PAPER

Hierarchical Human Action Recognition with Self-Selection Classifiers via Skeleton Data

Commun. Theor. Phys. **70** (2018) 633–640 Vol. 70, No. 5, November 1, 2018

Hierarchical Human Action Recognition with Self-Selection Classifiers via Skeleton Data*

Ben-Yue Su (苏本跃),[1,2,†] Huang Wu (吴煌),[1,2] Min Sheng (盛敏),[2,3] and Chuan-Sheng Shen (申传胜)[3,‡]

[1]School of Computer and Information, Anqing Normal University, Anqing 246133, China

[2]The Key Laboratory of Intelligent Perception and Computing of Anhui Province, Anqing 246133, China

[3]School of Mathematics and Computational Science, Anqing Normal University, Anqing 246133, China

(Received April 30, 2018; revised manuscript received July 16, 2018)

Abstract *Human action recognition has become one of the most active research topics in human-computer interaction and artificial intelligence, and has attracted much attention. Here, we employ a low-cost optical sensor Kinect to capture the action information of the human skeleton. We then propose a two-level hierarchical human action recognition model with self-selection classifiers via skeleton data. Especially different optimal classifiers are selected by probability voting mechanism and 10 times 10-fold cross validation at different coarse grained levels. Extensive simulations on a well-known open dataset and results demonstrate that our proposed method is efficient in human action recognition, achieving 94.19% the average recognition rate and 95.61% the best rate.*

DOI: 10.1088/0253-6102/70/5/633
Key words: human action recognition, hierarchical architecture model, self-selection classifiers, optimal classification unit

1 Introduction

Human action recognition (HAR) is a hotspot of computer vision.[1] With the popularity of human centered-computing, HAR is of great importance in human-machine interaction, virtual reality, mixed reality, robotics, education, medical treatment, games, intangible cultural heritage and so on.[2] Due to individual diversity in human beings, styles of the same action performed by different persons are usually different. Because of psychological emotions, operating time and other reasons, the same action is performed even for the same person at same time, and the results may also be different.[3] Therefore, it is a huge challenge for HAR to deal with the complexity and variety of human actions. The study of HAR can be traced back to the early work of Johansson.[4] Experiments showed that most of the action can be direct recognized according to the position information of the joint, and the change of the human skeleton position can reflect the information of action. Many of the early research work were based on MoCap,[5−6] namely motion capture, a system of recording the movement of objects or people. However, MoCap needs to manually mark the position of joint point and thus is high-cost.

Subsequently, Kinect a simple and low-cost device, has been paid much attention. Kinect released by Microsoft was originally used as a peripheral to the Microsoft Xbox gaming console to enhance the gaming experience. However, its excellent product experience and advanced somatosensory technology have attracted the interest of academics, and more and more scholars use its technology for research activities. HAR is an important topic of these activities. Recently, research works on Kinect-based human motion recognition have made great progress. Firstly, Li *et al.*[7] proposed a Bag-of-3D-Point human motion recognition algorithm. Yang *et al.*[8] developed a fast HAR algorithm based on skeleton data. A compact human pose representation based on Histograms of 3D Joint Locations (HOJ3D) was reported by Xia *et al.*[9] Afterwards, Ellis *et al.*[10] extracted time series of human gestures from skeleton sequences for action recognition, which includes each posture and the whole motion information. Very recently, Shahroudy *et al.* introduced a large-scale dataset (NTU RGB+D) for HAR,[11] and put forward a Part-aware Long Short-Term Memory (P-LSTM) model, which is more effective than the traditional recurrent neural network. Similarly, an end-to-end two-stream recurrent neural network method was proposed by Wang *et al.*[12] Other methods, such as Refs. [13–15] are also used to study the position information of the skeleton joint points for HAR. However, the data acquired by Kinect is low quality and high noise, since Kinect uses structured light coding technology to acquire the depth data of the image,[16−17] which

*Supported by the National Nature Science Foundation of China under Grant Nos. 11475003, 61603003, and 11471093; the Key Project of Cultivation of Leading Talents in Universities of Anhui Province under Grant No. gxfxZD2016174; Funds of Integration of Cloud Computing and Big Data; Innovation of Science and Technology of Ministry of Education of China under Grant No. 2017A09116; and Anhui Provincial Department of Education Outstanding Top-Notch Talent-Funded Project under Grant No. gxbjZD26

†Corresponding author, E-mail: bysu@aqnu.edu.cn
‡E-mail: csshen@mail.ustc.edu.cn

makes the skeleton data drift. Therefore, it is another huge challenge for HAR using skeleton data.

