Hindawi
Journal of Sensors
Volume 2022, Article ID 4102552, 16 pages
https://doi.org/10.1155/2022/4102552

Research Article

A Lightweight Human Action Classification Method for Green IoT Sport Applications

Da Xiao,[1] **Tianyu Huang**,[1,2] **Yihao Li**,[1] **Chang Liu**,[1] **and Fuquan Zhang**[2,3]

[1]*Department of Computer Science and Technology, Beijing Institute of Technology, 100081, China*
[2]*Beijing Key Laboratory of Digital Performance and Simulation Technology, 100081, China*
[3]*College of Computer and Control Engineering, Minjiang University, 350108, China*

Correspondence should be addressed to Tianyu Huang; huangtianyu@bit.edu.cn

Received 3 May 2022; Accepted 15 June 2022; Published 30 June 2022

Academic Editor: Saru Kumari

This paper proposes a lightweight human action classification method for Green Internet of Things (IoT) sport applications. This method classifies the human motion data collected by wearables or other IoT devices with energy-efficient techniques, by enabling a small number of sample training and incremental classification to achieve the purpose of energy-efficient. To lessen the complexity of the model and reduce the number of samples required for parameter estimation, we propose a shared Hidden Conditional Random Field (sHCRF) model. The sHCRF model adds a shared-classification layer structure to reduce the parameter computation. In the experiments, the classification accuracy of the sHCRF model is above 95%. This paper introduces an incremental learning method based on knowledge distillation. The new model suppresses the forgetting of existing classification knowledge while fitting new data to learn new classification knowledge. In the incremental scenarios, the classification accuracy of the sHCRF model is above 70%. The experimental results show that this method can lightly implement convenient and fast automatic classification of action acquisition.

1. Introduction

In recent years, smart wearables and mobile sensors have become an integral part of human's daily lives [1] with the development of Green Internet of Things (IoT). These wearables can be used to capture the human motion data when people exercise or engage in other daily activities. By analyzing and classifying the captured motion data, it can be used to assist exercise and monitor the human health, etc., in which the results of analysis or classification can be fed back through sport applications, like sport watches and smart treadmill. Human action recognition is a higher-level work for computers to understand human motion. Among them, the motion categories are the base for human action recognition. Due to the complexity and temporal impact of human motion, high-dimensional features of motion data complicate human action classification. Modeling of human motion involves the relationship between thousands of variables. Researchers have proposed human action recognition models based on neural networks [2], like convolutional neural networks (CNN) [3], Recurrent Neural Networks (RNN) [4], and Graph Convolutional Neural (GCN) [5] networks. However, these models require more energy, and they need to be trained on a large number of manually labelled samples. The training process is time-consuming and labor-intensive. Our work is aimed at building a lightweight action classification method to achieve the purpose of energy efficiency, which enables the model training on a small number of samples while saving storage and computational overhead. The probabilistic graphical model has a strong ability to model the relationship between variables based on prior knowledge, greatly reducing model parameters and, more importantly, reducing the amount of sample data required for parameter estimation. It is potential to model motion lightweightly based on the probabilistic graphical models.

The human motion data captured by sensors is time series data, and each frame represents the posture of the human body at the current moment. It is necessary to pay attention to the spatiotemporal information so as to accurately describe the characteristics of human action classification. The Hidden Conditional Random Field (HCRF) model, as an undirected probabilistic graphical model, is a discriminative model. It can label the entire sequence of samples as actions and use hidden state variables to capture intermediate structures. For action recognition task, the existing consensus is that the ideal model should be derived and optimized on the basis of maximizing the discriminant function.

This paper proposes a lightweight human action classification method. We introduce posture base, posture change base, and posture semantic base to characterize human motion data. The features are obtained based on the data collected by sensors corresponding to the main kinematic joints. In order to reduce the computational cost of training, we propose a shared Hidden Conditional Random Field (sHCRF) model by designing a shared-classification layer structure, which reduces parameter amounts. The incremental learning and classifying methods are designed by introducing knowledge distillation. The framework of our method is shown in Figure 1. Human motion data is continuously collected by smart wearables, and then, the collected data is characterized. The features are sent to the sHCRF model for training. When the model fits new data, the model will add the distillation loss on the basis of the classification loss, so that the model can suppress forgetting owned knowledge while learning new classification knowledge. After training the model, the updated classification knowledge will be uploaded to the IoT. The contributions of this paper are as follows:

(1) A lightweight human action classification method for Green IoT sport applications is proposed. The method is based on probabilistic graphical model training on a small number of samples. In the meantime, it can realize automatic incremental learning and classifying by knowledge distillation

(2) Human action features are designed, including posture sequence, posture change sequence, and posture semantic sequence. They can describe human posture features and the temporal correlation between human postures

(3) A shared-classification layer structure is introduced to improve the HCRF probabilistic graphical model. It lessens parameter amount and achieves better classification

2. Related Work

Our lightweight action classification method is related to feature description of human motion, the human action classification model, and incremental learning methods.

2.1. Feature Representation Method of Human Motion Data. Human body motion data contain complex information due to the many multidimensional data involved. The semantic motion information of the motion data cannot be directly reflected. So, it is necessary to extract the features of the data. The extracted motion data features provide a basis for similarity measurement between different actions. Forbes and Fiume [6] used the weighted Principal Component Analysis (PCA) dimensionality reduction algorithm to extract the feature representation of motions. Still, the corresponding relationship between the features extracted by this method and the motion semantics is not apparent. Müller et al. [7] proposed a method for indexing the geometric parts of the human body. They defined 31 Boolean features to describe the geometric relationship of the human posture, ensuring the unity of logical similarity and numerical similarity of human motion. Their method has groundbreaking significance, but it had complicated feature definition. Liang et al. [8] used the technique of subspace division to represent the geometric features of human posture. They defined a set of feature vectors to represent motion data based on this method. This method is concise for the feature definition of human posture.

Combining the geometric features of the human body and the method of subspace division, the motion data features in this paper are divided into parts that describe specific human postures, posture changes, and features that characterize the primary state of posture. The first is from the static and dynamic perspectives, and the last is related to the parameter construction of the model. Semantic information in the same essential state in different actions may differ.

2.2. Human Action Classification Method. Action classification methods for human action need to model the temporal and spatial information in motion data. In recent years, researchers have proposed many action classification methods based on neural network. The methods they used include the model based on RNN [9], CNN [10, 11], GCN [12–14], and Long Short-Term Memory (LSTM) [15]. The probabilistic graphical model is one of the popular solutions to the problem of action classification. The probabilistic graphical model is divided into a directed graph model and an undirected graph model. Both of them are suitable for modeling sequence models. Ma et al. [16] used the Hidden Markov Model (HMM) to recognize human action. Samr and Nizar [17] used the Beta-Liouville HMM action classification method. Wen et al. [18] used a Hierarchy Dirichlet Process-Hidden Markov Model (HDP-HMM) to represent the action class. This method can automatically obtain the number of hidden states during the learning process. Using undirected probabilistic graphs, Vrigkas et al. [19] used the HCRF model to recognize the human activities. The modified Hidden Condition Random Field (mHCRF) model based on the HCRF model was proposed by Zhang and Gong [20]. Their method optimized the algorithm by introducing the exact hidden state sequence obtained from the HMM to prevent it from falling into a local optimum.

We use the sHCRF model for action classification, which is based on a probabilistic graph model. It is suitable for modeling the temporal correlation in sequence data, and the graph model supports sequence data input with different lengths.

