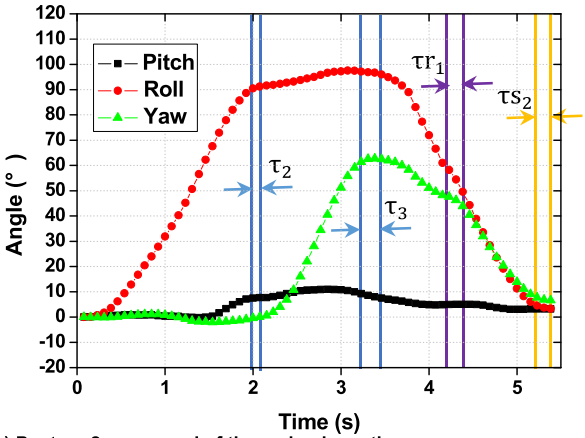


FIGURE 10. Sampling points after preprocessing operation.



(a) Posture 1 composed of two simple motions.



(b) Posture 2 composed of three simple motions.

FIGURE 11. Complex motion movement trend.

simple motions. In this article, WTS is used to segment complex motions. WTS belongs to special point segmentations, and has advantages compared with other segmentations, such as high compression ratio and small fitting error [24].

The process of 3D measurements recording complex motions is shown in Fig. 11. During the execution of the motion, some stationary regions appear, such as τ_1, τ_2 and τ_3 shown in Fig. 11. During τ_1, τ_2 and τ_3, the data of pitch, roll and yaw are relatively invariant.

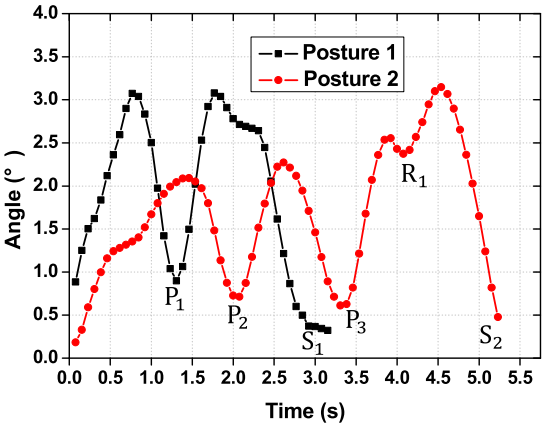


FIGURE 12. Weighted tangent values.

Let $D(n) = \{d(0), d(1), d(2) \ldots d(n)\}$ represent the time series of arbitrary complicated motion M_i, where $d(i) = (d(x(i)), d(y(i)), d(z(i)))$ $(0 \leq i \leq n)$. Interval characteristics reflect the end of the former motion $d(i)$ and the beginning of the latter motion $d(i+1)$. A complex motion is composed of many simple motions, and different simple motions have different trajectories and directions. At the end of the former motion, its movement trend will gradually weaken to zero, and at the beginning of the latter motion, it will gradually increase from zero. Therefore, the interval characteristics between two adjacent simple motions $d(i)$ and $d(i+1)$ is used as the segmentation point.

In this article, WTS is used to determine the points with interval attribute as the segmentation location. The tangents of $\{d(i)\}$ are calculated according to (22). $T(x(i))$ is the tangent value of the pitch, $T(y(i))$ is the tangent value of the roll, and $T(z(i))$ is the tangent value of the yaw. The weighted tangent value WT(i) is obtained by (23), and the curve of WT(i) is shown in Fig. 12.

$$\begin{cases} T(x(i)) = \dfrac{1}{2}\left|d(x(i+1)) - d(x(i-1))\right| \\ T(y(i)) = \dfrac{1}{2}\left|d(y(i+1)) - d(y(i-1))\right| \\ T(z(i)) = \dfrac{1}{2}\left|d(z(i+1)) - d(z(i-1))\right| \end{cases} \quad (22)$$

$$WT(i) = w_x T(x(i)) + w_y T(y(i)) + w_z T(z(i))) \quad (23)$$

where $w_x + w_y + w_z = 1$. Set $w_x = w_y = w_z$, then. $w_x = 1/3$, $w_y = 1/3$, $w_z = 1/3$

We search for local minima on $WT(i)$. Local minima are caused by three conditions. In one case, the extreme points P_1, P_2 and P_3 are caused by the motion conversions τ_1, τ_2 and τ_3 in Fig. 11, which can be used as the segmentation points. In another case, the extreme point R_1 is caused by the change of slow motion τ_{r1} in Fig. 11, which is regarded as a pseudo segmentation point and needs to be removed. And In the last case, the extreme points S_1 and S_2 are caused by the end of the motions τ_{s1} and τ_{s2} in Fig. 11, which can be removed easily. We remove the pseudo segmentation points

by setting the threshold σ_s. The local minimum points that satisfies (24) is regarded as the segmentation point, otherwise it is removed.

$$WT\ (i) < \sigma_s \tag{24}$$

$$\sigma_s = \frac{1}{n}\sum_{i=1}^{n} WT\ (i) \tag{25}$$

D. FEATURE EXTRACTION

In order to reduce the dimensions of simple motions, we extract the eigenvalues of simple motions. Complex motions contain both temporal and spatial information. Time information can be represented by time series. Spatial information includes direction and position. Based on the direction information, WTS is used to segment the complex motions, and then the simple motions are obtained which only contain the location information. Therefore, we extract the feature parameters of simple motions, which only contain location information.

The location-related feature values are as follows: RMS (root mean square), AVG (average), IQR (interquartile range), MAD (mean absolute deviation), VAR (variance value), SD (standard deviation), MAX (maximum value), and MIN (minimum value) [13], [69], [70]. RMS is used to measure the deviation between the observed value and the true value, and it is used to measure the relationship between the measured value and the true value, which is not applicable in this article. AVG is used to reflect the overall situation of the data and describes the centralized location of the data, and it can describe the central tendency of data. VAR, SD, MAD and IRQ are used to reflect the degrees of data distribution and fluctuation. IRQ does not use all the data information and is prone to deviation. MAD is not as good as SD and VAR in expressing the degree of distribution. SD has the same units as the data, and it can reflect the fluctuation of data more intuitively than VAR. MAX and MIN are used to reflect the boundary positions that the motion can reach in each dimension, and they can reflect the start and end positions of the motion, which are very important for us to recognize them.