To overcome the above two challenges, on the one hand, we propose a hierarchical human motion recognition model to coarse grain actions, classifying human actions layer-by-layer, and each layer adopts different kinds of data features with or without time-varying. On the other hand, we design an optimal classification unit by selecting the best classifier based on the training data itself to enhance the recognition rate.

The rest of the paper is organized as follows: We analyze human body structure stratification and propose hierarchical recognition strategy in the next section. In Sec. 3, we introduce the optimal classification unit and describe the function of self selection classifier. Simulation results are presented in Sec. 4, and discussion about the practicability, robustness and extensibility of the method in Sec. 5. At last, the main conclusions are addressed in Sec. 6.

2 Hierarchical Architecture

2.1 *Human Body Structure Stratification*

Human body is a complex system, consisting of nine subsystems, where the human body movement system is an important one. The exercise system consists of three organs: bone, bone connection, and skeletal muscle. Bones are connected in different shapes to form the skeleton to make up the basic frame of the human body. The bone connection, so-called the joint, is a locomotion axis. The skeletal muscle provides power for motion. From the point of view of human movement, bone is passive, while skeletal muscle is active. As a result, the physical information of passive motion and the active EMG signal are main factors in the field of HAR. In the field of computer vision, optical sensors are generally used to acquire the physical information of motion. In common sense, the basic posture of the human body is usually composed of the head, neck, chest, abdomen and limbs.

Human movement can be generally divided into translation, rotation, and compound movement. However, the movement and structure of the human body are dependent. In the following, we will stratify the structure of the human body. The first layer is a whole, and the second layer has five parts. Accordingly, we also separate the human body movement into two layers, corresponding to the whole body movement and the sectional movement respectively. Thus, the structure of human motion system exhibits hierarchical behavior, playing a guiding rule in HAR.

Kinect skeleton tracking technology uses 20 human skeleton joint points to represent a human model. These 20 points are as follows: HipCenter (HC), Spine (S), Shoulder Center (SC), Head (H), Left Shoulder (LS), Left Elbow (LE), Left Wrist (LW), Left Hand (LH), Right Shoulder (RS), Right Elbow (RE), Right Wrist (RW), Right Hand (RH), HipLeft (HL), Left Knee (LK), Left Ankle (LA), Left Foot (LF), HipRight (HR), Right Knee (RK), Right Ankle (RA), and Right Foot (RF). Based on the Kinect skeleton model, we split human body into five parts, and draw its schematic illustration in Fig. 1, where the Part 1 corresponds to the Waist and Head, the Part 2 the Left Arm, the Part 3 the Right Arm, the Part 4 the Left Leg, and the Part 5 the Right Leg.

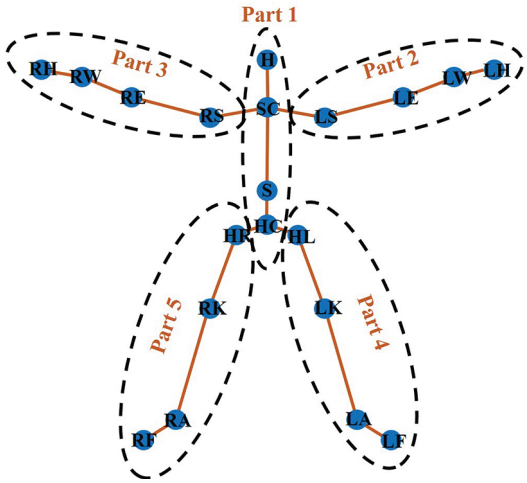

Fig. 1 (Color online) Kinect human model and 5 parts, where Part 1 includes HC, S, SC, and H; Part 2 includes LS, LE, LW, and LH; Part 3 includes RS, RE, RW, and RH; Part 4 includes HL, LK, LA, and LF; Part 5 includes HR, RK, RA, and RF.