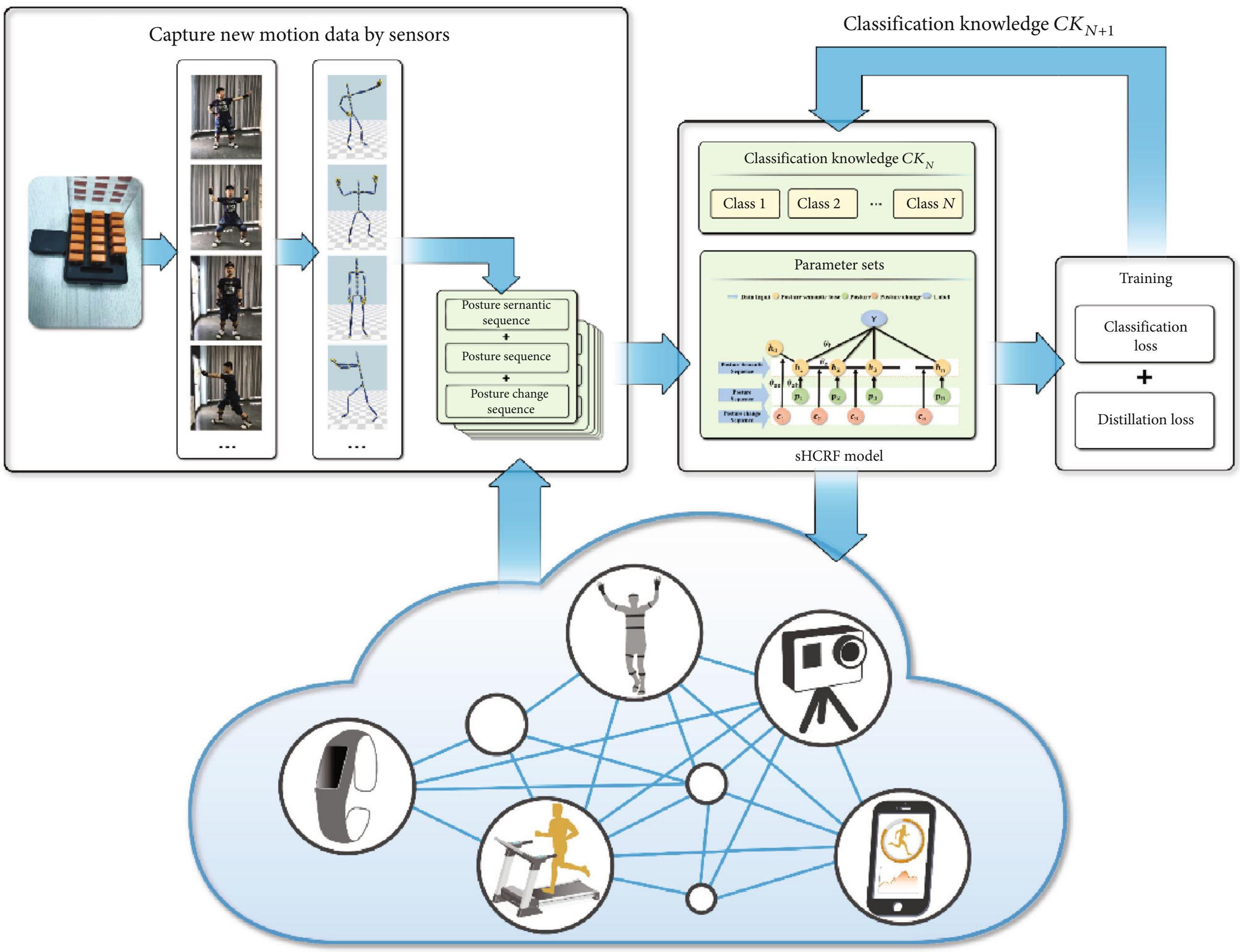
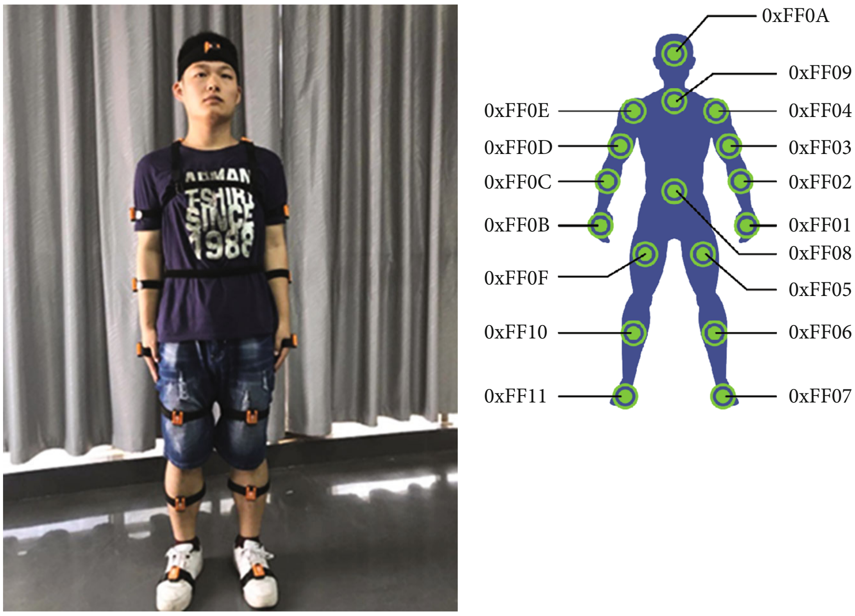


Figure 1: The framework of the action classification for Green IoT.

2.3. Incremental Learning Method. At present, among the methods for achieving category increment, the incremental learning [21] method is showing adequate results in various classification techniques. Incremental learning is a system that continuously learns from new samples and saves most previously learned knowledge. Whenever new data are added, it is not necessary to rebuild all the knowledge bases. Only the changes caused by the new data update the original knowledge bases. The incremental learning method is more closely aligned with the principles of human thinking.

Incremental learning methods based on regularization usually do not need to use old data to let the model review the tasks it has learned. The Learning without Forgetting (LwF) algorithm proposed by Li and Hoiem [22] does not need to use the data of an old task and can fit new data while suppressing the forgetting of old knowledge. The main idea of the LwF algorithm comes from the knowledge distillation method proposed by Hinton et al. [23], which predicted the new model on a new task similar to the prediction of the old model on the new task. Irfan et al. [24] proposed a model which can handle both multitask and single incremental task scenarios as opposed to various existing models that cover only the multitask scenarios. The Elastic Weight Consolidation (EWC) algorithm based on the Bayesian framework proposed by Kirkpatrick et al. [25] introduces an additional parameter-related regular loss. The loss encourages the new model parameters obtained by further task training to be as close as possible to the old model parameters according to the importance of different parameters. In the replay-based incremental learning method, when training a new task, part of the old representative data is retained and used for the model to review the old knowledge learned. It may overfit the retained old data. Lopez-Paz and Ranzato [26] proposed the Gradient Episodic Memory (GEM) algorithm for this problem. It only updates the parameters of the new task without interfering with the parameters of the old task. It modifies the gradient update direction of the new task in an inequality-constrained manner so that the model does not increase the old one. Concurrently with the loss of the task, it tries to minimize the loss value of the new task.

Most incremental learning methods are based on convolutional neural networks, and they have few applications to probabilistic graphical models. Moreover, there is no clearly applicable method in action classification. Based on the analysis and conclusions presented in Section 2.2, the action classification model proposed in this paper uses the



(a) Schematic diagram of sensor wearing

(b) Sensor position correspondence

Figure 2: Motion capture sensor set.

probability graph model as its basic model and introduces the idea of "shared parameters" in the depth model, which has the conditions for category increment.

3. Collection and Characterization of Motion Capture Data

The human action features in this paper are extracted from motion capture data. The motion capture data is represented as the skeleton hierarchy, which refers to the motion nodes as joints. They have characteristics that describe human posture, posture changes, and features that define deep semantic information of human actions. The first two are composed of a posture base and a posture change base, respectively, and describe the static and dynamic characteristics of the data, respectively. The last feature is referred to as the posture semantic base in this paper. The posture semantic base represents the essential characteristics of a human posture. The complete set of posture semantic base defined in this paper is the posture semantic base set $\mathscr{H}_D$, where D is the size of the collection. The state transition set T_C represents all posture semantic base transitions between the posture semantic bases, where C is the size of the set.

3.1. Motion Capture Data Collection. The motion acquisition device consists of 17 node sensors and a hub developed by our research group. As shown in Figure 2, there is a 9-axis sensor of each node, containing a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer. The precision of the sensor is +0.5 degrees. The original data collected are gravity acceleration, rotation rate, and magnetic force data, respectively. The reconstructed motion data is solved by a 9-axis fusion algorithm. It is identified by the specific data ID of each sensor, where the position of sensor nodes corresponds to human joints. The IDs are shown in Table 1.

Table 1: Sensor position correspondence.

MTS_ ID	Corresponding location	MTS_ ID	Corresponding location
0xFF01	Left hand	0xFF0B	Right hand
0xFF02	Left low-arm	0xFF0C	Right low-arm
0xFF03	Left up-arm	0xFF0D	Right up-arm
0xFF04	Left shoulder	0xFF0E	Right shoulder
0xFF05	Left up-leg	0xFF0F	Right up-leg
0xFF06	Left low-leg	0xFF10	Right low-leg
0xFF07	Left foot	0xFF11	Right foot
0xFF08	Waist		
0xFF09	Chest		
0xFF0A	Head		

3.2. Posture Base. Human motion capture data are sequence data of multiple frames of human posture arranged along the time axis. Each frame is composed of three-dimensional rotation data for each joint of the human body. In this paper, we define the number of primary moving joints M. The primary moving joints of the human body include the joints on the arms (including shoulders, elbows, and wrists), leg joints (including hips, knees, and ankles), and torso joints (mainly the abdomen). When M is 13, the primary moving joint is composed of all arm joints and leg joints, plus abdominal joints. This paper takes the root joint coordinate system of the initial posture of the human motion data as the absolute coordinate system of the primary moving joint. The rotation matrix of the moving joint is obtained from its Euler rotation angle. The direction vector of the moving joint is taken from a unit vector parallel to the direction vector of the coordinate axis in the absolute

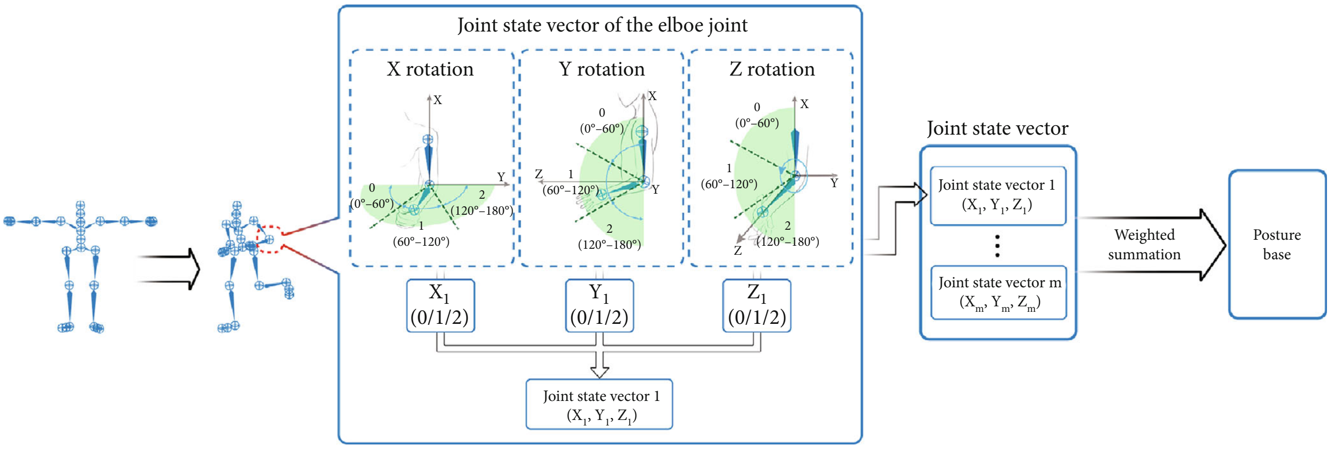


Figure 3: The construction process of the posture base.

coordinate system. In the local coordinate system of the joint, the spherical coordinate of the moving joint is composed of the angle between the direction vector and the axis of the local coordinate system.