Simple motions are sampled in three-dimensional space, we set $MAX = \{MAXx, MAXy, MAXz\}$, $MIN = \{MINx, MINy, MINz\}$, $AVG = \{AVGx, AVGy, AVGz\}$, and $SD = \{SDx, SDy, SDz\}$ as the eigenvalues of s_i, as shown in (26)-(29).

$$\begin{cases} MAXx = max\ \{s\ (x(0))\ ,\ s\ (x(2)) \ldots s\ (x(n))\} \\ MAXy = max\ \{s\ (y(0))\ ,\ s\ (y(2)) \ldots s\ (y(n))\} \\ MAXz = max\ \{s\ (z(0))\ ,\ s\ (z(2)) \ldots s\ (z(n))\} \end{cases} \tag{26}$$

$$\begin{cases} MINx = min\ \{s\ (x(0))\ ,\ s\ (x(2)) \ldots s\ (x(n))\} \\ MINy = min\ \{s\ (y(0))\ ,\ s\ (y(2)) \ldots s\ (y(n))\} \\ MINz = min\ \{s\ (z(0))\ ,\ s\ (z(2)) \ldots s\ (z(n))\} \end{cases} \tag{27}$$

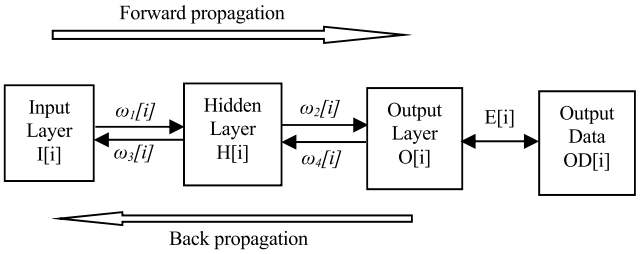


FIGURE 13. Flow chart of back propagation of BPNN.

$$\begin{cases} AVGx = \frac{1}{N}\sum_{i=0}^{N} s(x\ (i)) \\ AVGy = \frac{1}{N}\sum_{i=0}^{N} s(y\ (i)) \\ AVGz = \frac{1}{N}\sum_{i=0}^{N} s(z\ (i)) \end{cases} \tag{28}$$

$$\begin{cases} SDx = \sqrt{\frac{1}{N}\sum_{i=0}^{N} (s(x\ (i)) - AVGx)^2} \\ SDy = \sqrt{\frac{1}{N}\sum_{i=0}^{N} (s(y\ (i)) - AVGy)^2} \\ SDz = \sqrt{\frac{1}{N}\sum_{i=0}^{N} (s(z\ (i)) - AVGz)^2} \end{cases} \tag{29}$$

E. BPNN CLASSIFICATION

According to the obtained basic motion set, BPNN is used to classify simple motions. The training process of BPNN is shown in Fig. 13, where I[i] is the network input data, H[i] is the hidden layer data, O[i] is the output data calculated by forward propagation, and OD[i] is the original output data. ω_1[i] is the weight between the input layer and the hidden layer during forward propagation, and ω_2[i] is the weight between the hidden layer and the output layer during forward propagation, ω_3[i] is the update weight between the input layer and the hidden layer during back propagation, and ω_4[i] is the update weight between the input layer and the hidden layer during back propagation. E[i] is the error between O[i] and OD[i].

The number of hidden nodes is very important in the BP network, and it has a great influence on the performance of the established network model. If the number of nodes is too small, the network cannot be trained or the network performance is very poor. If there are too many nodes, the network training time will be prolonged, and the training will easily fall into the local minimum point. This is also the inherent reason for "overfitting" in the training process. Therefore, it is very important to find a network structure with a suitable hidden layer structure.

In the experiment of selecting the best network structure, we set the number of hidden layers to 1-10, and the number

TABLE 3. Recognition results of different hidden layers and different hidden layer nodes.

Hidden Layers	10 nodes	15 nodes	20 nodes	25 nodes	30 nodes	35 nodes	Aver age
1	7.67	6.33	5.00	5.00	4.67	3.33	5.33
2	6.33	6.33	6.33	3.00	8.00	3.67	5.61
3	4.67	3.67	4.67	2.00	3.00	4.67	3.78
4	9.33	2.67	5.67	3.33	6.33	4.33	5.28
5	5.67	3.67	4.33	4.00	1.33	6.00	4.17
6	3.67	4.67	5.00	1.00	3.00	3.33	3.44
7	5.67	3.67	5.00	5.00	5.00	5.67	5.00
8	10.0	8.67	2.67	2.67	4.67	8.67	6.22
9	9.00	7.00	4.33	3.67	2.00	4.00	5.00
10	10.3	13.00	4.00	5.00	3.00	6.33	6.94
Average	7.23	5.97	4.70	3.47	4.10	5.00	

of hidden layer nodes to 10-35. 10 testers are invited to participate in the collection of data. Each tester repeats each basic motion 6 times, and gets a total of 600 basic motions, and we divided the 600 basic motions into 6 groups according to the number of repetitions. Then divided the 6 groups into three parts randomly, each part includes 200 basic motions. According to the idea of cross-validation, two parts are selected as training data each time, and the third is selected for verification.

In the training process, in order to handle multi-classification problems more accurately, we choose ''traingdx'' for training function, ''logsig'' and ''softmax'' for activation function, and ''learngdx'' for learning function. The ''net.trainParam.epochs'' is set as 10000, the ''net.trainParam.goal'' is set as 10^{-10}, the ''net.trainParam.show'' is set as 100, and the ''net.trainParam.lr'' is set as 0.05.

Three sets of experiments were performed on different network structures, and the average recognition error rate was obtained, as shown in Table 3. When the hidden layer is 6 layers and the number of hidden layer nodes is 25, the recognition error rate is the smallest, and it is the best network structure.

F. REORGANIZATION OF COMPLEX MOTIONS

Simple motions $\{s_{ij}\}$ are classified to get the corresponding basic motions $\{b_{ij}\}$ according to BPNN. And the complex motion M_i is decomposed into simple motion $\{s_{i1}, s_{i2}, s_{i3} \ldots, s_{ip}\}$ by WTS. The serial number of the subscripts 1, 2, 3, . . . , p is used as the time stamps. According to the time stamp, the original complex motion M_i is reorganized M_i'. $M_i' = \{b_{i1}, b_{i2}, b_{i3} \ldots, b_{ip}\}$ contains readable basic motion, which is a recognized complex motion.

V. EXPERIMENTAL RESULTS

Multiple sets of experiments are used to verify the effectiveness of SOM combined with SSED, WTS and BPTS. First, we use WTS to segment complex motions and verify the segmentation effect. Second, we have completed SOM clustering in the upper limb motions according to the SSED. Finally, BPTS is tested to identify complex motions on the verification set.

