

Research Article

Data Feature Extraction Method of Wearable Sensor Based on Convolutional Neural Network

Baoying Wang 

College of Electronics and Internet of Things, Chongqing College of Electronic Engineering, Chongqing 401331, China

Correspondence should be addressed to Baoying Wang; wangbaoying@cqcet.edu.cn

Received 17 November 2021; Revised 2 December 2021; Accepted 16 December 2021; Published 25 January 2022

Academic Editor: Bhagyaveni M.A

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

With the rapid development of society and science technology, human health issues have attracted much attention due to wearable devices' ability to provide high-quality sports, health, and activity monitoring services. This paper proposes a method for feature extraction of wearable sensor data based on a convolutional neural network (CNN). First, it uses the Kalman filter to fuse the data to obtain a preliminary state estimation, and then it uses CNN to recognize human behavior, thereby obtaining the corresponding behavior set. Moreover, this paper conducts experiments on 5 datasets. The experimental results show that the method in this paper extracts data features at multiple scales while fully maintaining data independence, can effectively extract corresponding feature data, and has strong generalization ability, which can adapt to different learning tasks.

1. Introduction

Wearable devices are smart devices that can be installed on people, animals, and objects, and can sense, transmit, and process information [1]. Sensors are the core carrier, and wearable sensors are an extension of human senses. With the rapid development of the Internet of everything and society, human health issues have also received widespread attention. Various health indicators and parameters of the human body can also be connected to the Internet for monitoring and analysis. Perceive human behavior through wearable sensors, which can be used in medical health, sports fitness, disease prevention, and treatment, especially in the daily monitoring of elderly people [2–8]. With the intelligent development of wearable devices, sensors such as accelerometers, magnetometers, and heart rate meters are built into smartphones and smart bracelets [9]. The sensors can continuously track and collect human behavior data, fully protecting personal privacy as well. Users only need to wear their mobile phones or smart bracelet to complete data collection.

Most of the data collected by wearable sensors are sequence data. Currently, models based on recurrent neural networks (RNN) and convolutional neural networks (CNN)

are mainly used to predict and identify sequence data [10, 11]. Although the accuracy of the RNN-based data prediction and recognition model is slightly higher than the accuracy of the CNN-based model, the cyclic nature and data-dependent characteristics of the RNN model make it difficult to achieve high parallelization on the edge side, which leads to low computing efficiency on the edge hardware platform. CNN can achieve higher computational parallelism and higher applicability. Therefore, this paper uses a CNN-based model to extract the data features of wearable sensors and perform human behavior recognition [12, 13].

2. Related Work

Figure 1 shows the process of feature extraction of wearable device sensing data and the output results based on CNN. First, the human body's acceleration, heart rate, and other data are collected through the sensor in the wearable device, and the original data is scanned with a fixed time window to obtain samples as the input of the CNN model. Then, the model is used to identify its specific behavior category. After obtaining each behavior category, an individual behavior dataset is generated.

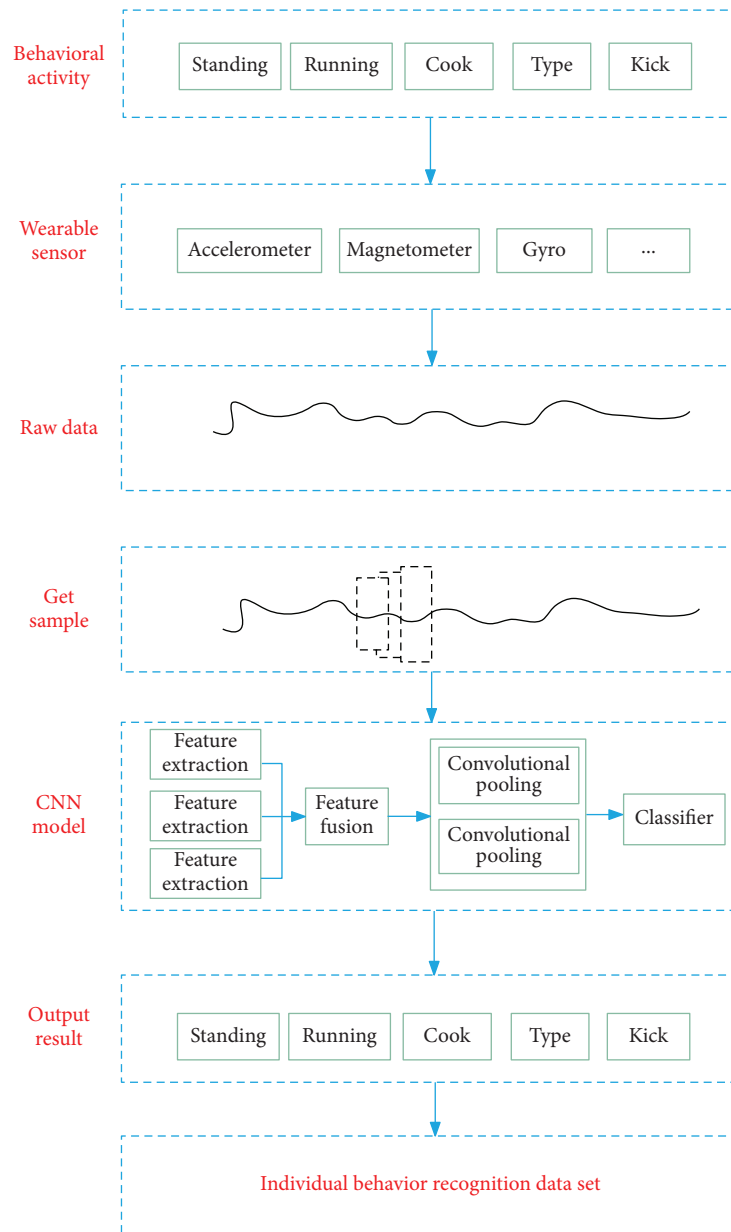


FIGURE 1: Data feature extraction process based on wearable sensors.