To get the posture base of the joint, we first divided the rotation space of the joint into three subspaces by each axis according to the knowledge of human kinematics and related motion experience. Then, we selected a direction vector of the joint and calculate the angles between the vector and each coordinate axis. We used the angles to determine the subspace where the vector is located by each axis, and we obtained a joint state vector, whose length was 3. We also obtained other state vectors by selecting different direction vectors. Finally, we selected the required values from the obtained state vector to get a weighted sum. The weighted sum is the posture base of the joint. As shown in the construction process of the posture base of the elbow joint in Figure 3, we first select the lower arm bone as the direction vector of the elbow joint. The rotation space of the elbow joint is divided into three subspaces by each axis, which are $0^\circ \sim 60^\circ$, $60^\circ \sim 120^\circ$, and $120^\circ \sim 180^\circ$, respectively. A three-bit code is defined according to the spherical coordinates, and each bit code can take 0, 1, or 2 according to the angle interval of each dimension in the spherical coordinates. The code is a ternary code. At this point, the information about the direction vector of the joint rotating around itself is missing; so, it is necessary to introduce the spherical coordinate information of other vectors perpendicular to the direction vector to obtain other ternary code and to construct the posture base of the elbow joint in the current frame. The formula of posture base can be expressed as follows:

$$pb_i^k = f_{\mathrm{p}}\left(\mathbf{J}_i^k, \mathbf{V}_k\right) = \sum_{l=0}^{n} Z^{\alpha\left(\mathbf{J}_i^k, \mathbf{V}_k^l\right)}, \tag{1}$$

$$k \in (0, M], i \in (0, N], n \in [2, 3], l \in [1, n]. \tag{2}$$

pb_i^k represents the posture base of the kth joint in the ith frame data. $f_p(\mathbf{J}_i^k, \mathbf{V}_k)$ is the characteristic function. $\mathbf{J}_i^k$ represents the spherical coordinates of the kth joint in the ith frame data. $\mathbf{V}_k$ is the direction vector set of the kth joint. $\sum_{l=0}^{n} Z^{\alpha}$ is the weighted sum of joint states for the purpose of compressing information. n represents the number of joint state vectors, and its value is related to the rotational freedom of the joint. Z represents the number of divisions of the joint's subspace. Function α is the method to determine the state of the included angle in the corresponding dimension. $\mathbf{V}_k^l$ represents the lth direction vector in $\mathbf{V}_k$. Function α is defined as

$$\alpha\left(\mathbf{J}_\mathbf{i}^\mathbf{k}, \mathbf{V}_\mathbf{k}^\mathbf{l}\right) = \begin{cases} 0, & \left\langle \mathbf{J}_i^k, \mathbf{V}_k^l \right\rangle \in (\beta_1, \beta_2), \\ 1, & \left\langle \mathbf{J}_i^k, \mathbf{V}_k^l \right\rangle \in (\beta_2, \beta_3), \\ 2, & \left\langle \mathbf{J}_i^k, \mathbf{V}_k^l \right\rangle \in (\beta_3, \beta_4), \end{cases} \tag{3}$$

$$0^\circ \leq \beta_1 < \beta_2 < \beta_3 < \beta_4 \leq 180^\circ. \tag{4}$$

$\langle \mathbf{J}_i^k, \mathbf{V}_k^l \rangle$ represents the angle between $\mathbf{J}_i^k$ and $\mathbf{V}_k^l$. β_1, β_2, β_3, and β_4 are subspace boundary values of joint rotation space. In Figure 3, $\beta_1 = 0^\circ$, $\beta_2 = 60^\circ$, $\beta_3 = 120^\circ$, and $\beta_4 = 180^\circ$.

3.3. Posture Sequence and Posture Change Sequence. The human posture sequence is a time series composed of postures. The posture $\mathbf{p}_i$ of the ith frame can be expressed as $\{pb_i^1, pb_i^2, \cdots, pb_i^M\}$. The posture sequence $\mathbf{P}^N$ is expressed as $\{\mathbf{p}_1, \mathbf{p}_2, \cdots, \mathbf{p}_N\}$.

The posture change base is the code obtained by changes of the spherical coordinates of the corresponding joint between adjacent moments—increasing, decreasing, and unchanged—similar to the coding method of the posture base. Its code is computed by the included angle change value of each dimension. The coding formula is expressed as follows:

$$pc_i^k = f_{\mathrm{c}}\left(\mathbf{J}_i^k\right) = \sum_{l=0}^{n} 3^{d\left(\mathbf{J}_i^{k,l}, \mathbf{J}_{i+1}^{k,l}\right)}, \tag{5}$$

$$k \in (0, M], i \in (0, N], n \in [2, 4], l \in [1, 3]. \tag{6}$$

pc_i^k represents the posture change base of the kth joint between the ith and the $(i+1)$-th frame. $f_c(\mathbf{J}_i^k)$ is the characteristic function. $\sum_{l=0}^{n} 3^d$ is the weighted summation, where 3 indicates that the angle changes of the joints in three coordinate axis directions need to be considered. d is the coding function for changing spherical coordinates. It is defined as

$$d\left(\mathbf{J}_i^{k,l}, \mathbf{J}_{i+1}^{k,l}\right) = \begin{cases} 0, & \mathbf{J}_{i+1}^{k,l} - \mathbf{J}_i^{k,l} > 0, \\ 1, & \mathbf{J}_{i+1}^{k,l} - \mathbf{J}_i^{k,l} = 0, \\ 2, & \mathbf{J}_{i+1}^{k,l} - \mathbf{J}_i^{k,l} < 0. \end{cases} \quad (7)$$

$\mathbf{J}_i^{k,l}$ and $\mathbf{J}_{i+1}^{k,l}$ are the lth dimensional angles in $\mathbf{J}_i^k$ and $\mathbf{J}_{i+1}^k$, respectively. The posture change $\mathbf{c}_i$ of the ith and the $(i+1)$-th frame can be expressed as $\{pc_i^1, pc_i^2, \cdots, pc_i^M$ $\}$. The posture change sequence $\mathbf{C}^N$ can be expressed as $\{\mathbf{c}_1, \mathbf{c}_2, \cdots, \mathbf{c}_N\}$.

3.4. Posture Semantic Base. The posture semantic base of a frame in the human motion data represents the essential characteristics of the current posture. The posture semantic base corresponding to different postures may be the same, but it may represent different semantic information among the actions. A collection of posture semantic bases in a movement can be a feature representation of the campaign. This collection is a subset of the posture semantic base set $\mathscr{H}_D$.

The posture semantic base considers the dynamic and static characteristics of human motions. It is a K-dimensional integer vector $\mathbf{h}^K = \{\mu_1, \cdots, \mu_p, \upsilon_1, \cdots, \upsilon_q\}$, where p -dimensional data $\boldsymbol{\mu} = \{\mu_i | i = 1, \cdots, p\}$ describe the state of p motion joints of the human body and represent the state of the selected joint in the human body coordinate system. This method divides the joint rotation space into two subspaces, corresponding to the rotation angle of the joint. Taking the shoulder joint as an example, this method divides the rotation space of the shoulder joint into upper and lower subspaces, where the horizontal plane is used as the interface. The subspace is for determining the state of the joint; in addition, the q-dimensional data $\boldsymbol{\nu} = \{\nu_j | j = 1, \cdots, q\}$ describe the movement of the human body and are reflected by the angle between the direction of human motion and the direction of the human face in the current frame. For example, when $K = 8$, $p = 6$, and $q = 2$, $\boldsymbol{\mu}$ represents 6 states of the shoulders, hips, root joint, and abdomen joints; $\boldsymbol{\nu}$ represents the movement of the human body in the horizontal and vertical directions. The size of the posture semantic base set $\mathscr{H}_D$ is related to the definition of the posture semantic base, which is 2^K.

The posture semantic base is used as a hidden state node in the HCRF model. But in the HCRF model, the hidden state sequence is uncertain. In our method, we designed the posture semantic sequence as a certain input of the hidden state layer to optimize the model.

4. An Incremental Action Classification Method Based on the sHCRF Model

This paper proposes a new action classification model based on the human action features described above, namely, the shared Hidden Conditional Random Field model. The model was improved based on HCRF, and it introduces a structure with a shared-classification layer. To solve the problem of data storage during the learning process, this study used batched incremental learning.

4.1. The Shared-Classification Layer Structure. The sHCRF model introduces a shared-classification layer structure, which has two layers. One is the shared layer, which is used to extract the motion semantic information from the features. The other is the classification layer, which uses the information from the shared layer to classify the samples. The structure can reduce the redundancy of model parameters, as well as maintain high classification accuracy.

The shared layer is composed of two parts, one is used to process posture sequence, and it is a parameter matrix of size $M \times D$. The other is used to process posture change sequence, and it is a $M \times C$ parameter matrix. The former is constructed based on the posture semantic base set $\mathscr{H}_D$, and the latter is constructed based on the posture semantic base transition set T_C. Both the parts get the spatiotemporal information of the human action from the features. Figure 4 shows the shared layer for posture. As shown in Figure 4, in the shared layer for posture of the sHCRF model, each column vector $\mathbf{h}_i$ represents a posture semantic base. There are N posture semantic bases in the structure. Each action has its posture semantic base set, which is a subset of the posture semantic base set $\mathscr{H}_D$.

The shared layer's output is input to the classification layer. Each category corresponds to a classification layer. The number of the columns of the two-parameter matrices in the classification layer is D and C, respectively, and the number of the rows of both the two-parameter matrices is the number of the data categories. The classification layer uses these data to calculate the probability that the sample belongs to each category. At last, the category with the highest probability is used as the label of the sample.

The inference process of the sHCRF model, which uses the shared-classification structure, is as follows. When performing human action classification tasks, the model first obtains the semantic information of the human posture and posture changes in the human motion data. It then brings this information into the specific human body motion and analyzes the time-space relationship of motion again. Finally, it obtains the probability that the input sample belongs to the current motion category.