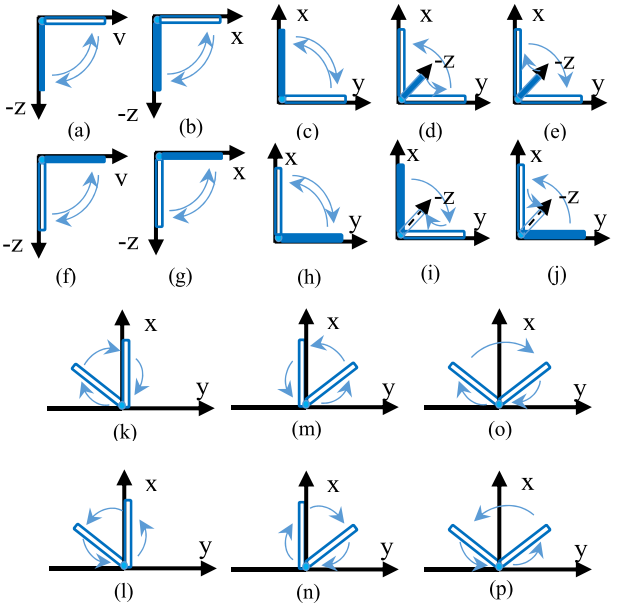


FIGURE 14. Schematic diagram of 16 complex motions, the x-axis direction is the front of the human body, the y-axis direction is the right side of the human body, the z-axis direction points to the ground, and the -z-axis direction points to the sky.

TABLE 4. Correct segmentation rate of complex motions.

	No.1	No.2	No.3	No.4	No.5	Average
CSR (%)	96.88	95.83	100	100	97.92	98.13

A. EXPERIMENT OF SEGMENTAION ALGORITHM

$D(n) = \{d(0), d(1), d(2) \ldots d(n)\}$ includes direction information and position information. WTS uses direction to segment $D(n)$.

In this article, 16 complex motions $M_i(i = 0, 1, \cdots, 15)$ are designed for segmentation experiments as shown in Fig. 14, and each motion is collected 30 times. The data are divided into five groups, and each group contains 96 data. Firstly, the segmentation point P_i and the simple sequence s_i of complex motion $M_i(i = 0, 1, \cdots, 15)$ are determined by manual segmentation. Then, WTS was used to segment $M_i(i = 0, 1, \cdots, 15)$ to obtain P_i' and s_i'. If P_i is the same as P_i' and s_i is the same as s_i', the segmentation performed this time is correct. We obtained the correct segmentation rate (CSR) of five groups of $M_i(i = 0, 1, \cdots, 15)$ and calculated their average values, as shown in Table 4.

B. RECOGNITION OF BASIC MOTIONS

In order to verify the classification performance of the BPNN, we designed a recognition experiment on the basic motions. 10 testers are invited to participate in the collection of data. Each tester repeats each basic motion 6 times, and gets a total of 600 basic motions.

Since the speed and amplitude of each person's motions are different, we have designed two sets of experiments to verify the accuracy of motion recognition: User-Independent (UI)

TABLE 5. The basic motion recognition rate of UI and UD.

	No.1	No.2	No.3	No.4	No.5	No.6	Average
UI (%)	96.67	100	96.67	100	100		98.67
UD (%)	100	100	100	96	100	100	99.33

and User-Dependent (UD) motion recognition. In UI motion recognition, training set and test set are different people's data, and there are not related. In UD motion recognition, training set and test set have an inner relationship, and both of them include everyone's data.

1) UI MOTION RECOGNITION

10 testers were divided into 5 groups, and the experiment was carried out using a five-fold cross validation method. Each time one group was used as the test set, and the other four groups were used as the training set. We conducted five experiments, and each experiment had 480 basic motions in the training set and 120 basic motions in the test set. The recognition rates of the experiment are shown in Table 5.

2) UD MOTION RECOGNITION

Each motion was repeated six times for each tester. We divided 600 basic motions into 6 groups according to the number of repetitions, and each group contained 100 basic motions (ten testers, ten motions, performed once). Then use the cross-validation method to conduct six experiments, each time one group was used as the test set, and the other five groups were used as the training set. The recognition rates of the experiment are shown in Table 5.

C. RECOGNITION OF COMPLEX MOTIONS

In order to verify the classification performance of the BPTS, we designed a recognition experiment for complex motion. Complex motions are decomposed into simple motions by WTS, and the sequence with the time stamp is generated. Then the simple motions are recognized by BPTS. Finally, complex motions are recognized according to the time stamps.

1) RECOGNITION OF SIMPLE MOTIONS

In this experiment, 10 testers are invited to participate in the collection of data. There is a total of 16 complex motions in Fig. 14. Each tester repeats each motion 3 times, and gets a total of 480 complex motions which are randomly divided into 5 groups.

We use WTS to segment complex motion into simple ones. Simple motions are classified into basic motions by manual recognition. Then the BPNN is used to recognize the simple motion. The result of BP classification is compared with that of manual classification. If the results are the same, it means that the BP recognition is correct; if not, it means that the BP recognition is wrong. The correct recognition rate (CRR) is shown in Table 6.

TABLE 6. Recognition rate of simple motions.

	No.1	No.2	No.3	No.4	No.5	Average
CRR (%)	99.36	98.71	98.08	97.44	97.44	98.21

TABLE 7. Correspondence between input and output of BPNN.

Basic Motion	Output Identification	Digital number
Raise your arms sideways (RS)	1000000000	1
Lower your arms sideways (LS)	0100000000	2
Raise your arms forward (RF)	0010000000	3
Lower your arms forward (LF)	0001000000	4
Move your arm to the right (MR)	0000100000	5
Move your arm to the left (ML)	0000010000	6
Raise your arm left front (RLF)	0000001000	7
Lower your arms right rear (LRR)	0000000100	8
Raise your arm right front (RRF)	0000000010	9
Lower your arms left rear (LLR)	0000000001	0

TABLE 8. The numbering sequence after the complex motion is segmented into the basic motion.