2.1. Wearable Sensors and Data Collection. Wearable devices first appeared in the middle of the 20th century. The traditional method is to place sensors directly on the user to collect user motion data. The sensors used in data acquisition equipment mainly include accelerometers, gyroscopes, magnetometers, smartphones, barometers, pressure sensors, and EMG signal detectors, among which accelerometers and gyroscopes are more common. The accelerometer mainly measures the force of the device and can sense acceleration in any direction; the gyroscope judges the current movement state of the device through its rotation state (angle and angular velocity). With the development of smart devices, researchers generally design embed sensors in mobile devices. Chacon et al. designed a new pulse oximeter by embedding sensors in portable medical equipment,

providing real-time monitoring of the physical condition of the elderly [14]. For the feature extraction of wearable sensor data, complex behaviors in general life can be expressed through various basic action combinations. For example, eating snacks includes several basic actions such as holding and eating.

Gu et al. introduced the emerging pattern to describe the discriminative pattern of significant changes between data categories and identified sequential, interlaced, and concurrent activities in a unified framework, thereby identifying sensor features and classifying activities [15]. Wang et al. proposed a hierarchical recognition method to infer ongoing activities in a multistage process to better distinguish similar activities and improve overall performance [16]. In terms of complex behavior recognition,

Chang et al. considered the interaction between people and the surrounding environment using sensors distributed on people and objects to collect behavioral data, which improves the extraction of data features of complex behaviors [17].

2.2. Data Feature Extraction. In human behavior recognition, the most representative algorithm is the improved dense trajectories (IDT) algorithm [11]. It is a dense optical flow trajectory algorithm based on dense trajectories (DT). Before CNN was applied, IDT was the best algorithm for human behavior recognition. With the development of CNN, Ye et al. proposed the two-stream method to extract behavioral features in video space and time. Compared with IDT, the two-stream method is more robust [18]. The convolutional 3D method proposed by Torfi et al. can directly process video [19]. The effect is very good in spatiotemporal feature extraction. This method effectively improves the recognition speed, but the accuracy is low. Jin et al. put forward the time-sensitive network (TSN) method [20]. By segmenting the long video, the problem of the inability to model the long-term structure in the two-stream method is solved, but the TSN method has a complex structure. The recognition speed is slow. Wu et al. discussed the deep local video feature (DLVF) method based on the TSN method, assigning different weights to different segments in TSN and fusing the judgment results of different segments more reasonably [21]. Chanti et al. envisioned a behavior recognition method based on temporal and spatial weighted gesture motion characteristics through dynamic time warping and Fourier time pyramid algorithm modeling, and finally improved the accuracy of human behavior recognition [22].

3. Wearable Sensor Data Feature Extraction Method Based on CNN

The data feature extraction studied in this paper is suitable for sensors installed on people, animals, and objects, but the sensors must move with the appendages so that the relevant data features can be extracted, and their behaviors can be analyzed to obtain a behavior set. For example, when a person is running, their running behavior is recognized through the data characteristics of parameters such as legs, hands, and heart rate.

For the extraction of wearable sensor data features, the acceleration data, angular velocity data, and magnetic field data output by the wearable sensor are first fused by the extended Kalman filter fusion algorithm. Then, the preliminary state estimation of the human behavior represented by the wearable sensor output data is obtained, and the CNN model is used to process the collected data in parallel to get the result of human behavior recognition.

3.1. Data Fusion. At present, the commonly used human behavior data feature decomposition methods are estimation methods based on Kalman filtering, which mainly include unscented Kalman filtering, extended Kalman filtering, and volume Kalman filtering [23]. The estimation

methods based on non-Kalman filtering include the complementary filtering algorithm and the gradient descent algorithm. Through the comparison of several methods, this paper selects the extended Kalman filter method to identify the behavior of the sensor.

The Kalman filter is an algorithm designed for linear systems at the beginning of its design and cannot be applied to nonlinear systems, but in practice, most systems are nonlinear, and it is impossible to directly use the Kalman filter to solve nonlinear systems. In response to this problem, scholars began to transform the Kalman filter to improve and optimize its performance so that it can be applied to nonlinear systems. In this process, an extension of the Kalman filter to nonlinear systems has been produced.

This paper uses the data fusion algorithm based on the extended Kalman filter to realize the human behavior recognition capability of the inertial sensor. Figure 2 shows the specific implementation process of the data fusion algorithm based on the extended Kalman filter. As shown in the figure, the system has three different types of input data. The gyro data is measured in deg/s, and the angular velocity is measured on three axes. The accelerometer data measures acceleration in g . The magnetometer data measures the magnetic field in Gaussian units. The gyro data is used as an input to update the prior state estimation. At the same time, the accelerometer and magnetometer data are used as a fixed reference to correct the error in the prior state estimation. The following is the specific implementation process of this method.

In the data fusion algorithm based on the extended Kalman filter, it is defined that $\{n\}$ represents the navigation coordinate system and $\{b\}$ represents the sensor coordinate system. First, the initial Euler angle is calculated using the accelerometer and magnetometer data acquired at the initial moment. The initial pitch angle x_0 and the initial roll angle y_0 are obtained from the acceleration data, and the initial yaw angle z_0 is obtained from the magnetic field data which are as follows:

$$\begin{aligned} x_0 &= \arcsin \frac{a_y}{\sqrt{a_x^2 + a_y^2 + a_z^2}}, \\ y_0 &= \arctan \frac{-a_x}{a_z}, \\ z_0 &= \arctan \frac{m_x \cos(y_0) + m_b \sin(y_0)}{m_x \sin(y_0) \sin(x_0) + m_y \cos(x_0) - m_z \cos(y_0) \sin(x_0)}, \end{aligned} \quad (1)$$

where $a_k = [a_x, a_y, a_z]^T$ defines the acceleration data at the initial moment, and $m_k = [m_x, m_y, m_z]^T$ presents the magnetic field data at the initial moment. Through the conversion between the Euler angle and a quaternion, the quaternion of the initial moment state can be obtained.