4.2. sHCRF Model Introduction. The sHCRF model further optimizes the classic HCRF model by introducing a shared-classification layer structure. This structure significantly reduces the model's computational complexity and improves its speed of motion modeling and classification accuracy. The classical HCRF model obtains the classification probability of a given input by fitting the conditional

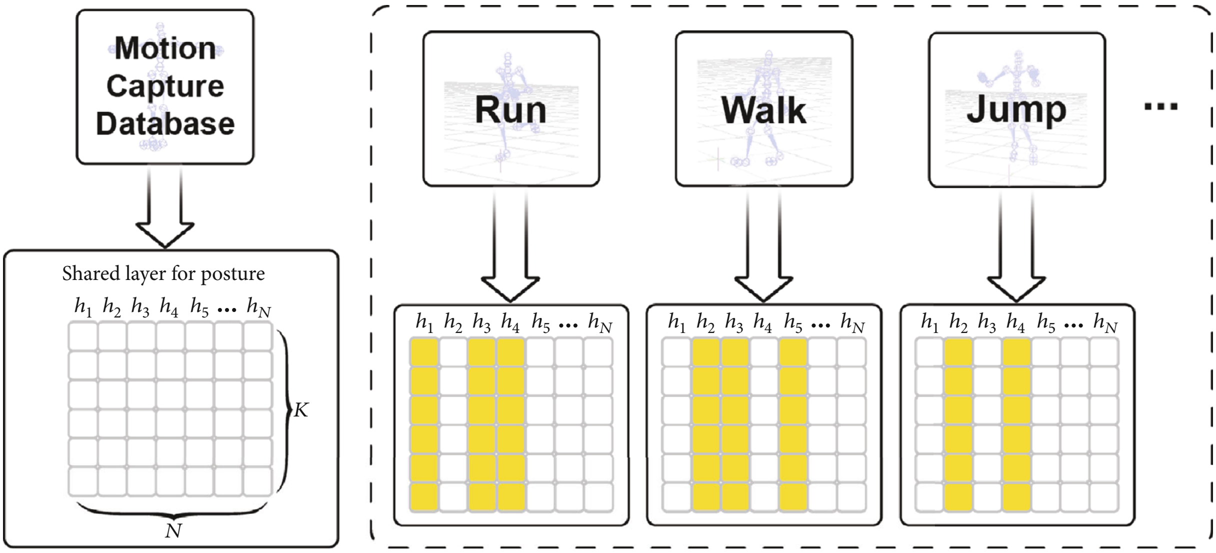

Motion Capture Database
Shared layer for posture
h_1 h_2 h_3 h_4 h_5 … h_N
K
N
Run
Walk
Jump
…
h_1 h_2 h_3 h_4 h_5 … h_N

Figure 4: Diagram of sHCRF for motion categories. The left part corresponds to the shared layer for posture of the sHCRF model, and the right part shows the corresponding posture semantic base set (a subset of the posture semantic base set $\mathscr{H}_D$) in different motions.

probability $P(Y \mid \mathbf{P}^N;\Theta)$, where $\mathbf{P}^N$ is the sequence data with input length N, Y is the sample label, and Θ is the parameter of the model. The model assumes that the hidden state sequence H of the sample, which can be understood as the posture semantic sequence in the sHCRF model, is uncertain, and all hidden state arrangements are considered in the calculation. The computational complexity is $O(Q \times 2^{KN})$, Q is the number of motion categories, K is the dimension of the hidden state, and N is the length of the input sequence. The conditional probability formula calculated by the model is as follows:

$$P\left(Y \mid \mathbf{P}^{N};\Theta\right)=\frac{\sum_{H}e^{\phi\left(y,H,\mathbf{P}^{N};\Theta\right)}}{\sum_{y,H}e^{\phi\left(y,H,\mathbf{P}^{N};\Theta\right)}}. \tag{8}$$

$\mathbf{P}^N$ is the human posture sequence. ϕ is the potential function of the model, and it is defined as

$$\phi\left(y,H,\mathbf{P}^{N};\Theta\right)=\sum_{j}f_{h}\left(\mathbf{h}_{j}\right)\times\theta_{h}\left(y,\mathbf{h}_{j}\right)+\sum_{j}f_{e}\left(e_{j}\right)\times\theta_{e}\left(y,e_{j}\right)+\sum_{j}\mathbf{f}\left(\mathbf{p}_{j}\right)\times\boldsymbol{\theta}\left(\mathbf{h}_{j}=\mathbf{a}\right). \tag{9}$$

f_* is the characteristic function of the model, and both $f_h(\mathbf{h}_j)$ and $f_e(e_j)$ are the one-dimensional integers. $f_h(\mathbf{h}_j)$ is the feature corresponding to the hidden state $\mathbf{h}_j$. $f_e(e_j)$ is the feature corresponding to the jth state transition e_j. $\mathbf{f}(\mathbf{p}_j)$ is a multidimensional vector, which is the feature of the posture of the jth frame $\mathbf{p}_j$. Θ is divided into $\boldsymbol{\theta}, \theta_h, \theta_e$ according to the corresponding characteristics. $\boldsymbol{\theta}$ represents the degree of correlation between the observation node and the hidden state node, which is a multidimensional vector. $\mathbf{a}$ is a current hidden state, and its length is the same as the length of the human posture at the current moment. θ_h is the weight of the corresponding hidden state in a particular category, and θ_e is the weight of the corresponding state transition in a category. The model structure is shown in Figure 5.

In classification of human action, based on the input of certain human action features, the posture semantic sequence of the input is also determined. Equations (8) and (9) can be simplified to Equations (10) and (11), respectively. The computational complexity has been reduced to $O(Q \times N)$.

$$P\left(Y \mid \mathbf{P}^{N};\Theta\right)=\frac{e^{\phi\left(y,H\left(\mathbf{P}^{N}\right);\Theta\right)}}{\sum_{y}e^{\phi\left(y,H\left(\mathbf{P}^{N}\right);\Theta\right)}}, \tag{10}$$

$$\phi\left(y,H\left(\mathbf{P}^{N}\right);\Theta\right)=\sum_{j}\mathbf{f}_{h}\left(\mathbf{h}_{j}\right)\times\boldsymbol{\theta}_{h}\left(y,\mathbf{h}_{\mathbf{j}}\right)+\sum_{j}\mathbf{f}_{e}\left(\mathbf{c}_{j}\right)\times\boldsymbol{\theta}_{e}\left(y,\mathbf{c}_{j}\right). \tag{11}$$

Both $\mathbf{f}_h(\mathbf{h}_j)$ and $\mathbf{f}_e(e_j)$ are the multidimensional vectors. Figure 6 shows the structure of the mHCRF model. The model has a certain hidden state sequence as the input, and there is also a posture change sequence.

The improved potential function does not consider the human body posture sequence or posture change sequence. These two feature sequences carry richer posture information than the hidden state sequence. The lack of this information reduces the effectiveness of a model. Therefore, this paper proposes an sHCRF, in which the shared parameter draws on the "sharing mechanism" of the convolutional network, which not only effectively reduces the number of model parameters but also extracts the corresponding information of the human action sequence under a hidden state (or state transition). The model considers the series of human actions while introducing shared parameters and

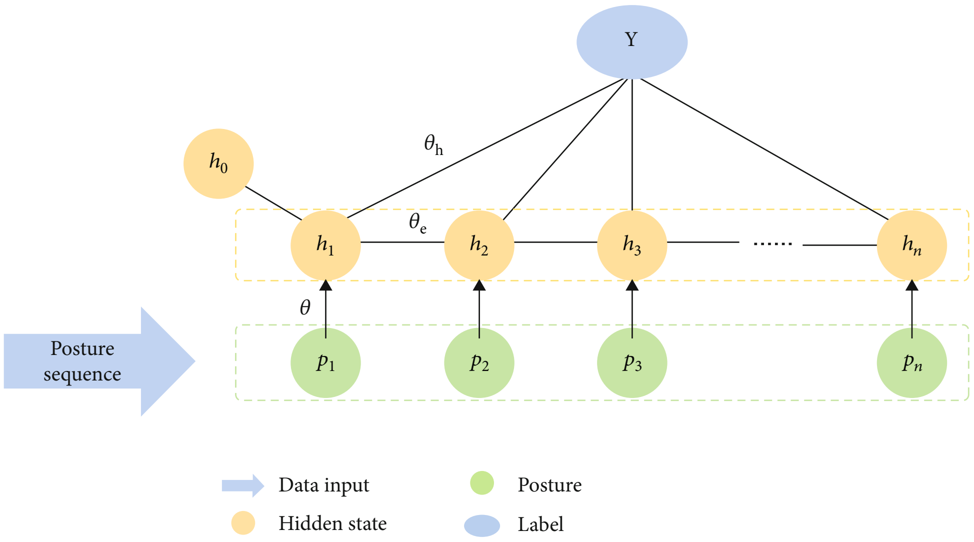


FIGURE 5: HCRF model structure, in which the hidden state sequence considers the combination of all hidden states.

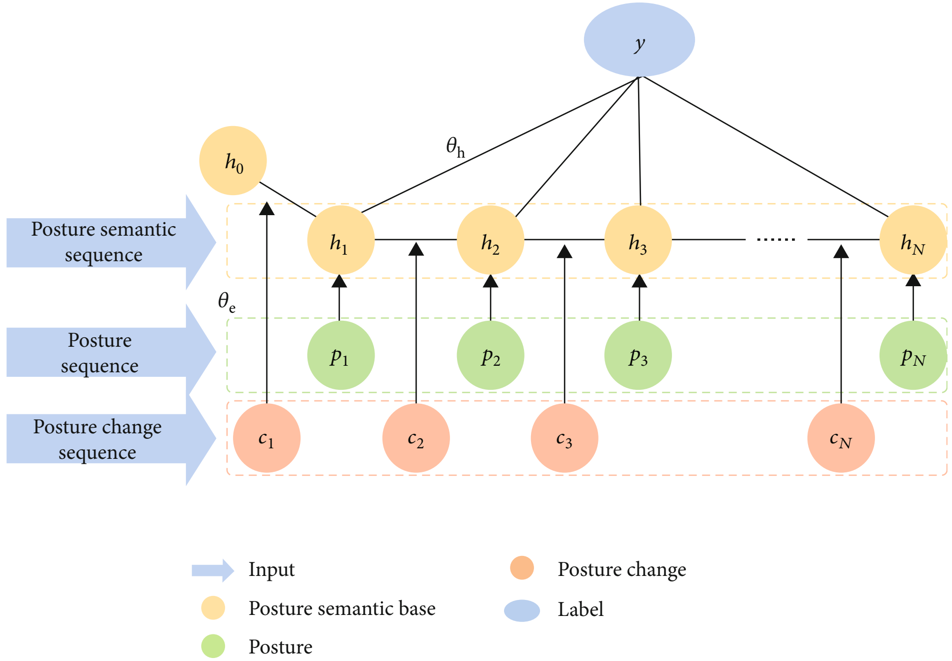


FIGURE 6: mHCRF model structure, in which the hidden state sequence is certain.