Complex motion label	Simple motions	Basic motions	Numbering sequence
M_0	$\{s_1, s_2\}$	$\{RS, LS\}$	$\{1, 2\}$
M_1	$\{s_1, s_2\}$	$\{LS, RS\}$	$\{2, 1\}$
M_2	$\{s_1, s_2\}$	$\{RF, LF\}$	$\{3, 4\}$
M_3	$\{s_1, s_2\}$	$\{LF, RF\}$	$\{4, 3\}$
M_4	$\{s_1, s_2\}$	$\{MR, ML\}$	$\{5, 6\}$
M_5	$\{s_1, s_2\}$	$\{ML, MR\}$	$\{6, 5\}$
M_6	$\{s_1, s_2\}$	$\{RS, ML\}$	$\{1, 6\}$
M_7	$\{s_1, s_2\}$	$\{MR, LS\}$	$\{5, 2\}$
M_8	$\{s_1, s_2\}$	$\{RF, MR\}$	$\{3, 5\}$
M_9	$\{s_1, s_2\}$	$\{ML, LF\}$	$\{6, 4\}$
M_{10}	$\{s_1, s_2, s_3\}$	$\{RLF, MR, LF\}$	$\{7, 5, 4\}$
M_{11}	$\{s_1, s_2, s_3\}$	$\{RF, ML, LRR\}$	$\{3, 6, 8\}$
M_{12}	$\{s_1, s_2, s_3\}$	$\{RRF, ML, LF\}$	$\{9, 6, 4\}$
M_{13}	$\{s_1, s_2, s_3\}$	$\{RF, MR, LLR\}$	$\{3, 5, 10\}$
M_{14}	$\{s_1, s_2, s_3\}$	$\{RLF, MR, LLR\}$	$\{7, 5, 10\}$
M_{15}	$\{s_1, s_2, s_3\}$	$\{RRF, ML, LRR\}$	$\{9, 6, 8\}$

2) COMBINED OUTPUT OF BASIC MOTIONS

In order to observe the classification results of complex motions more conveniently and intuitively, we make a digital number for each basic motion, as shown in Table 7.

The 16 complex motions M_i in Figure 12 are decomposed into simple motions s_i by WTS, and the time stamp of them are generated. Then according to the digital number in Table 7, the basic motions $\{b_i\}$ of M_i are obtained as shown in Table 8.

D. RESULT ANALYSIS

As shown in Table 5, the recognition rates of UI and UD are both over 98.67%, which shows that our recognition method is user-independent. The correct segmentation rate of WTS can reach 98.13% as shown in Table 4. The correct recognition rate of the simple motion is 98.21% as shown in Table 6. Therefore, the total recognition rate of complex motions is 96.37%.

TABLE 9. Correct segmentation rate of different segmentation algorithms.

Algorithm	MAX	AVERAGE
WTS	99.32%	98.16%
SSC	72.50%	56.77%
LRR	74.30%	61.11%
LRLRR	92.00%	69.78%
RS	83.90%	75.96%
FS	93.60%	85.09%
SSR	92.10%	83.51%
AutoPlait	—	97.00%

TABLE 10. The configuration of some parameters or structures.

Algorithm	Parameter or structure configuration
ATC	The duration of the time window is 100 ms; The threshold voltage is 1.65V.
DF	The size of window is 150; The size of stride is 50; The number of gesture classes is 16; The number of trees in scanning RFs is 50; The number of trees in cascade RFs is 101.
DCN	9 convolutional layers, 4 pooling layers, 3 fully connected layers; The size of first convolutional layer is 5x5 and the stride is 3, other convolutional layers is 3x3 and 1; The stride value of pooling layers is 2.
DTW	Assign 8 neighborhood coding values according to direction.
PCA+SVM	The energy loss rate of PCA is 0.01; The kernel function of SVM is Gaussian kernel, and $\sigma = 10^{-0.44}$, γ=1.
WPT+UKFNN	The scale is 3 and the node is 8; The dmey wavelet is used to extract feature values; The dmey wavelet to extract feature values is 2; The sliding step size is 4.
HBU-LSTM	The first layer of the model is Bi-LSTM layer; The last layer of the model is ULSTM layer; Two dropout layers; The activation function in output layer is Softmax.

We compare the segmentation rate of WTS with that of other algorithms, such as Sparse subspace clustering (SSC) [71], Low rank representation (LRR) [72], Laplacian regularized low-rank representation (LRLRR) [73], Rough segmentation (RS) and Fine segmentation (FS) [74], Supervised Time Series Segmentation and State Recognition(SSR) [75], and AutoPlait [76], and the result is shown in Table 9. The value of MAX refers to the average of the best results in multiple experiments. The value of AVERAGE refers to the average of multiple sets of experimental data. The stability of the algorithm can be judged according to the difference between the two data. As shown in the Table 9, WTS has higher segmentation rate and more stable segmentation effect.

We compare the recognition rate of BPTS with that of other algorithms, such as Average Threshold Crossing (ATC) [77], CNN [78], Deep Forest algorithm (DF) [79], Deep Convolutional Network (DCN) [80], Dynamic Time Warping (DTW) [81], PCA [82], Edge Oriented Histogram (EOH) and SVM [83], WPT and unscented Kalman neural network (UKFNN) [84], Hybrid Bidirectional Unidirectional Long Short-Term Memory (HBU-LSTM) [85], and PCA and SVM [13]. Some parameters or structures of other algorithms are configured as shown in Table 10. The recognition accuracy of them are shown in Table 11. The recognition rate of BPTS reached 96.37%, which is the highest recognition accuracy.

TABLE 11. Performance comparison of several different recognition algorithms.

Algorithm	Number of motions	Recognition accuracy
BPTS	Basic 10	99.00%
	Complex 16	96.37%
ATC	4	92.87%
CNN	25	96.20%
DF	16	96.00%
DCN	Basic 7	97.10%
	Complex 7	85.30%
DTW	24	90.00%
PCA	4	91.25%
EOH+SVM	24	93.75%
WPT+UKFNN	6	94.83%
HBU-LSTM	two benchmark datasets	90%
PCA+SVM	4	96%

VI. CONCLUSION

In this article, we propose an algorithm to determine the number of clusters and a method to identify complex motions. Firstly, SSED based on SOM is proposed to determine the optimal number of clusters, and it is not limited by the number of clusters. The basic motion set is obtained by SOM clustering, and the eigenvalue is used to map basic motions to distinct activities.

Secondly, WTS is proposed to segment complex motions into simple ones, and simple motion sequence is generated. WTS maintains the same shape and characteristics as the original sequence. Compared with other segmentation algorithms, WTS has a good segmentation rate.

Finally, BPTS is conducted to classify the simple motions according to the basic motion set. And the complex motion is recognized according to the time stamp.

In this article, the motions of human upper limbs are used as test samples to verify the effectiveness of this method. SOM combined with SSED determines that the number of clusters of human upper limb motion is 10. The experimental results show that WTS is effective, and the correct segmentation rate is greater than 98.13%. Experiments on UI motion and UD motion verify the effectiveness of our algorithm, and performance comparison of several different recognition algorithms were conducted. Compared with other recognition algorithms, our method shows the highest recognition rate of 96.37%.