The quaternion is a four-dimensional vector that can express various poses in space without any singularity. The posture represented by the quaternion can be understood as a coordinate system rotated by another coordinate system. Then, the final coordinate relative to the original coordinate can be regarded as a rotation of the original coordinate,

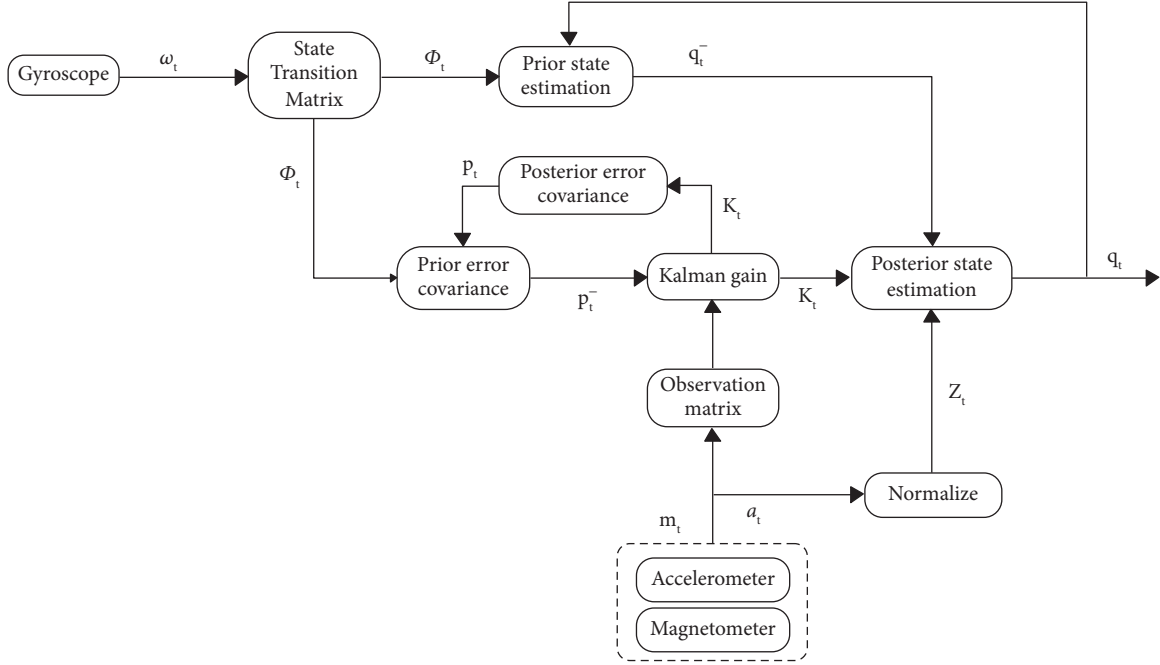


FIGURE 2: Flowchart of the data fusion algorithm based on the extended Kalman filter.

assuming that the angle of rotation is θ , $K = [k_x \ k_y \ k_z]^T$ represents the equivalent rotation axis, then $q = [q_0 \ q_1 \ q_2 \ q_3]^T = [\cos \theta/2 \ k_x \sin \theta/2 \ k_y \sin \theta/2 \ k_z \sin \theta/2]^T$ and $q_0^2 + q_1^2 + q_2^2 + q_3^2 = 1$. The state transition equation in the continuous domain mainly uses quaternion differential equations, which can be defined as follows:

$$\dot{q} = \frac{1}{2} \Omega(\omega_t)q, \quad (2)$$

where \dot{q} represents the differential of the quaternion, $\Omega(\omega_t)$ is the rotation matrix derived from the nature of the quaternion, expressed by the angular velocity data $\omega_t = [\omega_x, \omega_y, \omega_z]^T$ measured by the gyroscope at the time t and has the following form:

$$\Omega(\omega_t) = \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix}. \quad (3)$$

The application field of the Kalman filter is the discrete domain. For this reason, it is necessary to convert the quaternion differential equation in the continuous domain into the discrete domain. By displaying the derivation, the continuous-time system can be rewritten as follows:

$$\begin{aligned} \dot{q} &= \lim_{T \rightarrow 0} \frac{q(t+T) - q(t)}{T} \\ &= \frac{1}{2} \Omega(\omega_t)q(t). \end{aligned} \quad (4)$$

To transform this equation into a discrete-time equation, the time step T between each execution of the algorithm is

used in the digital system. In this way, we can know the following:

$$\begin{aligned} q(t+T) &= q(t) + \frac{1}{2} \Omega(\omega_t)q(t)T \\ &= \left(I + \frac{1}{2} \Omega(\omega_t)T \right) q(t). \end{aligned} \quad (5)$$

Because the bias of the gyroscope is also considered in the system, this paper defines the state vector of the system as a unit quaternion and gyroscope bias. The update equation of the state vector can be obtained through the above-mentioned analysis, which is as follows:

$$\begin{bmatrix} q_{t+1}^- \\ \omega_{bt+1} \end{bmatrix} = \Phi_t \bullet \begin{bmatrix} q_t \\ \omega_{bt} \end{bmatrix}, \quad (6)$$

where q_t is the optimal estimated value of the quaternion of the unit behavior at the previous moment, q_{t+1}^- is the sub-optimal estimated value of the quaternion of the unit behavior at the current moment, and Φ_t is the state transition matrix. The specific expression is as follows:

$$\Phi_t = \begin{bmatrix} I_{4 \times 4} + \frac{T}{2} \Omega(\omega_t) & 0 \\ 0 & I_{3 \times 3} \end{bmatrix}. \quad (7)$$

The quaternion norm is not retained in the state equation. Since the angular position can only be correctly represented by a unit quaternion, it is necessary to introduce a normalization unit to ensure the integrity of the data.

The Jacobian matrix of the state transition equation is solved as follows:

$$G_t = \frac{\partial[q_{t+1}/\omega_{bt+1}]}{\partial[q_t/\omega_{bt}]}. \quad (8)$$

It is solved by assuming that the noise between each axis of the accelerometer, gyroscope, and magnetometer is independent of each other. In this system, the state transition matrix that the system relies on is described by the gyroscope data, so the process noise covariance matrix directly depends on the gyroscope. In the measurement model, the measurement noise comes from the noise of the accelerometer and magnetometer. Therefore, the process noise covariance matrix Q_t and the measurement noise covariance matrix R_t can be determined as follows:

$$Q_t = \begin{bmatrix} \omega_n & 0 \\ 0 & \omega_{bn} \end{bmatrix}, \quad (9)$$

$$R_t = \begin{bmatrix} a_n & 0 \\ 0 & m_n \end{bmatrix}.$$

In the abovementioned formula, ω_n is the gyroscope measurement noise, ω_{bn} is the gyroscope bias noise, a_n is the accelerometer measurement noise, and m_n is the magnetometer measurement noise.