2.2 *Hierarchical Recognition Strategy*

According to the human body structure stratification, we propose a hierarchical strategy, which simplifies the complex human body behavior by classifying different kinds of action. Our proposed hierarchical strategy is described in Fig. 2. More specifically, we divide the original categories $[C]^0_{1,1}$ into several major categories $[C]^1_{1,1}, [C]^1_{1,2}, [C]^1_{1,3}, ..., [C]^1_{1,N}$ at the first level. Then, for each major category, we also divide it into several subcategories at the second level and repeat this operation. Finally, we get subcategories that can not be subdivided.

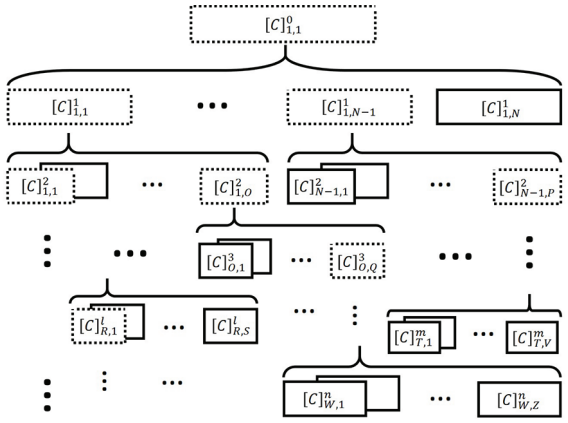

Fig. 2 Hierarchical classification strategy, where $[C]^k_{i,j}$ represents the category of some actions, parameter k denotes the index of layers to which the current action belongs, i indicates the order of its parent category, and j is the local index. Note that the number of sub-categories is different from each other, even in the same level.

Next, we take the following actions, as examples such as waving with right hand, punching with right hand, waving with both hands and punching with both hands, and use two-level classification method to apply our strategy. Firstly, waving with right hand and punching with right hand are coarse-grained as one kind of movement, e.g. actions-with-right-arm, and waving with both hands and punching with both hands are regarded as another kind, e.g. actions-with-both-arms. Secondly, we further classify these two kinds of movement into more visible actions. Note that here the detailed actions of fingers are not considered.

3 Self-Selection Classifiers

3.1 *Optimal Classification Unit*

The general process of HAR is shown in Fig. 3, where feature extraction and classifier construction are the key points. There are two types of feature extraction: one is based on traditional knowledge acquisition and the other is based on deep learning. Obviously the latter has become a research hot topic in recent years, and has received much progress. In this present work, we are interested in the classifier construction.

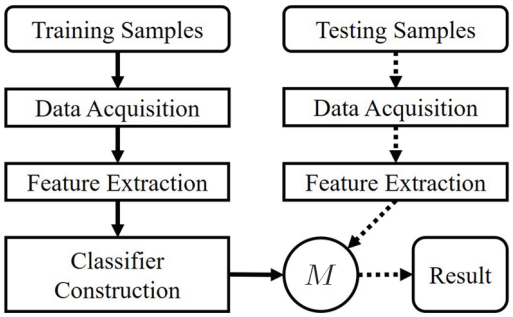

Fig. 3 General process of HAR. M denotes the model learned through training samples.

Traditionally, the classifier construction depends on one's sufficient prior knowledge or experience. It is thus difficult to design an adaptive classifier, which can perform better identification automatically. In this article, we design an optimal classification unit (OCU), as shown in Fig. 4, where the dotted part is the OCU. The main steps are summarized as below:

(i) Giving training data with labels;

(ii) Designing a set of classifiers {C(1),C(2),...,C(N)};

(iii) Inputting the training data to each classifier by cross-validation, and outputting a recognition rate matrix;

(iv) Selecting an optimal classifier C(*) by the voting mechanism (refer for details to the Subsec. 3.2. Probability voting mechanism);

(v) Testing C(*) and outputting the final predict labels.

Furthermore, OCU is also suitable for hierarchical classification. The use of OCU in deep hierarchical classification model is shown in Fig. 5. More specifically, we design $[\mathrm{OCU}]^1$ for the first level classification and use $[\mathrm{OCU}]^2$ for each category at the second level, and so on. Ultimately, we obtain the final results at the last level.