the determined hidden state sequence. Equations (10) and (11) are improved as follows:

$$P\left(Y \mid \mathbf{P}^N ; \Theta\right) = \frac{e^{\phi\left(y,H,\mathbf{P}^N;\Theta\right)}}{\sum_y e^{\phi\left(y,H,\mathbf{P}^N;\Theta\right)}}. \tag{12}$$

$\boldsymbol{P}^N$ is the human posture sequence, and ϕ is defined as

$$\begin{aligned}\phi\left(y, H, \mathbf{P}^N ; \Theta\right) = {} & \sum_j \mathbf{f}\left(\mathbf{p}_j\right) \times \boldsymbol{\theta}_{\mathrm{sh}}\left(\mathbf{h}_j = \mathbf{a}\right) \times \theta_h\left(y, \mathbf{h}_j\right) \\ & + \sum_j \mathbf{f}_e\left(\mathbf{c}_j\right) \times \boldsymbol{\theta}_{\mathrm{se}}\left(\mathbf{c}_j = b\right) \times \theta_e\left(y, \mathbf{c}_j\right).\end{aligned} \tag{13}$$

$\mathbf{p}_j$ is a human posture of the jth frame, and $\mathbf{c}_j$ is a posture change between the jth and the $(j+1)$-th frame. $\boldsymbol{\theta}_{\mathrm{sh}}$ and $\boldsymbol{\theta}_{\mathrm{se}}$ are shared parameters from the shared layer, whose function is to extract the semantic information of the input features and compress its size. The lengths of $\boldsymbol{\theta}_{\mathrm{sh}}$ and $\boldsymbol{\theta}_{\mathrm{se}}$ are the same as $\mathbf{f}(\mathbf{p}_j)$ and $\mathbf{f}_e(\mathbf{c}_j)$, respectively. θ_h and θ_e are classification parameters from the classification layer, which are used as weights to determine the category of posture and posture change. The dimension of both parameters is 1. Figure 7 shows the structure of the sHCRF model. There are three types of inputs to the model, posture semantic sequence, posture sequence, and posture change sequence. The function of posture semantic sequence is to extract the

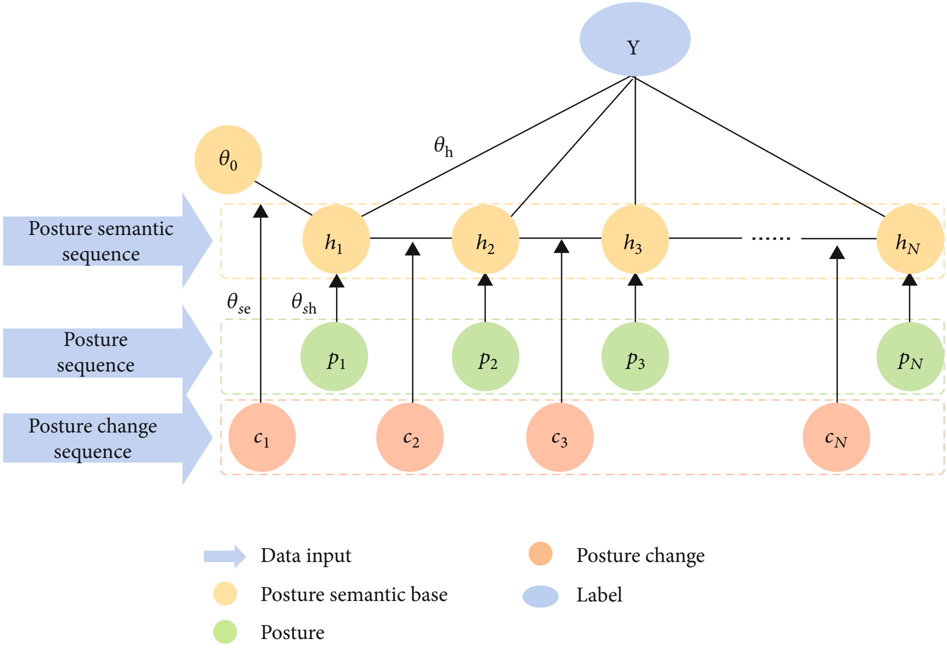


FIGURE 7: sHCRF model structure, in which the shared-classification layer structure is introduced.

information of posture sequence and posture change sequence according to some semantics. The model calculates the probability that the extracted information belongs to a certain category.

4.3. An Incremental Action Classification Based on Knowledge Distillation. To adapt to the continuous input of new categories of action samples, this paper uses the incremental learning method based on the sHCRF model to retain the old knowledge while acquiring new knowledge. In general, it is best to retrain the model by combining old data with new data, but additional storage space is required for the old data. To simulate the learning and memory mechanism of the human brain, the sHCRF model only uses new data for training. It prevents the forgetting of old knowledge without old data by adding distillation loss based on classification loss. Let $\mathscr{L}_C$ be the classification loss function. Let $\mathscr{L}_D$ be the distillation loss function. Let $\mathscr{L}$ be the total loss function, where λ be the custom hyperparameter. Generally, let λ be 1. They are defined as follows:

$$\mathscr{L}_C = y_n \times \log \hat{y}_n, \tag{14}$$

$$\mathscr{L}_D = y_o \times \log \hat{y}_o, \tag{15}$$

$$\mathscr{L} = \mathscr{L}_C + \lambda \mathscr{L}_D + \mathscr{R}. \tag{16}$$

y_n is the label of new data, which uses one-hot encoding [27]. As a hard label, it gives the accurate category of the sample. $\hat{y}_n$, $\hat{y}_o$, and y_o are the vectors of the probabilities of each label. The first two are the calculated results, which are the prediction of the new model on the new data. The last is the prediction of the old model on the new data, which is a soft label. $\mathscr{R}$ is the regular term of the parameter. To adapt to the distillation loss, we made further improvements to the conditional probability function $P(Y \mid X; \Theta, T)$ of the sHCRF model:

$$P(Y \mid X; \Theta, T) = \frac{e^{\phi(y,H,X;\Theta)/T}}{\sum_y e^{\phi(y,H,X;\Theta)/T}}. \tag{17}$$

The newly added temperature coefficient T can control the smoothness of the probability distribution of the output. y_o, $\hat{y}_n$, and $\hat{y}_o$ are defined as follows:

$$y_o = (P(Y_i \mid X; \Theta_{old}, T = t) \mid Y_i \in Y_{old}), \tag{18}$$

$$\hat{y}_n = (P(Y_i \mid X; \Theta_{new}, T = 1) \mid Y_i \in Y_{new}), \tag{19}$$

$$\hat{y}_o = (P(Y_i \mid X; \Theta_{new}, T = t) \mid Y_i \in Y_{old}). \tag{20}$$

y_o and $\hat{y}_o$ are shown in Equations (18) and (20), where the temperature coefficient $T = t$, $t > 1$. $\hat{y}_n$ is shown in Equation (19), and the temperature coefficient $T = 1$. Y_{old} is the label set of the old data, and Y_{new} is the label set of the new data.

The knowledge distillation algorithm of the sHCRF model does not need the old data to participate in the training, and the model can retain the old knowledge while fitting the new data. The new model adds new classification parameters based on the old model. Before training, the old model is used to predict the new human motion data to obtain the soft label of the old knowledge. The temperature coefficient of the prediction function $T = t\ (t > 1)$, then the soft label is used as a kind of "pseudo label." The new data and the correct label are used as the input of the loss function. In the training process, the new model uses the prediction function of temperature coefficient $T = t\ (t > 1)$ and $T = 1$ to predict the new data. The prediction result of the former and the "pseudo label" do cross-entropy to obtain the

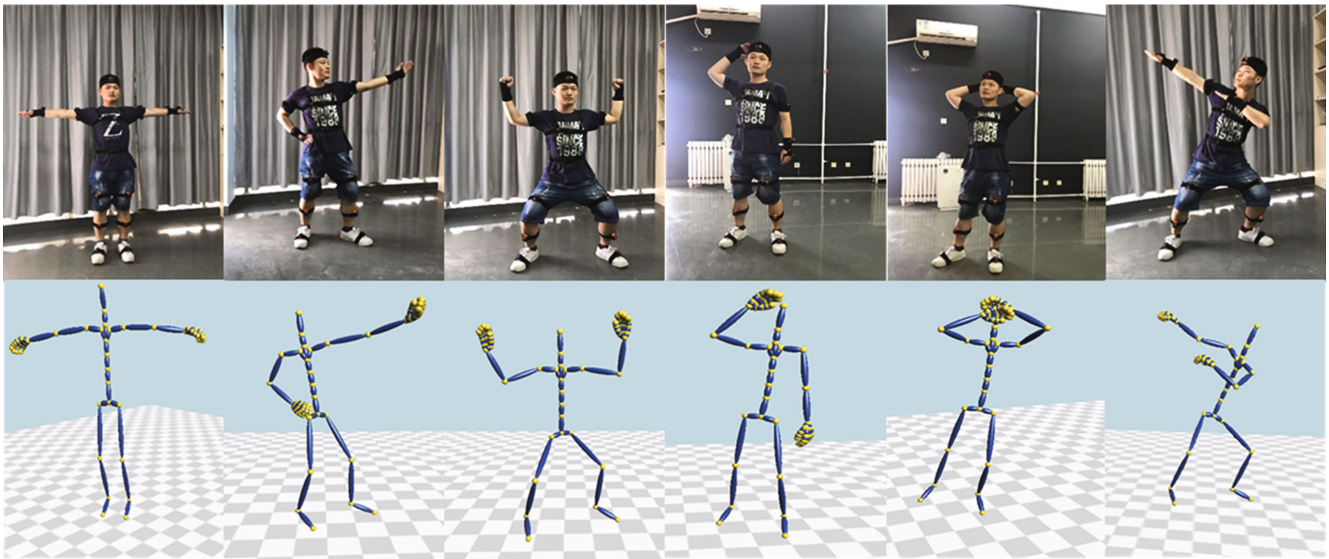


Figure 8: Motion capture and data reconstruction.

model's distillation loss, and the latter's prediction result is compared with the correct label. The cross-entropy obtains the classification loss of the model, and the two and the regular parameter terms constitute the total loss of the model.