Therefore, the prior error covariance at time t can be further solved as follows:

$$P_t^- = G_t P_{t-1} G_t^T + Q_{t-1}. \quad (10)$$

P_{t-1} represents the posterior error covariance matrix of the previous filter iteration. The behavioral state quaternion obtained by (6) is calculated from the output data of the gyroscope and the optimal behavioral state at the previous moment. It is the behavior obtained during the long-term operation by relying only on the output of the gyroscope. The state is not reliable, so it is necessary to fuse the accelerometer and magnetometer data to modify the state to obtain the final optimal behavior.

To calculate the measurement equation of the magnetometer, first convert the magnetic field data from the sensor coordinate system to the navigation coordinate system to obtain the magnetic field data in the navigation coordinate system. m^b represents the magnetic field data in the sensor coordinate system, and m^n represents the magnetic field data in the navigation coordinate system, which are as follows:

$$\begin{bmatrix} 0 \\ m_x^n \\ m_y^n \\ m_z^n \end{bmatrix} = q_t^- \otimes \begin{bmatrix} 0 \\ m_x^b \\ m_y^b \\ m_z^b \end{bmatrix} \otimes (q_t^-)^{\bullet}. \quad (11)$$

In the measurement model, it is assumed that the magnetic field in the navigation coordinate system has no data in the east direction and only in the north and sky directions. After the magnetic field data is processed and then converted to the carrier coordinate system, the

measured value \bar{m}_b of the magnetic field strength can be obtained, which is as follows:

$$\bar{m}_b = (R_b^n)^{-1} \begin{bmatrix} 0 \\ \sqrt{m_x^n \times m_x^n + m_y^n \times m_y^n} \\ m_z^n \end{bmatrix}. \quad (12)$$

The measurement equation of the accelerometer is calculated. In the navigation coordinate system, the reference direction of the gravity field is $g = [0 \ 0 \ 1]^T$, and the measured value of acceleration \bar{a}_b is obtained from the equation:

$$\bar{a}_b = (R_b^n)^{-1} \cdot g. \quad (13)$$

The obtained magnetic field data and acceleration data are normalized, and the processed data is denoted as Z_t . Finally, the total measurement equation can be obtained from the following equation:

$$h_t(q_t^-) = \begin{bmatrix} \bar{a}_b \\ \bar{m}_b \end{bmatrix}. \quad (14)$$

The Jacobian matrix of the measurement equation is calculated as follows:

$$H_t = \frac{\partial h_t(q_t^-)}{\partial(q_t^-)}. \quad (15)$$

Through the calculated prior error covariance, the Jacobian matrix of the measurement equation, and the measurement noise covariance matrix, the Kalman gain can be calculated as follows:

$$K_t = P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1}. \quad (16)$$

The posterior error covariance matrix is updated, which is as follows:

$$P_t = (I - K_t H_t) P_t^-. \quad (17)$$

Finally, a more accurate posterior state estimate can be obtained, which is as follows:

$$q_t = q_t^- + K_t (Z_t - h_t(q_t^-)). \quad (18)$$

3.2. Feature Extraction of Sensor Data Based on CNN. The basic calculation process of the CNN designed in this paper is shown in Figure 3, where N pairs of feature data are taken and weight parameters are considered as input. The corresponding N pairs of data are multiplied, and then the N results of the multiplication are accumulated.

Convolution calculation is the largest calculation part of the CNN model. To improve the computational efficiency of convolution calculations, this paper uses multiple channels and multiple filters to perform calculations in parallel. The specific process is shown in Figure 4. In the figure, the input feature data size is $H_{in} \times W_{in}$, the convolution kernel size is $K_x \times K_y$, the input channel is CH_{in} , the number of convolution kernels is K , and N is the channel parallelism.

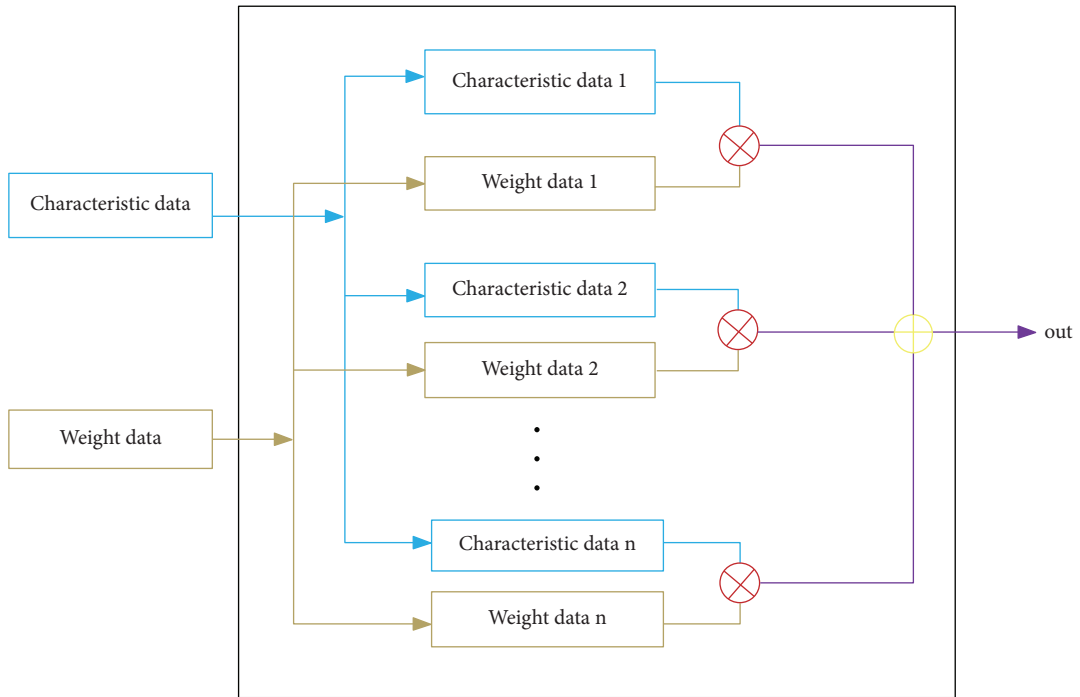


FIGURE 3: Convolution calculation module.