REFERENCES

[1] D. Carter-Davies, J. Chen, F. Chen, M. Li, and C. Yang, "Mechatronic design and control of a 3D printed low cost robotic upper limb," in *Proc. 11th Int. Workshop Human Friendly Robot. (HFR)*, Shenzhen, China, Nov. 2018, pp. 1–6.

[2] C. Cortés, A. Ardanza, F. Molina-Rueda, A. Cuesta-Gómez, L. Unzueta, G. Epelde, O. E. Ruiz, A. De Mauro, and J. Florez, "Upper limb posture estimation in robotic and virtual reality-based rehabilitation," *BioMed Res. Int.*, vol. 2014, pp. 1–18, Jul. 2014, doi: 10.1155/2014/821908.

[3] F. Scotto di Luzio, C. Lauretti, F. Cordella, F. Draicchio, and L. Zollo, "Visual vs vibrotactile feedback for posture assessment during upper-limb robot-aided rehabilitation," *Appl. Ergonom.*, vol. 82, Jan. 2020, Art. no. 102950, doi: 10.1016/j.apergo.2019.102950.
[4] A. Tewari, B. Taetz, F. Grandidier, and D. Stricker, "A probabilistic combination of CNN and RNN estimates for hand gesture based interaction in car," in *Proc. IEEE Int. Symp. Mixed Augment. Real. (ISMAR-Adjunct)*, Nantes, France, Oct. 2017, pp. 1–6.
[5] T. Lu, "A motion control method of intelligent wheelchair based on hand gesture recognition," in *Proc. IEEE 8th Conf. Ind. Electron. Appl. (ICIEA)*, Melbourne, VIC, Australia, Jun. 2013, pp. 957–962.
[6] Z. Gao, P. Wang, H. Wang, M. Xu, and W. Li, "A review of dynamic maps for 3D human motion recognition using ConvNets and its improvement," *Neural Process. Lett.*, vol. 52, no. 2, pp. 1501–1515, Oct. 2020, doi: 10.1007/s11063-020-10320-w.
[7] H.-H. Pham, L. Khoudour, A. Crouzil, P. Zegers, and S. A. Velastin, "Exploiting deep residual networks for human action recognition from skeletal data," *Comput. Vis. Image Understand.*, vol. 170, pp. 51–66, May 2018, doi: 10.1016/j.cviu.2018.03.003.
[8] H. B. Naeem, F. Murtaza, M. H. Yousaf, and S. A. Velastin, "Multiple batches of motion history images (MB-MHIs) for multi-view human action recognition," *Arabian J. Sci. Eng.*, vol. 45, no. 8, pp. 6109–6124, Mar. 2020, doi: 10.1007/s13369-020-04481-y.
[9] H. Wang, B. Yu, K. Xia, J. Li, and X. Zuo, "Skeleton edge motion networks for human action recognition," *Neurocomputing*, vol. 423, pp. 1–12, Jan. 2021, doi: 10.1016/j.neucom.2020.10.037.
[10] K. Zhang and W. Ling, "Joint motion information extraction and human behavior recognition in video based on deep learning," *IEEE Sensors J.*, vol. 20, no. 20, pp. 11919–11926, Oct. 2020, doi: 10.1109/JSEN.2019.2959582.
[11] B. Li, B. Bai, and C. Han, "Upper body motion recognition based on key frame and random forest regression," *Multimedia Tools Appl.*, vol. 79, nos. 7–8, pp. 5197–5212, Jul. 2018, doi: 10.1007/s11042-018-6357-y.
[12] M. Emperuman and S. Chandrasekaran, "Hybrid continuous density HMM-based ensemble neural networks for sensor fault detection and classification in wireless sensor network," *Sensors*, vol. 20, no. 3, p. 745, Jan. 2020, doi: 10.3390/s20030745.
[13] L. Zhao and W. Chen, "Detection and recognition of human body posture in motion based on sensor technology," *IEEJ Trans. Electr. Electron. Eng.*, vol. 15, no. 5, pp. 766–770, May 2020, doi: 10.1002/tee.23113.
[14] T. Zhang, H. Wang, J. Chen, and E. He, "Detecting unfavorable driving states in electroencephalography based on a PCA sample entropy feature and multiple classification algorithms," *Entropy*, vol. 22, no. 11, p. 1248, Nov. 2020, doi: 10.3390/e22111248.
[15] Z. Geler, V. Kurbalija, M. Ivanović, and M. Radovanović, "Weighted kNN and constrained elastic distances for time-series classification," *Expert Syst. Appl.*, vol. 162, Dec. 2020, Art. no. 113829, doi: 10.1016/j.eswa.2020.113829.
[16] J. Shao, Y. Niu, C. Xue, Q. Wu, X. Zhou, Y. Xie, and X. Zhao, "Single-channel SEMG using wavelet deep belief networks for upper limb motion recognition," *Int. J. Ind. Ergonom.*, vol. 76, Mar. 2020, Art. no. 102905, doi: 10.1016/j.ergon.2019.102905.
[17] Y. Sun and H. Bozdogan, "Segmentation of high dimensional time-series data using mixture of sparse principal component regression model with information complexity," *Entropy*, vol. 22, no. 10, p. 1170, Oct. 2020, doi: 10.3390/e22101170.
[18] J. Bian, X. Mei, Y. Xue, L. Wu, and Y. Ding, "Efficient hierarchical temporal segmentation method for facial expressionsequences," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, pp. 1680–1695, May 2019, doi: 10.3906/elk-1809-75.
[19] G. Feng and W. Hou, "Segmentation and recognition of continuous gesture based on chaotic theory," *Behaviour Inf. Technol.*, vol. 39, no. 11, pp. 1246–1256, Nov. 2020, doi: 10.1080/0144929X.2019.1661519.
[20] V. S. Siyou Fotso, E. Mephu Nguifo, and P. Vaslin, "Grasp heuristic for time series compression with piecewise aggregate approximation," *RAIRO Oper. Res.*, vol. 53, no. 1, pp. 243–259, Jan. 2019, doi: 10.1051/ro/2018089.
[21] H. Ren, X. Liao, Z. Li, and A. AI-Ahmari, "Anomaly detection using piecewise aggregate approximation in the amplitude domain," *Appl. Intell.*, vol. 48, no. 5, pp. 1097–1110, May 2018, doi: 10.1007/s10489-017-1017-x.