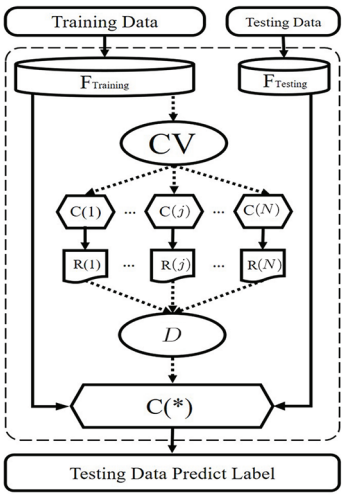

Fig. 4 Schematic illustration of construction of OCU. $\mathrm{F}_{\mathrm{Training}}$ and $\mathrm{F}_{\mathrm{Testing}}$ represent the feature of training data and testing data, respectively. CV denotes the cross-validation method for determining the best classifier, C(1), C(2),C(j), and C(N) denote classifiers, R(1), R(2), R(j), and R(N) denote the recognition results by different classifiers, D is probability voting mechanism for decision-making, and C(*) indicates the optimal classifier.

3.2 *Probability Voting Mechanism*

We start from a recognition rate matrix $N \times M$, where the row number $1, 2, ..., N$ represent the index of classifier, and M denotes m-fold cross-validation, and the voting mechanism of the optimal classifier is shown in Fig. 6. The main idea is as follows: firstly, the probability of each classifier having maximal recognition rate in M tests is calculated (D1), and output the index of the classifier with maximal probability to "Unique?". If output indices, the corresponding classifier will be regarded as the optimal one C(*) and outputs, otherwise, these indices are transferred to D2. Secondly, D2 calculates the average recognition rate of those classifiers given by D1 and outputs the index of the classifier with maximum rate. If outputs indices are unique, the corresponding classifier will be regarded as C(*) and outputs; otherwise, these indices are transferred to D3, and D3 randomly chooses a classifier finally.

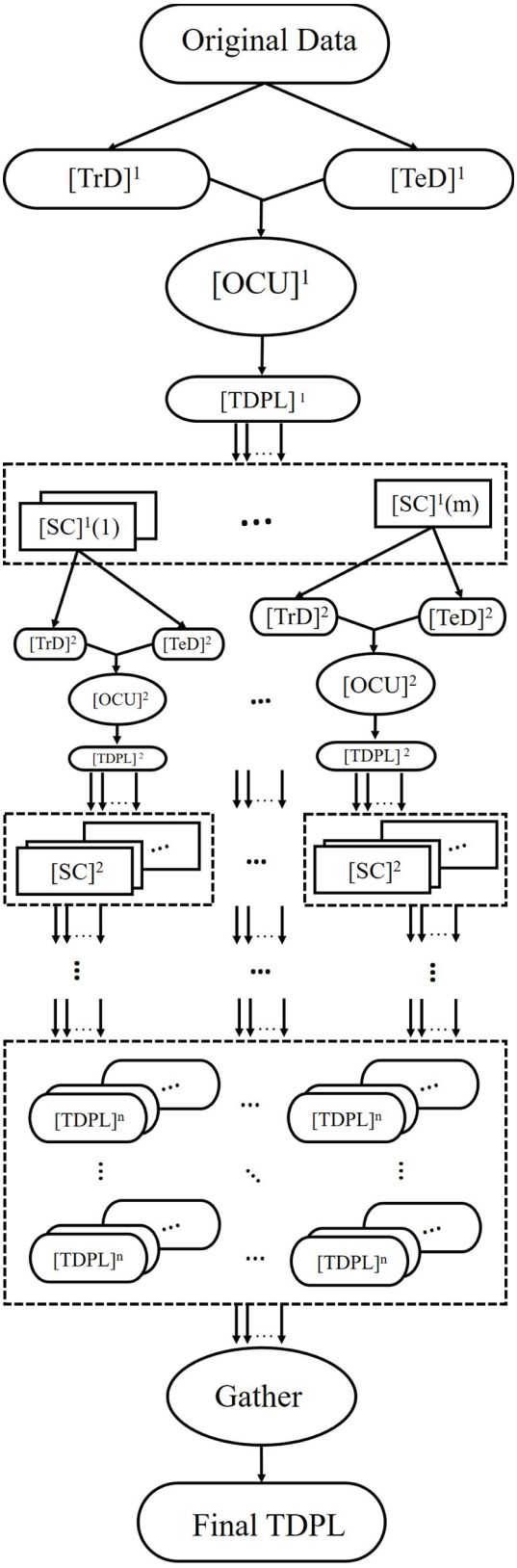

Fig. 5 OCU in deep hierarchical classification model. TrD, TeD, TDPL, and SC are the abbreviation of Training Data, Testing Data, Testing Data Predict Label, and Sub-Category, respectively. The superscript of the symbol "$[]^n$" represents the index of layers. m indicates the number of categories in the local layer.