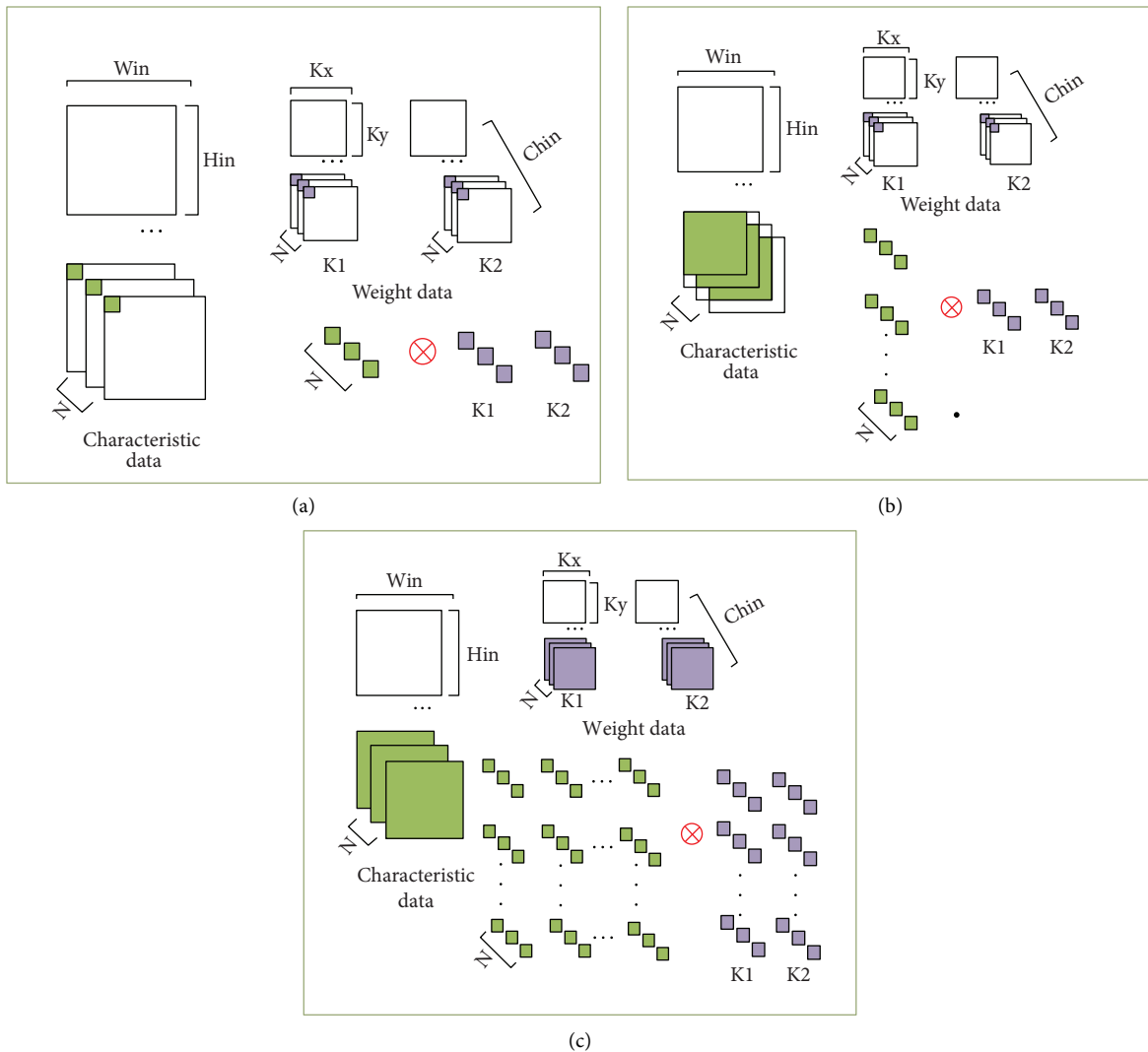


FIGURE 4: Convolution operation flowchart.

Multichannel parallelism cuts the feature data and each weight data into multiple sub-blocks along the input direction. The input channel dimension of each sub-block is N , and N corresponding data is taken out from the feature data sub-block and weight sub-block, respectively. Multi-filter parallelism is the simultaneous calculation of input weights and multiple convolution kernels. Figure 4 shows the process of the sub-block convolution operation. The specific calculation process is as follows:

- (1) As shown in Figure 4(a), the feature data and the first data in the n channel directions of the 2 convolution kernels are taken out at the same time, and the corresponding data taken out respectively are multiplied, then accumulated in the channel direction, and the result is output.
- (2) As shown in Figure 4(b), the weight remains unchanged, and the feature data is slipped, so that the first data on the N channels of the weight is multiplied by part of the feature data and accumulated in the direction of the channel.
- (3) As shown in Figure 4(c), the weight data is changed so that the calculation can cover all the feature data and weight data of the sub-block, and all the data of the 3×3 convolution kernel on n channels and the entire feature data are calculated. The result of multiplying the feature data corresponding to the 9-weight data of the 3×3 convolution kernel and accumulating in the channel direction.
- (4) The next N channel sub-block is taken along the channel direction to repeat the abovementioned calculation, and the result is accumulated to output the characteristic data.

The maximum pooling calculation is shown in Figure 5. The feature data is divided into multiple $n \times n$ sub-blocks along the length and width, and the maximum value is taken in each sub-block. The length and width of the output feature data are the original feature data lengths. And the width of $1/n$.

Pooling is an operation used to reduce the size of data.

As shown in Figure 6, to improve the computational efficiency of the pooling module, this paper designs two subfunctions, horizontal pooling and vertical pooling, and uses pipeline parallel calculation to obtain the output results. At the same time, multichannel parallel acceleration is performed on each subfunction.