[22] J. Zhang and Z. M. Ma, "An algorithm of finding similar subpattern based on time series," *Comput. Technol. Develop.*, vol. 16, no. 1, pp. 140–142 and 146, 2016, doi: 10.3969/j.issn.1673-629X.2006.01.045.
[23] D. Z. Zhou and M. Q. Li, "Time series segmentation based on series importance point," *Comput. Eng.*, vol. 34, no. 23, p. 14, 2008, doi: 10.3969/j.issn.1000-3428.2008.23.006.
[24] W. Y. Sun and X. L. Su, *Summary of Time Series Segmentation Metheds*. Beijing, China: Sciencepaper Online, 2012.
[25] S. Madan and K. J. Dana, "Modified balanced iterative reducing and clustering using hierarchies (m-BIRCH) for visual clustering," *Pattern Anal. Appl.*, vol. 19, no. 4, pp. 1023–1040, Apr. 2015, doi: 10.1007/s10044-015-0472-4.
[26] P. K. Prasad and C. P. Rangan, *Privacy Preserving BIRCH algorithm for Clustering Over Vertically Partitioned Databases* (Lecture Notes in Computer Science). Seoul, South Korea, 2006, pp. 84–89.
[27] A. Prasanth and S. Valsala, "Semantic chameleon clustering analysis algorithm with recommendation rules for efficient Web usage mining," in *Proc. IEEE-GCC Conf. Exhib. (GCCCE)*, Manama, Bahrain, Feb. 2017, pp. 1–9.
[28] D. Saravanan, "CURE clustering technique suitable for video data retrieval," in *Proc. IEEE Int. Conf. Comput. Intell. Comput. Res. (ICCIC)*, Chennai, India, Dec. 2016, pp. 306–309.
[29] W. Cai, J. Zhao, and M. Zhu, "A real time methodology of cluster-system theory-based reliability estimation using k-means clustering," *Rel. Eng. Syst. Saf.*, vol. 202, Oct. 2020, Art. no. 107045, doi: 10.1016/j.ress.2020.107045.
[30] W. Chang, X. Ji, Y. Liu, Y. Xiao, B. Chen, H. Liu, and S. Zhou, "Analysis of University students' behavior based on a fusion K-means clustering algorithm," *Appl. Sci.*, vol. 10, no. 18, p. 6566, Sep. 2020, doi: 10.3390/app10186566.
[31] Z. Li, G. Wang, and G. He, "Milling tool wear state recognition based on partitioning around medoids (PAM) clustering," *Int. J. Adv. Manuf. Technol.*, vol. 88, nos. 5–8, pp. 1203–1213, May 2016, doi: 10.1007/s00170-016-8848-1.
[32] N. Verma and N. Baliyan, "PAM clustering based taxi hotspot detection for informed driving," in *Proc. 8th ICCCNT*, Delhi, India, 2017, pp. 1–7.
[33] M. H. Rad and M. Abdolrazzagh-Nezhad, "Data cube clustering with improved DBSCAN based on fuzzy logic and genetic algorithm," *Inf. Technol. Control*, vol. 49, no. 1, pp. 127–143, Mar. 2020, doi: 10.5755/j01.itc.49.1.23780.
[34] R. Scitovski and K. Sabo, "DBSCAN-like clustering method for various data densities," *Pattern Anal. Appl.*, vol. 23, no. 2, pp. 541–554, Apr. 2019, doi: 10.1007/s10044-019-00809-z.
[35] A. Idrissi, H. Rehioui, A. Laghrissi, and S. Retal, "An improvement of DENCLUE algorithm for the data clustering," in *Proc. 5th ICTA*, Marrakech, Morocco, 2015, pp. 1–6.
[36] X. G. Yu and Y. Jian, "A new clustering algorithm based on KNN and DENCLUE," in *Proc. 4th Int. Conf. Mach. Learn. Cybern.*, Canton, China, 2005, pp. 2033–2038.
[37] V. Bureva, E. Sotirova, S. Popov, D. Mavrov, and V. Traneva, *Generalized Net of Cluster Analysis Process Using STING: A Statistical Information Grid Approach to Spatial Data Mining* (Lecture Notes in Computer Science), vol. 10333. May 2017, pp. 239–248, doi: 10.1007/978-3-319-59692-1_21.
[38] C. Dai, J. Wu, D. Pi, S. I. Becker, and B. Johnson, "Brain EEG time-series clustering using maximum-weight clique," *IEEE Trans. Cybern.*, early access, Mar. 9, 2020, doi: 10.1109/TCYB.2020.2974776.
[39] L. C. Zhao, Z. W. Shang, J. Tan, X. L. Luo, T. P. Zhang, Y. Wei, and Y. Y. Tang, "Adaptive parameter estimation of GMM and its application in clustering," *Future Gener. Comput. Syst.*, vol. 106, pp. 250–259, Jan. 2020, doi: 10.1016/j.future.2020.01.012.
[40] V. Athira and R. Sindhu, "GMM cluster based choroid segmentation in EDI OCT images," in *Proc. Int. Conf. Comput. Methodol. Commun. (ICCMC)*, Erode, India, 2017, pp. 727–731.
[41] J. A. Webb, N. R. Bond, S. R. Wealands, R. M. Nally, G. P. Quinn, P. A. Vesk, and M. R. Grace, "Bayesian clustering with AutoClass explicitly recognises uncertainties in landscape classification," *Ecography*, vol. 30, no. 4, pp. 526–536, May 2007, doi: 10.1111/j.0906-7590.2007.05002.x.
[42] C. Pizzuti and D. Talia, "P-AutoClass: Scalable parallel clustering for mining large data sets," *IEEE Trans. Knowl. Data Eng.*, vol. 15, no. 3, pp. 629–641, Jun. 2003, doi: 10.1109/TKDE.2003.1198395.
[43] H. R. Medeiros, F. D. B. de Oliveira, H. F. Bassani, and A. F. R. Araujo, "Dynamic topology and relevance learning SOM-based algorithm for image clustering tasks," *Comput. Vis. Image Understand.*, vol. 179, pp. 19–30, Feb. 2019, doi: 10.1016/j.cviu.2018.11.003.

[44] A. Bustamam, M. A. Rivai, and T. Siswantining, "Implementation of spectral clustering on microarray data of carcinoma using self organizing map (SOM)," in *Proc. AIP Conf.*, 2018, Art. no. 20240, doi: 10.1063/1.5064237.

[45] X. D. Zhao and Z. Qi, "Study on cluster analysis on the basis of SOM neural network," *J. Jilin Province Econ. Manage. Cadre College*, no. 2, pp. 81–84, Apr. 2008.