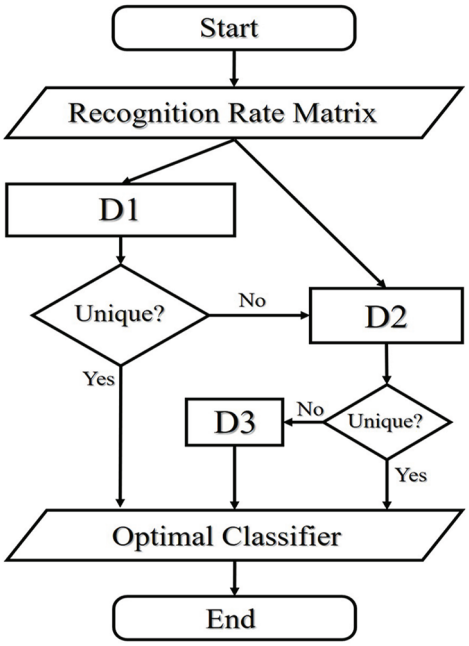

Fig. 6 The voting mechanism of OCU. Step D1 is to obtain the probability of each classifier having maximal recognition rate in M tests, and outputs the index of the classifier with maximal probability, D2 calculates the average rate over M trials of the classifiers offered by D1, and outputs the index of the classifier with maximal average rate, and D3 randomly chooses a classifier given by D2.

4 Simulation

4.1 *Environment*

In this study, the simulation environment includes hardware and software. We use the hardware setup: Intel(R) Core(TM) i5-7500 CPU @ 3.40GHz 3.41GHz, 8.00GB RAM(7.90GB usable), and the software setup: Windows 10 64-bit operating system and *MATLAB* R2017b.

We take the public dataset MSRAction3D (MSRAction3D Skeleton Real3D). There are 20 action types in this dataset, as shown in Fig. 7, each action type consists of 10 subjects, and each subject is performed two or three times. The entire dataset has a total of 567 sample sequences, where 547 effective samples are taken and 20 data samples are missing. At the same time, we test with the whole dataset in subsequent experiments. Since the dataset sample size is not very large, in order to ensure the accuracy, we simulate by using ten times 10-fold cross validation method.

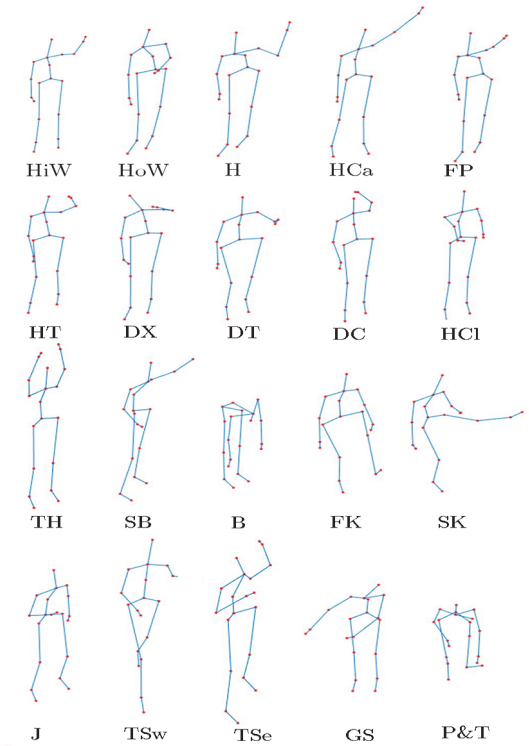

Fig. 7 Actions in MSRAction3D dataset, high arm wave (HiW), horizontal arm wave (HoW), hammer (H), hand catch (HCa), forward punch (FP), high throw (HT), draw x (DX), draw tick (DT), draw circle (DC), hand clap (HCl), two hand wave (TH), side-boxing (SB), bend (B), forward kick (FK), side kick (SK), jogging (J), tennis swing (TSw), tennis serve (TSe), golf swing (GS) and pick up & throw (P&T).