Assuming that the input feature data is pooled with the maximum value of $n \times n$ and the step size is also n , the specific calculation process is as follows:

- (1) As shown in Figure 6(a), the input feature enters the horizontal pooling module, the first data is compared with the subsequent input $n - 1$ data, and the maximum value is recorded. Take the input n data as a group, and repeat the abovementioned operations to

obtain the result of horizontal pooling. Currently, the height of the intermediate data remains unchanged, and the width is $1/n$ of the original width.

- (2) As shown in Figure 6(b), first record the first W_{out} data of the intermediate feature input, and compare the $n - 1$ sets of W_{out} data input later with the first set of data at the corresponding positions to obtain the maximum value, and output the result. Take the input n groups of W_{out} data as a group.

4. Results and Discussion

The experiment in this paper is based on the three typical datasets of MHEALTH, WHARF, and USCHAD in the UCI_HAR dataset. These three datasets based on wearable devices are represented in the field of human behavior recognition and can reflect the overall performance of this method. To better illustrate the advantages of the method in this paper, experiments were also carried out on two relatively new datasets, the Stanford-ECM Dataset and DATAEGO [24, 25]. Among them, the Stanford-ECM Dataset was published in 2017. The DATAEGO dataset was made publicly available in 2018. Since the dataset may have problems such as unbalanced sample distribution, and these problems may cause overfitting of the model, it is necessary to prove through experiments whether the improvement in model performance is caused by overfitting. This paper also conducts experiments based on the dataset of the Stanford-ECM Dataset and analyzes the reasons for its performance improvement. To verify the versatility of the method, this paper also uses the network in the classification part of the zero-sample learning task. The semantic relationship parameters between the visible class and the unknown class and the prototype features of the visible class are synthesized as the virtual prototype features of the unknown class. The input and output of the network model are the semantic attributes of each category. The category is determined according to the semantic attributes. Furthermore, this is compared with the method proposed by Chen et al. to verify the performance of the algorithm in this paper [26].

4.1. Experimental Evaluation Index. In the experiment of this paper, three evaluation indicators are used, namely, precision A , recall rate R , and F_1 value. In the field of machine learning, A is the most common evaluation index, which indicates how many of all samples that the model judges to be positive are true positive samples, which is as follows:

$$A = \frac{TP + TN}{TP + TN + FP + FN}, \quad (19)$$

where TP defines a real example, which means that the positive sample is predicted to be the number of samples in the positive class; TN presents a true negative example, which shows that the negative class sample is predicted to be

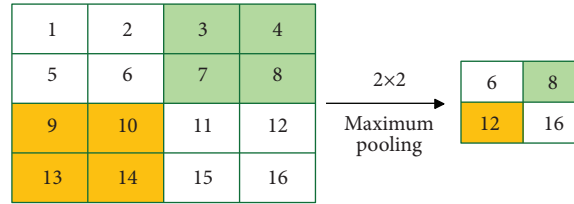


FIGURE 5: Max pooling calculation.

the number of samples in the negative class; FP describes a false positive example, which describes that the positive class sample is predicted to be the number of samples that are predicted to be negative; FN defines a false negative, which explains the number of samples that are predicted to be positive. R , known as recall rate, which indicates how many positive samples are judged by the model as positive samples, is also one of the common evaluation indicators in the field of machine learning. Its calculation method is as follows:

$$R = \frac{TP}{TP + FP} \quad (20)$$

The function of the F_1 value is to consider both A and R when evaluating the model and balancing precision and recall. It is also one of the commonly used evaluation indicators. The calculation method is as follows:

$$F_1 = \frac{2 \times A \times R}{A + R} \quad (21)$$

4.2. The Process of Sample Generation and the Division of Datasets. For sensor data, samples are obtained by sliding on the original data by setting a fixed time window. There are two ways for the time window to slide on the original data: one is to slide without overlap and the other is to slide with half overlap. The two sample generation methods have their advantages and disadvantages. Among them, the main advantages of generating samples in a nonoverlapping manner are that there is no overlap between the samples and that the experimental results will not cause deviation, but the number of samples generated in this way will be relatively small. With half overlap, the advantage of the method of generating samples is that it can generate as many samples as possible, but its disadvantage is that overlap will bring high deviations, resulting in better results than actual. When training the model, the dataset needs to be divided into a training set, a validation set, and a test set. Since the three datasets used in this paper have user information, cross-validation is used to train the model in the process of training the model. That is, one user data is reserved for model verification in each training, and the data of other users is used for the model verification train. Doing so means that each user's data will be used as a validation set, which means that multiple experiments are required. Considering

that there may be situations where the quality of the data collected by the user is low or beyond expectations, the maximum and minimum values are removed in the final evaluation of the model, and the average value is taken to evaluate the performance of the model more comprehensively.

4.3. Experiment and Result Analysis. This paper conducts experiments on the five abovementioned datasets and compares different methods. To ensure comparability between the methods, this paper sets the same experimental conditions and experimental end conditions during the training model. That is, the training model process ends when the loss value is less than or equal to 0.2 or after 200 iterations of training are completed. This paper uses Adadelta as the optimizer, and the Softmax classifier as the classifier. The learning rate of the model is 0.001, and the maximum number of iterations is 200.

It can be seen from Table 1 that compared with the method of Chen et al., the method in this paper has obvious advantages in the three commonly used evaluation indicators. This fully illustrates the necessity of data fusion and the reasonable setting of convolution kernels in data feature extraction for such tasks. The method in this paper has achieved better results on the MHEALTH dataset, and the accuracy reaches 94.08%, which is 2.21% higher than Chen et al.'s method, which is enough to prove the advantage of this method. In addition, the method in this paper also has obvious advantages in the WHARF dataset and the USCHAD dataset, but the effect is not significantly improved in the MHEALTH dataset. The reason is that the dataset is collected in different ways. The MHEALTH dataset has strict data during the collection process. The collection standard regulates the collection action. However, the WHARF dataset and the USCHAD dataset did not develop such strict collection standards during the collection process.

Figure 7 shows the experimental results of the two methods on the Stanford-ECM Dataset. It can be seen from Figure 7 that the method in this paper has achieved better results in 8 categories. With the same results of the method, 5 categories have achieved slightly worse results. However, the overall performance is still significantly improved, indicating that the performance improvement is not caused by the data distribution but the function of the method itself.