5. Experiments

5.1. Human Motion Data Introduction. We captured the human motion data by 17-sensor motion capture equipment developed by our work group and reconstructed the captured information into skeleton data, as shown in Figure 8. The human skeleton comprises 17 joints. All the data samples take the T-Posture as the initial posture, in which the T-Posture shows that the human body is upright and the arms are held flat. Table 2 shows the dataset *Data0*, which contains all the motion data used in the experiment, including Walk, Run, Soccer, Basketball, Jump, Jump Forward, Dance, Yoga, Sit, Pick, Swing, Clean, Yan Fei, and Push-up. The dataset has 14 motion categories. The number of frames is 1,056,405, and the total number of samples is 984.

The datasets listed in Table 3 are subsets of the dataset *Data0* in Table 2. Each subset called *DataX* is divided into old and new parts, representing old data in *DataX_Old* and new data in *DataX_New*. When the datasets in Table 3 display the classes they contain, the class name is represented by a numerical label, which corresponds to the class number in *Data0*. In *Data1*, the old part contains the classes of Walk, Run, and Jump. The new part contains the classes of Soccer, Basketball, Dance, and Swing. In *Data2*, the old part contains the classes of Soccer, Basketball, Dance, and Swing. The new part contains the classes of Walk, Run, and Jump. *Data1* and *Data2* have the same data categories, and *Data2* exchanged old and new data on the basis of *Data1*. In *Data3*, the old part contains the classes of Walk, Run, and Jump. The new part contains the classes of Swing, Yan Fei, and Push-up. The degree of overlap between old and new data in *DataX* is defined by comparing the similarity of pose semantic sets of all action classes in *DataX_Old* and *DataX_New*. It is affected by two factors: one is the proportion of the intersection of posture semantic base sets in the old and new data in the old data, and the second is the proportion of the intersection of the posture semantic base sets of the new and old data in the key posture semantic base in the old data. The high overlap degree in *Data1* shows that more than 90% of posture semantic bases in the old data

Table 2: The whole motion dataset *Data0* in the experiment.

No.	Class name	Frames	Samples
1	Walk	57369	150
2	Run	23082	150
3	Soccer	13228	24
4	Basketball	51764	56
5	Jump	88572	152
6	Jump Forward	31624	72
7	Dance	246308	120
8	Yoga	80632	24
9	Sit	136066	44
10	Pick	32458	68
11	Swing	17528	40
12	Clean	109188	24
13	Yan Fei	93970	28
14	Push-up	64616	32

Table 3: Subsets of *Data0*. These subsets can be divided into *DataX_Old* and *DataX_New* sets, where *X* is the serial number of the subset to which they belong.

DataX (dataset=)	Old (*DataX_Old*=)	New (*DataX_New*=)
Data1	1, 2, 5	3, 4, 7, 11
Data2	3, 4, 7, 11	1, 2, 5
Data3	1, 2, 5	11, 13, 14

appear in the new data. The high overlap degree in *Data2* shows that the intersection of the posture semantic sets of the old and new data coincides with most of the key posture semantic bases in the old data. The overlap degree in *Data3* is less than that in the previous two datasets, indicating that more than half of the posture semantic bases in *DataX_Old* is unique to the old data.

5.2. Experiments. We designed three experiments in this paper. The first explored the sHCRF model's performance, including the model's training efficiency and the classification accuracy of the model. The second explored the memory ability of the sHCRF model. The last explored the energy consumption of the sHCRF model. In each experiment, we selected three-quarters of each type of motion samples as training samples.

The experiment of the sHCRF model's performance: this experiment used the HCRF model and the mHCRF model as the benchmarks to verify the performances of the sHCRF model in classification and training. The input of the HCRF model was only the posture sequence, while both the mHCRF model and the sHCRF model had two types of input. One was included three types of features: posture sequence, posture change sequence, and posture semantic sequence. The other lacked a posture change sequence based on the previous one.

The experiment of the sHCRF model's memory ability: this experiment had three steps. In the first step, we compared the total classification accuracy of the model with the distillation loss using different temperature T. In the second step, we compared the classification accuracy of the model with and without the distillation loss. The experiment was carried out in the datasets with different overlap degrees. We first let the model fit the old data. Then, we initialized the new classification layer parameters corresponding to the new category based on the model fitted to the old data. We used the model to fit the new data with and without distillation loss, respectively. Finally, we used the updated two new models to test on the test set containing all data, respectively. In the third step, we compared the total classification accuracy with and without distillation of the sHCRF model in incremental learning.

The experiment of energy consumption of the sHCRF model: this experiment used the HCRF model and the mHCRF model as the benchmarks to show that the sHCRF model can reduce the energy consumption. The results were presented in the form of energy consumption ratio.

Table 4 shows the settings of hyper parameters in the experience. The first column in the table represents where the hyperparameters are located.

6. Results and Discussion

6.1. Performance of the sHCRF Model. The experiment to verify the model's performance first tested the classification accuracy of the model. Then, we tested the model's training efficiency in the multiclass action classification task, and the model's parameters' amount. In the experiment, the training efficiency was represented by the single iteration time, which had been normalizing. The number of categories in the dataset ranges from 3 to 14 of Table 2. When the number of categories increased, the input samples also increased. Multiple experiments were conducted on each task by using data from random categories in the *Data0* dataset. Since the parameter optimization of the model may achieve local optimization during training, multiple experiments were needed to obtain optimal results. In the comparative experiments with the baseline: HCRF model and mHCRF model, each model's classification accuracy and the single iteration time with different multiclass tasks were compared. Among them, the mHCRF model was divided into two types according to whether its input included posture change sequences, namely, mHCRF(1) and mHCRF(2). The mHCRF(1) indicates that the input of the mHCRF model does not include the posture change sequence, while the input of the mHCRF(2) model includes the posture change sequence. In addition, the posture semantic sequence is used as input in the mHCRF model in this paper, but in the original method, the annotation of HMM is the input. The sHCRF model was divided into two types according to whether its input included posture change sequences, namely, sHCRF(1) and sHCRF(2). The sHCRF(1) indicates that the input of the sHCRF model does not include the posture change sequence, while the input of the sHCRF(2) model includes the posture change sequence.

Table 4: The settings of hyper parameters in the experience.

Parameters' location	Hyper parameters	Value
Equations (1) and (5)	M	13
$\mathbf{h}^K$	K	7
$\mathscr{H}_D$	D	128
T_C	C	16484
Equation (16)	λ	1
Equations (18) and (20)	T	2

Figure 9 shows the classification accuracy of the model in the multiclass action classification task, and Figure 10 shows the single iteration time in the multiclass action classification task. When only considering the influence of input feature, as shown in the results of sHCRF(2), sHCRF(1), mHCRF(2), mHCRF(1), and HCRF in Figures 9 and 10, the structure of the sHCRF(2) model is the same as the mHCRF(2) model. The classification accuracy of the sHCRF and the mHCRF models was higher than that of the HCRF model, the accuracy of the sHCRF and mHCRF models was more than 90%, and the accuracy of HCRF was about 80%. The mHCRF(2) model had the higher classification accuracy than the mHCRF(1), and the accuracy of the mHCRF(2) was more than 95%. The sHCRF(2) model had the higher classification accuracy than the sHCRF(1), and the accuracy of the sHCRF(2) was more than 95%. The single iteration time of the sHCRF(1) is the shortest, followed by the mHCRF(1). The HCRF took the most single iteration time. When only considering the influence of the model's structure, compare the results of sHCRF(2) and mHCRF(2) in Figures 9 and 10. The accuracy of the sHCRF(2) model

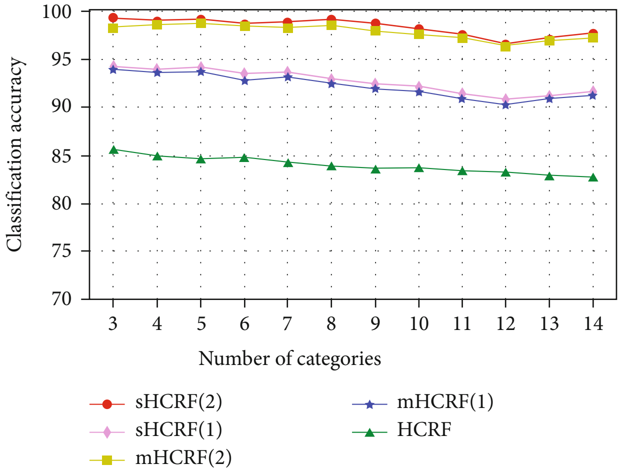


Figure 9: Classification accuracy results of each model, when the sHCRF model and the benchmark models are used to identify tasks on datasets with different numbers of classes.

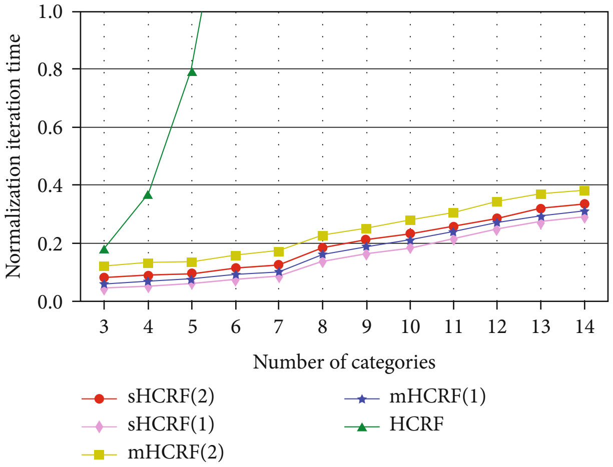


Figure 10: Normalization iteration time of each model, when the sHCRF model and the benchmark models identify tasks on datasets with different numbers of categories.

was higher than that of the mHCRF(2) model. The accuracy of the mHCRF(2) model was slightly inferior. The single iteration time of the sHCRF(2) model was shorter than that of the mHCRF(2) model. Generally, the improvement of input features contributed significantly to the classification accuracy, and the improvement of the model's structure can speed up the single iteration time.