4.2 *Process*

(i) Feature Extraction

Table 1 Each level classification target on MSRAction3D. The abbreviations are the same as Fig. 7.

First-level category	Second-level category
Actions with right arm	HiW, HoW, H, HCa, FP, HT, DX, DT, DC, TSw, SB with single hand
Actions with both arms	HCl, TSe, TH, SB with both hands
Actions with right leg	FK, SK
Actions with waist	B
Actions with arms and legs	J
Actions with waist and both arms	GS
Actions with waist and right arm	P&T

Firstly, we divide the original actions into two-level categories, and each level includes seven kinds of actions, as shown in Table 1. For the first level actions, we employ $(x_i^{(t)}, y_i^{(t)}, z_i^{(t)})$ $(i = 1, 2, 3, \ldots, 20; t = 1, 2, 3, \ldots, T)$ to describe the location of the i-th joint point at frame t, where T is the number of all frames (or time) of each sample. Then, we calculate the mean value $M(x_i)$ and the variance $D(x_i)$ of coordinates for each point through time T characterizing the global features, and Eq. (1) is as follows:

$$M(x_i) = \frac{\sum_{t=1}^{T} x_i^{(t)}}{T},$$
$$D(x_i) = \frac{\sum_{t=1}^{T} (x_i^{(t)} - M(x_i))^2}{T}. \tag{1}$$

Analogously, we can get $M(y_i)$, $M(z_i)$, and $D(y_i)$, $D(z_i)$. The features of the first-level classification are thus described by the following vector $\boldsymbol{F}$:

$$\boldsymbol{F} = (M(x_i), M(y_i), M(z_i), D(x_i), D(y_i), D(z_i)). \tag{2}$$

Since RS (Right Shoulder) is relatively stable compared to other points of right arm motion, we choose it as the origin of coordinates. Note that the end effector can well translate the movement of the robot arm. Here, we define RH as the end-effector denoted by E_3, where the subscript 3 denotes the third part of the human body. Denoting $E_3^{(t)}$ the location of the point RH at time t, the relative position exhibiting spatial information $E_3^{(t)}$ can be given by

$$E_3^{(t)} = P_{\mathrm{RH}}^{(t)} - P_{\mathrm{RS}}^{(t)} = (x_{E_3}^{(t)}, y_{E_3}^{(t)}, z_{E_3}^{(t)}). \tag{3}$$

To better characterize the temporal features of the actions, the speed v can be calculated as follows,

$$v(E_3^{(t)}) = (v(x_{E_3}^{(t)}), \quad v(y_{E_3}^{(t)}), \quad v(z_{E_3}^{(t)})),$$
$$v(x_{E_3}^{(t)}) = x_{E_3}^{(t+1)} - x_{E_3}^{(t)}, \quad v(y_{E_3}^{(t)}) = y_{E_3}^{(t+1)} - y_{E_3}^{(t)},$$
$$v(z_{E_3}^{(t)}) = z_{E_3}^{(t+1)} - z_{E_3}^{(t)}, \tag{4}$$

where $v(x_{E_3}^{(t)})$, $v(x_{E_3}^{(t)})$, and $v(x_{E_3}^{(t)})$ denote the component of axis X, Y, and Z at the current frame $t(t = 1, 2, \ldots, T-1)$, respectively.

Therefore, the characteristics $\boldsymbol{S}$ of the second-level classification can be described as follows:

$$\boldsymbol{S} = (\boldsymbol{S}_{E_1} \oplus \boldsymbol{S}_{E_2} \oplus \boldsymbol{S}_{E_3} \oplus \boldsymbol{S}_{E_4} \oplus \boldsymbol{S}_{E_5}),$$
$$\boldsymbol{S}_{E_j} = (E_j, v(E_j)), \tag{5}$$

where symbols $\oplus$ and $|$ in Eq. (5) denote that $a \oplus b := a$ or b or (a, b), and $a|b := a$ or b respectively.