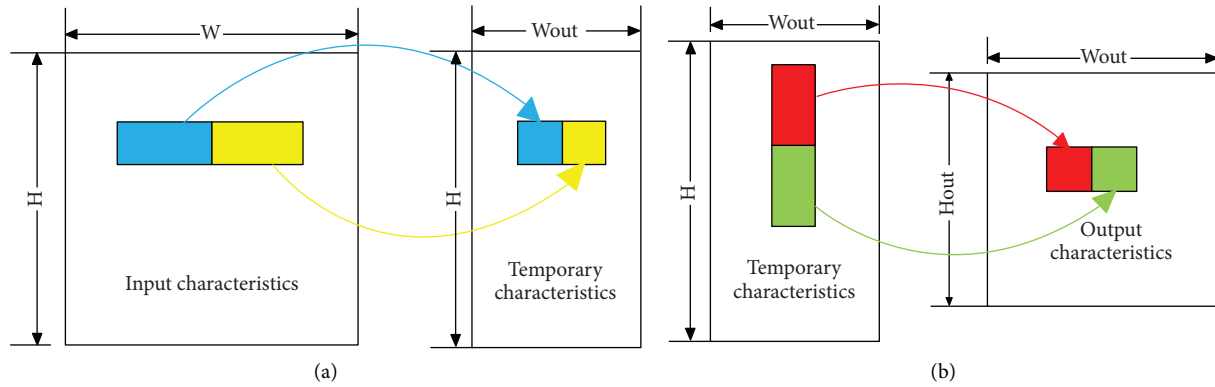


FIGURE 6: Pooling flowchart.

TABLE 1: Experimental results of two methods.

Data set	Method	A (%)	R (%)	F_1 (%)
HEALTH	Chen et al.	98.1	91.3	90.1
	This paper	94.	94.1	93.1
WHARF	Chen et al.	67.1	42.9	42.6
	This paper	68.7	44.1	43.8
USTAD	Chen et al.	78.8	74.6	71.4
	This paper	79.6	76.8	73.2
Stanford-ECM Dataset	Chen et al.	92.3	88.1	88.8
	This paper	93.7	88.8	90.1
DATAEGO	Chen et al.	55.2	43.8	43.0
	This paper	58.2	45.7	44.7

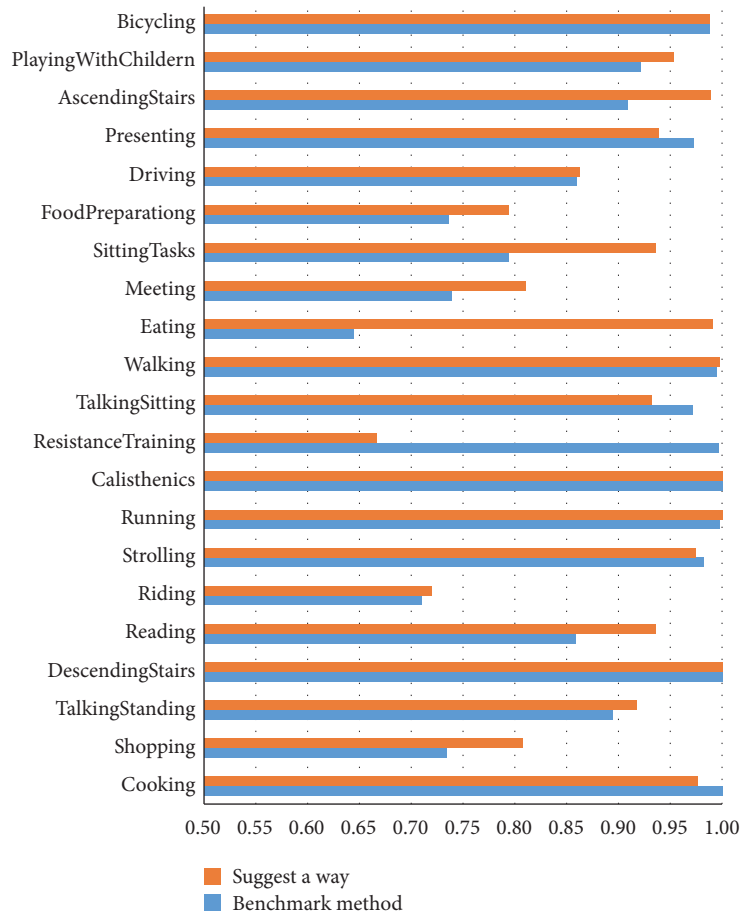


FIGURE 7: Experimental results based on the dataset of the Stanford-ECM Dataset.

5. Conclusions

This paper proposes a method for feature extraction of wearable sensor data based on CNN. This method first uses a Kalman filter to fuse the data to obtain a preliminary state estimation, and then uses CNN to recognize human behavior and obtain the corresponding behavior set. This paper conducts experiments on five datasets. The experimental results show that this paper can extract data features at multiple scales while fully maintaining data independence and can effectively extract corresponding feature data. In addition, the method in this paper has strong generalization ability and can adapt to different learning tasks. At present, wearable devices are mainly smart watches and bracelets, concentrated in medical and health, outdoor sports, audio-visual entertainment, and other fields. With the intensification of society's aging phenomenon, the wearable device market will usher in tremendous development in areas where health is the appeal. In the future, the accurate extraction of data characteristics of wearable sensors can be used to analyze users' various behaviors, the health index, and other data, which also provides strong support for disease prevention and treatment, health monitoring, and other aspects. At the same time, the accurate extraction of wearable sensor data features can complete various tasks such as object detection, face recognition, and image analysis, bringing a brand new experience to users. The accurate extraction of data features of wearable sensors can provide humans with more accurate and reliable basic datasets. Through these datasets, more intelligent and digital behavior parameters can be provided for artificial intelligence and help the development of artificial intelligence.

Data Availability

The datasets used and/or analyzed during the current study are available from the author on reasonable request.

Conflicts of Interest

The author declares no conflicts of interest.

Acknowledgments

This work was supported by the Science and Technology Research Project of the Chongqing Education Commission (KJQN201803105) and the Chongqing Science and Technology Commission (cstc2018jscx-msybX0270).