As shown in Figure 9, when the number of categories increased, the classification accuracy of the sHCRF(2) model stayed above 95%. The classification accuracy of the mHCRF(2) model was similar to that of the sHCRF(2) model. The accuracy of the mHCRF(2) model was slightly inferior. The accuracy of the mHCRF(1) model is lower than that of the previous two models. However, the lowest accuracy is still 90%. The accuracy of the mHCRF(1) model was slightly lower than that of the sHCRF(1) model. The classification accuracy of the HCRF model remained above 80%. But as the number of categories increased, the classification accuracy rate gradually decreased. The model maintained a high classification accuracy.

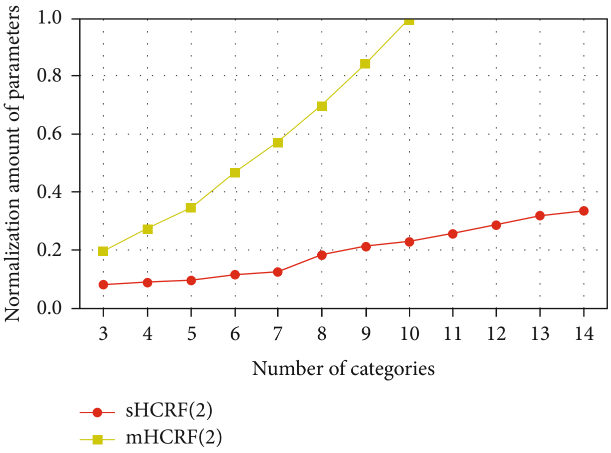


Figure 11: Normalization amount of parameters of each model, when the sHCRF model and the benchmark model identify tasks on datasets with different numbers of categories.

As shown in Figure 10, the single iteration time of the sHCRF(2), sHCRF(1), mHCRF(1), and mHCRF(2) models increased steadily with the increase in the number of categories. The single iteration time of the sHCRF(2) model was shorter than that of the mHCRF(2) model, and the single iteration time of the sHCRF(1) model was shorter than that of the mHCRF(1) model. Unlike the three models, the single iteration time of the HCRF model increases rapidly. The HCRF model spends more time on datasets with more than five categories than is displayed.

As shown in Figure 11, there were the results of the parameters' amount of the sHCRF(2) and mHCRF(2) models. The sHCRF(2) model had the minimum amount of parameters, followed by the mHCRF(2) model. When the number of categories increased, the amount of parameters of the mHCRF(2) model rapidly increased, and its parameters' amount was more than the sHCRF(2) model.

Based on the results of the first two performance indicators obtained from the comprehensive experiment, the sHCRF model and other undirected probability graph models have shown excellent performance in the comparison experiment of the motion classification task. It has good advantages in classification accuracy and training speed.

6.2. Performance of the sHCRF Model in Incremental Learning Scenarios. In the experiment of the performance of the sHCRF model in the incremental learning scenario, the study first tested the classification accuracy of the model with the distillation loss using different temperature T. The result is as shown in Figure 12. In Figure 12, when the value of T is not greater than 5, the old data classification accuracy fluctuates slightly around 80%. The new data classification accuracy is about 90%, and the total accuracy is around 85%. When the value of T increases gradually from 5, the old data classification accuracy began to decline significantly. The same applies to other results, the decline rate of the new data classification accuracy is relatively gentle. When using the distillation loss, the value of T will affect

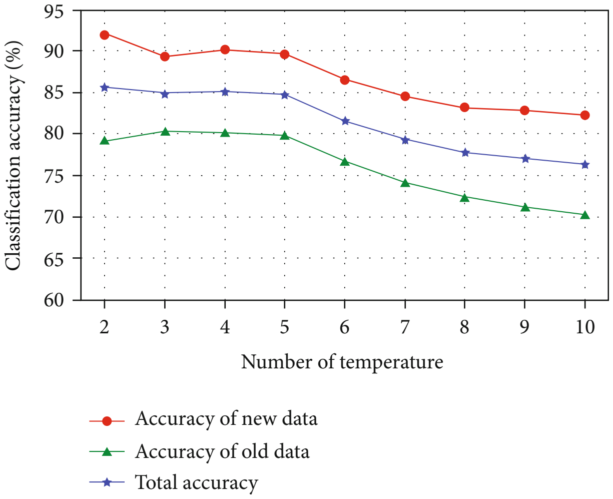


Figure 12: The classification accuracy of the model with distillation loss using different temperature T.

the classification accuracy of the model. The memory ability of old knowledge declines when the value of T exceeds a certain threshold. The threshold is related to the size of the model.

We also verified the classification accuracy of the model with and without distillation loss term, respectively. There were three datasets in the verification experiment, *Data1*, *Data2*, and *Data3*. The classification accuracy of old data is used as the experimental result in Figure 13. In the experiments on *Data1* and *Data2*, the results were 6.9% and 5.8%, respectively. Its ability to retain old knowledge is almost nonexistent. But the effects on *Data3* are not. On *Data3*, the model's classification accuracy of the old data is 33.4%, showing a specific ability to retain old knowledge, as determined by the sHCRF model's structure. When the overlap of *Data3* is low, fitting the new data has little impact on the old knowledge, which means that the parameters related to the old knowledge had not changed much. So, the old knowledge structure of the model was not completely destroyed, and the model has a particular memory ability.

After adding the distillation loss term, the model's memory ability improved further. This experiment was carried out on *Data1*, *Data2*, and *Data3*. The comparison between the classification accuracy of the model on *Data1*, *Data2*, and *Data3* with and without the distillation loss was given, as shown in Figure 13. In Figure 13, the model's classification accuracy of the old data on *Data1* has increased from 6.9% to 86.7%. The old data classification accuracy on *Data2* has increased from 5.8% to 78.8%. The old data classification accuracy on *Data3* has increased from 33.4% to 72.9%. In both *Data1* and *Data2*, the preserving rate of old knowledge of the model has been greatly improved, which effectively inhibits the forgetting of old knowledge by the model. The preserving rate of old knowledge of the model has been improved on *Data3*. The new data classification accuracy on *Data1* is from 98.9% to 94.3%. The new data classification accuracy on *Data2* is from 98.5% to 90.1%. The new data classification accuracy on *Data3* is from 96.0% to 88.0%. The accuracy still maintains high. The model has similar performance in the cognition accuracy of new and old data because of the balance between the model's classification loss and distillation loss. When the overlap degree is high, the distillation loss plays a more significant role in maintaining the old knowledge.

In Table 5, there is the comparison of the total accuracy of both the old and new data with and without distillation. On *Data1*, the total accuracy increased from 52.8% to 90.5% by adding the distillation loss. On *Data2*, the total accuracy is 52.2% without distillation, and it has improved to 84.5% by adding the distillation loss. On *Data3*, the total accuracy increased from 64.7% to 80.5% by adding the distillation loss.

We compared the total classification accuracy with and without distillation of the sHCRF model during incremental learning. The results are shown in Figure 14. We first set up the original model, which could classify three categories. When the categories were incremental, the classification accuracy of the model without distillation rapidly dropped to about 50%. The classification accuracy was around 50% as categories increased since then. The classification accuracy of the model with distillation also declined. The classification accuracy of the model was above 70% as categories increased.

6.3. Energy Consumption Analysis

6.3. Energy Consumption Analysis. We combined the training efficiency and the amount of parameters to estimate the ratio of the energy consumption of the sHCRF model to that of the baseline models. The energy consumption included the computation energy and the storage energy [28]. The computation energy was estimated from the computational complexity of the model, and the storage energy was estimated from the amount of parameters of the model. The experiment consisted of two contrasting experiments, the comparison of the sHCRF model and the mHCRF model, and the comparison of the sHCRF model and the HCRF model. In the experiment, we tested on datasets with different numbers of action categories. The results are shown in Figure 15.

As shown in Figure 15, "S/M" represents the comparison of the sHCRF model and the mHCRF model, and it represents a ratio. In the comparison of the sHCRF model and the mHCRF model, when the number of categories increased, the proportion of the storage energy dropped, and the sHCRF model consumed less storage energy than the mHCRF model. The proportion of the computation energy was less than 1. It means that the sHCRF consumed less computation energy. "S/H" represents the comparison of the sHCRF model and the HCRF model, and it represents a ratio. In the comparison of the sHCRF model and the HCRF model, the sHCRF model consumed less storage energy than the HCRF model. When the number of categories increased, the proportion of the computation energy dropped and almost dropped to 0.

Based on the results of the energy consumption, when the number of categories increased, the single iteration time and the amount of parameters of the sHCRF model were lower than those of the mHCRF model and the HCRF model. It could be estimated that the computation energy

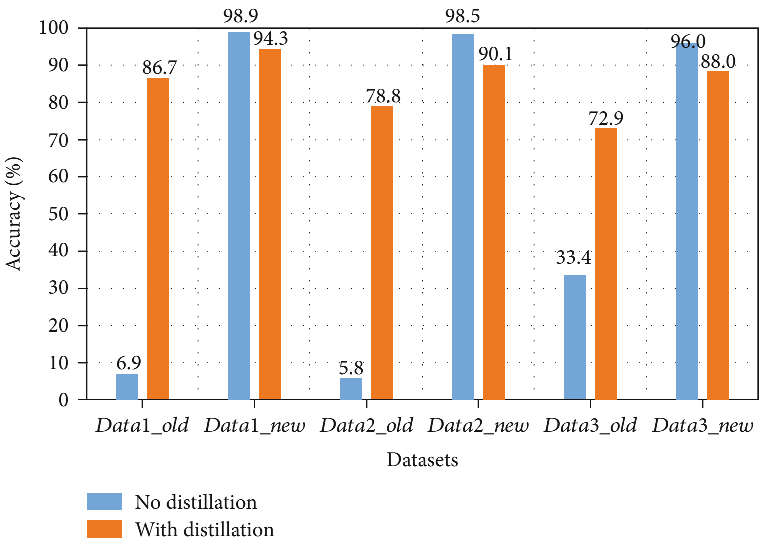


Figure 13: Comparison of the classification accuracy of the new and old data with and without distillation of the sHCRF model.