[46] K. El Hindi, H. AlSalman, S. Qasem, and S. Al Ahmadi, "Building an ensemble of fine-tuned Naive Bayesian classifiers for text classification," *Entropy*, vol. 20, no. 11, p. 857, Nov. 2018, doi: 10.3390/e20110857.

[47] Y. Ding, J. Ma, and Y. Tian, "Health assessment and fault classification for hydraulic pump based on LR and softmax regression," *J. Vibroeng.*, vol. 17, no. 4, pp. 1805–1816, Jun. 2015.

[48] H. Huang, H. Wang, and M. Sun, "Incomplete data classification with view-based decision tree," *Appl. Soft Comput.*, vol. 94, Sep. 2020, Art. no. 106437, doi: 10.1016/j.asoc.2020.106437.

[49] K. Wang, L. Cheng, and B. Yong, "Spectral-similarity-based kernel of SVM for hyperspectral image classification," *Remote Sens.*, vol. 12, no. 13, p. 2154, Jul. 2020, doi: 10.3390/rs12132154.

[50] Z. Jiang, Z. Bian, and S. Wang, "Multi-view local linear KNN classification: Theoretical and experimental studies on image classification," *Int. J. Mach. Learn. Cybern.*, vol. 11, no. 3, pp. 525–543, Mar. 2020, doi: 10.1007/s13042-019-00992-9.

[51] H. Jia, S. Wang, D. Zheng, X. Qu, and S. Fan, "Comparative study of motor imagery classification based on BP-NN and SVM," *J. Eng.*, vol. 2019, no. 23, pp. 8646–8649, Dec. 2019, doi: 10.1049/joe.2018.9075.

[52] H. Tang, J. He, Y. Zheng, J. Zhang, and L. Wei, "Video-based detection and classification of driving postures by feature distance extraction and BP neutral network," *Proc. SPIE*, vol. 11198, Jul. 2019, Art. no. 111980G, doi: 10.1117/12.2540471.

[53] S. Liu, W. Jiang, L. Wu, H. Wen, M. Liu, and Y. Wang, "Real-time classification of rubber wood boards using an SSR-based CNN," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 11, pp. 8725–8734, Nov. 2020, doi: 10.1109/TIM.2020.3001370.

[54] M. E. Valle, "Reduced dilation-erosion perceptron for binary classification," *Mathematics*, vol. 8, no. 4, p. 512, Apr. 2020, doi: 10.3390/math8040512.

[55] Y. Kabashima and S. Uda, "A BP-based algorithm for performing Bayesian inference in large perceptron-type networks," *Lect. Notes Comput. Sci.*, vol. 3244, no. 1, pp. 479–493, 2004, doi: 10.1007/978-3-540-30215-5_36.

[56] D. Perdios, M. Vonlanthen, F. Martinez, M. Arditi, and J.-P. Thiran, "CNN-based ultrasound image reconstruction for ultrafast displacement tracking," *IEEE Trans. Med. Imag.*, early access, Dec. 22, 2020, doi: 10.1109/TMI.2020.3046700.

[57] K. Xie, H. Yi, G. Hu, L. Li, and Z. Fan, "Short-term power load forecasting based on elman neural network with particle swarm optimization," *Neurocomputing*, vol. 416, pp. 136–142, Nov. 2020, doi: 10.1016/j.neucom.2019.02.063.

[58] MA. Mansor, MSM. Kasihmuddin, and S. Sathasivam, "Modified lion optimization algorithm with discrete hopfield neural network for higher order Boolean satisfiability programming," in *Proc. 2nd Int. Conf. Appl. Ind. Math. Statist.*, Kuantan, Malaysia, 2019, pp. 47–61.

[59] N. Jiang and T. Liu, "An improved speech segmentation and clustering algorithm based on SOM and K-Means," *Math. Problems Eng.*, vol. 2020, pp. 1–19, Sep. 2020, doi: 10.1155/2020/3608286.

[60] F. Nan, Y. Li, X. Jia, L. Dong, and Y. Chen, "Application of improved SOM network in gene data cluster analysis," *Meas. J. Int. Meas. Confed*, vol. 145, pp. 370–378, Oct. 2019, doi: 10.1016/j.measurement.2019.01.013.

[61] Y. Cao, Y. L. Wang, H. M. He, B. S. Yang, and W. H. Gui, "Intelligent scheduling in pre-burdening of iron ore: Canopy-Kmeans clustering algorithm and combinatorial optimization Canopy-Kmeans," *Control Theory Appl.*, vol. 34, no. 7, pp. 947–955, Jul. 2017.

[62] E. F. Sirat, B. D. Setiawan, and F. Ramdani, "Comparative analysis of K-Means and isodata algorithms for clustering of fire point data in sumatra region," in *Proc. 4th Int. Symp. Geoinform. (ISyG)*, Malang, Indonesia, Nov. 2018, pp. 1–6.

[63] M. Charrad, N. Ghazzali, V. Boiteau, and A. Niknafs, "NbClust: An R package for determining the relevant number of clusters in a data set," *J. Stat. Softw.*, vol. 61, pp. 1-36, Oct. 2014, doi: 10.18637/jss.v061.i06.

[64] S. Chatterjee and M. M. Mohanty, "Automatic cluster selection using gap statistics for pattern-based multi-point geostatistical simulation," *Arab. J. Geosci.*, vol. 8, no. 9, pp. 7691–7704, Dec. 2014, doi: 10.1007/s12517-014-1724-0.

[65] M. Yan and K. Ye, "Determining the number of clusters using the weighted gap statistic," *Biometrics*, vol. 63, no. 4, pp. 1031–1037, Dec. 2007, doi: 10.2307/4541456.

[66] T. S. Açar and N. A. Öz, "The determination of optimal cluster number by silhouette index at clustering of the European union member countries and candidate turkey by waste indicators," *Pamukkale Univ. J. Eng. Sci.*, vol. 26, no. 3, pp. 481–487, 2020, doi: 10.5505/pajes.2019.49932.

[67] A. Lensen, X. Bing, and M. Zhang, "Using particle swarm optimisation and the silhouette metric to estimate the number of clusters, select features, and perform clustering," in *Applications of Evolutionary Computation* (Lecture Notes in Computer Science), vol. 10199. Mar. 2017, pp. 538–554, doi: 10.1007/978-3-319-55849-3_35.

[68] A. M. Rajee and F. S. Francis, "A study on outlier distance and SSE with multidimensional datasets in K-means clustering," in *Proc. Int. Conf. Adv. Comput. (ICoAC)*, Chennai, India, 2013, pp. 33–36.