References

- [1] Z. Song, Z. Cao, Z. Li, J. Wang, and Y. Liu, "Inertial motion tracking on mobile and wearable devices: recent advancements and challenges," *Tsinghua Science and Technology*, vol. 26, no. 5, pp. 692–705, 2021.
- [2] M. Zhang, Y. Chen, and W. Susilo, "PPO-CPQ: a privacy-preserving optimization of clinical pathway query for E-healthcare systems," *IEEE Internet of Things Journal*, vol. 7, no. 10, Article ID 10660, 2020.
- [3] H. Shen, M. Zhang, H. Wang, F. Guo, and W. Susilo, "A cloud-aided privacy-preserving multi-dimensional data comparison protocol," *Information Sciences*, vol. 545, pp. 739–752, 2021.
- [4] J. Yan, Y. Meng, X. Yang, X. Luo, and X. Guan, "Privacy-Preserving localization for underwater sensor networks via deep reinforcement learning," *IEEE Transactions on Information Forensics and Security*, vol. 16, pp. 1880–1895, 2021.
- [5] I. C. Jeong, D. Bychkov, and P. C. Searson, "Wearable devices for precision medicine and health state monitoring," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 5, pp. 1242–1258, 2019.
- [6] K. Anton and A. Umek, "Wearable sensor devices for prevention and rehabilitation in healthcare: swimming exercise with real-time therapist feedback," *IEEE Internet of Things Journal*, vol. 6, pp. 1331–1341, 2019.
- [7] J. Hernandez, D. McDuff, K. Quigley, P. Maes, and R. W. Picard, "Wearable motion-based heart rate at rest: a workplace evaluation," *IEEE Journal of Biomedical and Health Informatics*, vol. 23, no. 5, pp. 1920–1927, 2019.
- [8] K. Ueafuea, C. Boonnag, T. Sudhawiyangkul et al., "Potential applications of mobile and wearable devices for psychological support during the COVID-19 pandemic: a review," *IEEE Sensors Journal*, vol. 21, no. 6, pp. 7162–7178, 2021.
- [9] I. M. Saied, S. Chandran, and T. Arslan, "Integrated flexible hybrid silicone-textile dual-resonant sensors and switching circuit for wearable neurodegeneration monitoring systems," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 13, no. 6, pp. 1304–1312, 2019.
- [10] Q. Meng, X. Lai, Z. Yan, C. Y. Su, and M. Wu, "Motion planning and adaptive neural tracking control of an uncertain two-link rigid-flexible manipulator with vibration amplitude constraint," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 10, pp. 1–15, 2021.
- [11] Y. Liu, J. Shen, W. Wang, H. Sun, and L. Shao, "Better dense trajectories by motion in videos," *IEEE Transactions on Cybernetics*, vol. 49, no. 1, pp. 159–170, 2019.
- [12] T. Sui, D. Marelli, X. Sun, and M. Fu, "Multi-sensor state estimation over lossy channels using coded measurements," *Automatica*, vol. 111, Article ID 108561, 2020.
- [13] W. Zhou, J. Liu, J. Lei, L. Yu, and J. N. Hwang, "GMNet: graded-feature multilabel-learning network for RGB-thermal urban scene semantic segmentation," *IEEE Transactions on Image Processing*, vol. 30, pp. 7790–7802, 2021.
- [14] P. J. Chacon, L. Pu, T. H. D. Costa et al., "A Wearable pulse oximeter with wireless communication and motion artifact tailoring for continuous use," *IEEE Transactions on Biomedical Engineering*, vol. 66, no. 6, pp. 1505–1513, 2019.
- [15] T. Gu, L. Wang, Z. Wu, X. Tao, and J. Lu, "A pattern mining approach to sensor-based human activity recognition," *IEEE Transactions on Knowledge and Data Engineering*, vol. 23, no. 9, pp. 1359–1372, 2011.
- [16] A. Wang, S. Zhao, C. Zheng, H. Chen, L. Liu, and G. Chen, "HierHAR: sensor-based data-driven hierarchical human activity recognition," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3353–3365, 2021.
- [17] X. C. Xiaobin, W. W. S. Zheng, and J. Z. Jianguo, "Learning person-person interaction in collective activity recognition," *IEEE Transactions on Image Processing*, vol. 24, no. 6, pp. 1905–1918, 2015.
- [18] W. Ye, J. Cheng, F. Yang, and Y. Xu, "Two-stream convolutional network for improving activity recognition using convolutional long short-term memory networks," *IEEE Access*, vol. 7, pp. 67772–67780, 2019.
- [19] A. Torfi, S. M. Iranmanesh, N. Nasrabadi, and J. Dawson, "3D convolutional neural networks for cross audio-visual

- matching recognition,” *IEEE Access*, vol. 5, pp. 22081–22091, 2017.
- [20] X. Jin, C. Xia, N. Guan et al., “Real-time scheduling of massive data in time sensitive networks with a limited number of schedule entries,” *IEEE Access*, vol. 8, pp. 6751–6767, 2020.
- [21] L. Wu, Y. Wang, L. Shao, and M. Wang, “3-D PersonVLAD: learning deep global representations for video-based person reidentification,” *IEEE Transactions on Neural Networks and Learning Systems*, vol. 30, no. 11, pp. 3347–3359, 2019.
- [22] D. A. Chanti and A. Caplier, “Deep learning for spatio-temporal modeling of dynamic spontaneous emotions,” *IEEE Transactions on Affective Computing*, vol. 12, no. 2, pp. 363–376, 2021.
- [23] J. Qi, K. Sun, J. Wang, and H. Liu, “Dynamic state estimation for multi-machine power system by unscented kalman filter with enhanced numerical stability,” *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 1184–1196, 2018.
- [24] K. Nakamura, S. Yeung, A. Alahi, and F. F. Li, “Jointly learning energy expenditures and activities using egocentric multimodal signals,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 6817–6826, Honolulu, HI, USA, July 2017.
- [25] R. Possas, S. P. Caceres, and F. Ramos, “Egocentric activity recognition on a budget,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5967–5976, Salt Lake City, UT, USA, June 2018.
- [26] Y. Chen and Y. Xue, “A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer,” in *Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*, pp. 1488–1492, Hong Kong, China, October 2015.