Table 5: The total accuracy of the sHCRF model of both the old and the new data with and without distillation.

Dataset name	Total accuracy (%)	
	No distillation	With distillation
Data1	52.8	90.5
Data2	52.2	84.5
Data3	64.7	80.5

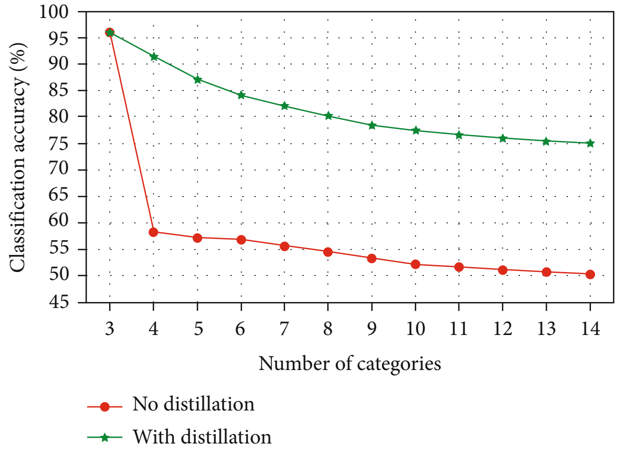


Figure 14: Comparison of the total classification accuracy with and without distillation of the sHCRF model in incremental learning process.

and the storage energy of the sHCRF model were less than those of the mHCRF model and the HCRF model. Combining the computation energy and the storage energy, it can be considered that the sHCRF model consumes less energy. The application system implemented based on the sHCRF model can save the computation and storage energy in terms of model training. The specific energy consumption also depends on the energy consumption of the chips, and related

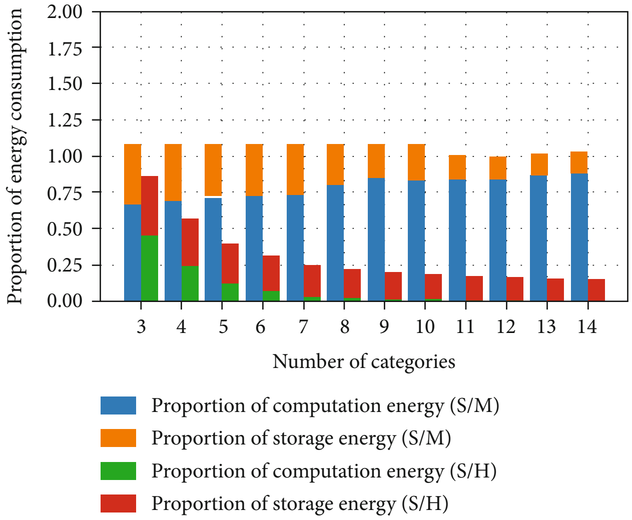


Figure 15: Energy consumption of each model, when the sHCRF model and the benchmark model identify tasks on datasets with different numbers of categories.

hardware. The energy consumption measurement of an application system will be the future work.

7. Conclusions

In this paper, we proposed a lightweight action classification method for Green IoT sport applications. We designed motion features which could describe the spatial and temporal information of the motion data. We proposed a human action classification model, namely, sHCRF, which can be applied to incremental learning scenarios. The model achieves the purpose of energy efficiency by reducing computation overhead and amount of sample data required for training. In general classification tasks, the model's classification accuracy is more than 95%. In the incremental learning scenario, this paper verifies that it can preserve the old knowledge. In the case of category increment, the preserving rate is about 70%. The model's learning ability balances the old and new knowledge. This can effectively control the growth rate of model capacity.

The knowledge distillation method used in this paper depends on the similarity of the training data. If the actions' distinctives are obvious, the effect of the traditional knowledge distillation algorithm is not significant. Further work will improve the model's preservation rate of old knowledge in incremental learning scenarios.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (NSFC, No. 62177005) and by the National Key Research and Development Program of China funded by the Ministry of Science and Technology (No. 2020YFC2007200 and No. 2020YFF0305200).

References

[1] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.

[2] F. Q. Zhang and L. J. Ma, "Multi-object crowd real-time tracking in dynamic environment based on neural network," *Journal of Network Intelligence*, vol. 7, no. 2, pp. 386–394, 2022.

[3] P. Agarwal and M. Alam, "A lightweight deep learning model for human activity recognition on edge devices," *International Conference on Computational Intelligence and Data Science*, vol. 167, pp. 2364–2373, 2020.

[4] S. Rani, H. Babbar, S. Coleman, A. Singh, and H. M. Aljahdali, "An efficient and lightweight deep learning model for human activity recognition using smartphones," *Sensors*, vol. 21, no. 11, p. 3845, 2021.

[5] Y. Jiang, X. Yang, J. Liu, and J. Zhang, "A lightweight hierarchical model with frame-level joints adaptive graph convolution for skeleton-based action recognition," *Security and Communication Networks*, vol. 2021, Article ID 2290304, 13 pages, 2021.

[6] K. Forbes and E. Fiume, "An efficient search algorithm for motion data using weighted PCA," in *Proceedings of the 2005 ACM SIGGRAPH / Eurographics symposium on Computer animation (SCA)*, pp. 67–76, Association for Computing Machinery, New York, NY, USA, 2005.

[7] M. Müller, T. Röder, and M. Clausen, "Efficient content-based retrieval of motion capture data," in *ACM SIGGRAPH 2005 Papers (SIGGRAPH)*, pp. 677–685, Association for Computing Machinery, New York, NY, USA, 2005.

[8] X. Liang, S. Zhang, Q. Li, N. Pronost, W. Geng, and F. Multon, "Intuitive motion retrieval with motion sensors," in *Proceedings of Computer Graphics International*, pp. 64–71, New York, USA, 2008.

[9] H. Ehtesham, "Learning video actions in two stream recurrent neural network," *Pattern Recognition Letters*, vol. 151, pp. 200–208, 2021.

[10] T. Teixeira, E. Granger, and A. Koerich, "Continuous emotion recognition with spatiotemporal convolutional neural networks," *Applied Sciences*, vol. 11, no. 24, article 11738, 2021.

[11] J. Li, X. Liu, M. Zhang, and D. Wang, "Spatio-temporal deformable 3D ConvNets with attention for action recognition," *Pattern Recognition*, vol. 98, pp. 107037–107037, 2020.

[12] J. Shi, C. Liu, C. Ishi, and H. Ishiguro, "Skeleton-based emotion recognition based on two-stream self-attention enhanced spatial-temporal graph convolutional network," *Sensors*, vol. 21, no. 1, pp. 205–216, 2021.

[13] K. Cheng, Y. Zhang, X. He, W. Chen, J. Cheng, and H. Lu, "Skeleton-based action recognition with shift graph convolutional network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 183–192, Seattle, WA , USA, 2020.

[14] D. Tian, Z. Lu, X. Chen, and L. H. Ma, "An attentional spatial temporal graph convolutional network with co-occurrence feature learning for action recognition," *Multimedia Tools and Applications: An International Journal*, vol. 79, no. 17-18, pp. 12679–12697, 2020.

[15] K. Muhammad, A. U. Mustaqeem, A. Ullah et al., "Human action recognition using attention based LSTM network with dilated CNN features," *Future Generation Computer Systems*, vol. 125, pp. 820–830, 2021.

[16] C. Ma, D. Yu, and H. Feng, "Recognition of badminton shot action based on the improved hidden Markov model," *Journal of healthcare engineering*, vol. 2021, Article ID 7892902, 8 pages, 2021.

[17] A. Samr and B. Nizar, "Multimodal action recognition using variational-based Beta-Liouville hidden Markov models," *IET Image Processing*, vol. 14, no. 17, pp. 4785–4794, 2020.

[18] R. Wen, Q. Wang, and Z. Li, "Human hand movement recognition using infinite hidden Markov model based sEMG classification," *Biomedical Signal Processing and Control*, vol. 68, article 102592, 2021.

[19] M. Vrigkas, E. Kazakos, C. Nikou, and I. A. Kakadiaris, "Human activity recognition using robust adaptive privileged probabilistic learning," *Pattern Analysis and Applications*, vol. 24, no. 3, pp. 915–932, 2021.

[20] J. Zhang and S. Gong, "Action categorization with modified hidden conditional random field," *Pattern Recognition*, vol. 43, no. 1, pp. 197–203, 2010.

[21] A. Zaman, F. Yangyu, M. Irfan, M. S. Ayub, L. Guoyun, and L. Shiya, "LifelongGlue: keypoint matching for 3D reconstruction with continual neural networks," *Expert Systems With Applications*, vol. 195, 2022.

[22] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 12, pp. 2935–2947, 2018.

[23] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," 2015, https://arxiv.org/abs/1503.02531.

[24] M. Irfan, Z. Jiangbin, M. Iqbal, Z. Masood, and M. H. Arif, "Knowledge extraction and retention based continual learning by using convolutional autoencoder-based learning classifier system," *Information Sciences*, vol. 591, pp. 287–305, 2022.

[25] J. Kirkpatrick, R. Pascanu, N. Rabinowitz et al., "Overcoming catastrophic forgetting in neural networks," *Proceedings of the National Academy of Sciences*, vol. 114, no. 13, pp. 3521–3526, 2017.

[26] D. Lopez-Paz and M. A. Ranzato, "Gradient episodic memory for continual learning," *Advances in Neural Information Processing Systems*, vol. 30, pp. 6467–6476, 2017.

[27] K. Raajitha, K. Meenakshi, and Y. M. Rao, "Design of thermometer coding and one-hot coding," in *2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV),*, pp. 601–609, Tirunelveli, India, 2021.

[28] T. J. Yang, Y. H. Chen, J. Emer, and V. Sze, "A method to estimate the energy consumption of deep neural networks," in *2017 51st Asilomar Conference on Signals, Systems, and Computers*, pp. 1916–1920, Pacific Grove, CA, USA, 2017.