[69] R. Xie and J. Cao, "Accelerometer-based hand gesture recognition by neural network and similarity matching," *IEEE Sensors J.*, vol. 16, no. 11, pp. 4537–4545, Jun. 2016, doi: 10.1109/JSEN.2016.2546942.

[70] J.-S. Wang and F.-C. Chuang, "An accelerometer-based digital pen with a trajectory recognition algorithm for handwritten digit and gesture recognition," *IEEE Trans. Ind. Electron.*, vol. 59, no. 7, pp. 2998–3007, Jul. 2012, doi: 10.1109/TIE.2011.2167895.

[71] E. Elhamifar and R. Vidal, "Sparse subspace clustering," in *Proc. IEEE Conf. CVPR*, Miami, FL, USA, Jun. 2009, pp. 20–25.

[72] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 1, pp. 171–184, Jan. 2013, doi: 10.1109/TPAMI.2012.88.

[73] M. Yin, J. Gao, and Z. Lin, "Laplacian regularized low-rank representation and its applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 3, pp. 504–517, Mar. 2016, doi: 10.1109/TPAMI.2015.2462360.

[74] J. Bian, X. Mei, Y. Xue, L. Wu, and Y. Ding, "Efficient hierarchical temporal segmentation method for facial expressionsequences," *Turkish J. Electr. Eng. Comput. Sci.*, vol. 27, no. 3, pp. 1680–1695, May 2019, doi: 10.3906/elk-1809-75.

[75] M. Y. Shi, P. Wang, and W. Wang, "Algorithm of supervised time series segmentation and state recognition," *Comput. Eng.*, vol. 46, no. 5, pp. 131–138, May 2020, doi: 10.19678/j.issn.1000-3428.0056243.

[76] Y. Matsubara, Y. Sakurai, and C. Faloutsos, "AutoPlait: Automatic mining of co-evolving time sequences," in *Proc. ACM SIGMOD Int. Conf. Manage. Data*, Snowbird, UT, USA, Jun. 2014, pp. 193–204.

[77] S. Sapienza, P. M. Ros, D. A. F. Guzman, F. Rossi, R. Terracciano, E. Cordedda, and D. Demarchi, "On-line event-driven hand gesture recognition based on surface electromyographic signals," in *Proc. IEEE Int. Symp. Circuits Syst.*, Florence, Italy, 2018, pp. 1–5.

[78] C. J. L. Flores, A. E. G. Cutipa, and R. L. Enciso, "Application of convolutional neural networks for static hand gestures recognition under different invariant features," in *Proc. IEEE Int. Conf. Electron., Electr. Eng. Comput. (INTERCON)*, Cusco, Peru, Aug. 2017, pp. 1–4.

[79] J. Zhao, J. Mao, G. Wang, H. Yang, and B. Zhao, "A miniaturized wearable wireless hand gesture recognition system employing deep-forest classifier," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Turin, Italy, Oct. 2017, pp. 1–4.

[80] P. Bao, A. I. Maqueda, C. R. del-Blanco, and N. García, "Tiny hand gesture recognition without localization via a deep convolutional network," *IEEE Trans. Consum. Electron.*, vol. 63, no. 3, pp. 251–257, Aug. 2017, doi: 10.1109/TCE.2017.014971.

[81] W. Ahmed, K. Chanda, and S. Mitra, "Vision based hand gesture recognition using dynamic time warping for Indian sign language," in *Proc. Int. Conf. Inf. Sci. (ICIS)*, Kochi, India, 2016, pp. 120–125.

[82] M. K. Ahuja and A. Singh, "Static vision based Hand Gesture recognition using principal component analysis," in *Proc. IEEE Int. Conf. MOOCs, Innov. Technol. Educ. (MITE)*, Amritsar, India, Oct. 2015, pp. 402–406.

[83] S. Nagarajan and T. S. Subashini, "Static hand gesture recognition for sign language alphabets using edge oriented histogram and multi class SVM," *Int. J. Comput. Appl.*, vol. 82, no. 4, pp. 28–35, Nov. 2013, doi: 10.5120/14106-2145.

[84] X. Shi, P. Qin, J. Zhu, S. Xu, and W. Shi, "Lower limb motion recognition method based on improved wavelet packet transform and unscented Kalman neural network," *Math. Problems Eng.*, vol. 2020, pp. 1–16, Apr. 2020, doi: 10.1155/2020/5684812.

[85] S. Ameur, A. B. Khalifa, and M. S. Bouhlel, "A novel hybrid bidirectional unidirectional LSTM network for dynamic hand gesture recognition with leap motion," *Entertainment Comput.*, vol. 35, Aug. 2020, Art. no. 100373, doi: 10.1016/j.entcom.2020.100373.

WENKANG YANG received the B.S. degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree in circuits and systems. His research interests include motion recognition and robot control.

QIAOLING DU received the Ph.D. degree in engineering from Jilin University, Changchun, China, in 2008. Since 2003, she has been serving as a Teacher with the Department of Electronic Information Engineering, College of Electronic Science and Engineering, Jilin University, where she has been an Assistant Professor since 2011. Her research interests include wireless sensor network positioning technology, robot path planning and navigation, and development of real-time water quality monitoring systems. She is a member of the Chinese Institute of Electronics.

JIANCHAO CUI received the B.S. degree from the School of Physics and Electronics Information, Inner Mongolia University for Nationalities, Tongliao, China, in 2018. He is currently pursuing the master's degree in integrated circuit engineering with the College of Electronic Science and Engineering, Jilin University. His research is on the posture recognition of humans and robots.

YANKAI WANG received the B.S. degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2019, where he is currently pursuing the master's degree in integrated circuit engineering. His research is on the motion control algorithm of climbing robot.

XINPO LU received the B.S. degree from the School of Physics and Electronic Engineering, Northeast Petroleum University, Daqing, China, in 2019. He is currently pursuing the master's degree in circuits and systems with the College of Electronic Science and Engineering, Jilin University. His research is on the stable posture adjustment algorithm of bionic robot.

CHUNXIAO QI received the B.S. degree from the College of Electronic Information Engineering, Changchun University of Science and Technology, Changchun, China, in 2018. She is currently pursuing the master's degree in integrated circuit engineering with the College of Electronic Science and Engineering, Jilin University. Her research is on the control systems for dexterous hands.

GUIPING ZHANG received the B.S. degree from the College of Electronic Science and Engineering, Jilin University, Changchun, China, in 2018, where he is currently pursuing the master's degree in integrated circuit engineering. His research is on the detection of chemical oxygen demand in surface water.