(ii) Classifier Construction

Human action can be recognized as a typical time-series signal, in which there is a strong correlation between the adjacent frames. Therefore, it is essential for HAR to analyze the change of the action trajectory of the joint points in both time and space. By using different classifiers we classify the data into two types according to their features. One is independent of time-varying, which includes Linear Discriminant Analysis (LDA), K-Nearest Neighbors (KNN), Naive Bayes (NB), C4.5 algorithm

(C4.5), Classification and Regression Trees (CART), Support Vector Machine (SVM), Random Forest (RF), Back-Propagation Neural Network (BPNN) and some Ensemble Algorithm such as Boosting, Bagging and Random Subspace. The other is dependent of time-varying. The classifiers for this type of data usually contain Hidden Markov Model (HMM) and Long Short-Term Memory (LSTM).

In combination with the features extracted from MSRAction3D dataset, we employ the classifiers without and with time-varying to deal with the first-level and second-level optimal classification unit respectively, where $[\mathrm{OCU}]^1$ and $[\mathrm{OCU}]^2$ as shown in Table 2.

Table 2 Classifiers in each level of OCU, where $[\mathrm{OCU}]^1$ and $[\mathrm{OCU}]^2$ represent the first-level and second-level optimal classification unit, respectively.

$[\mathrm{OCU}]^1$	LDA,[18] KNN,[19] NB,[20] C4.5,[21] CART,[22] SVM,[23] RF,[24] BPNN,[25] Boosting,[26] Bagging,[27] Random Subspace[28]
$[\mathrm{OCU}]^2$	HMM,[29] LSTM[30]

(iii) Algorithm Flow Chart

We give the flow chart of our simulation algorithm as shown in Fig. 8. Obviously, the figure shows that our algorithm is based on a two-level hierarchical classification model. On the first level of classification, the original data is divided into training data and testing data according to the 10-fold cross-validation method, and their features $\boldsymbol{F}$ are extracted according to Eqs. (1) and (2). Then, through the self-selection classifiers $[\mathrm{OCU}]^1$, sub-categories (SubC) are obtained. Based on these sub-categories, we match the predict labels with the actual ones. On the second level of classification, the features $\boldsymbol{S}$ of each sub-category are extracted according to Eqs. (3), (4), and (5). By using the optimal classification unit $[\mathrm{OCU}]^2$, we get the second sub-categories, i.e. the final recognition results.

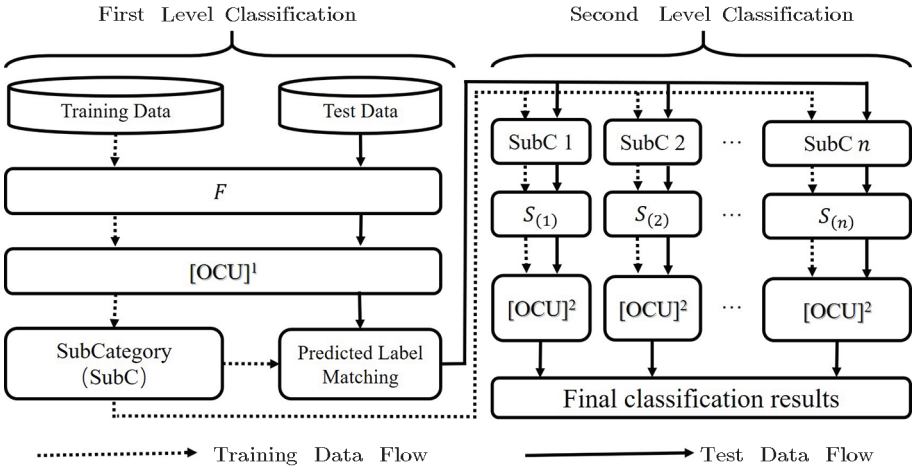

Fig. 8 The two-level classification algorithm flow chart.

4.3 *Result*

We perform ten times simulation tests and obtain the recognition rates, in which the maximum rate is 95.61%, the minimum is 92.87%, and the average is 94.19%. Obviously, our proposed algorithm has high potential for HAR. Furthermore, we give the confusion matrix corresponding to the best recognition result as shown in Table 3. In this matrix, the vertical coordinate output class represents the predict label of an action and the horizontal coordinate target class represents the true label of an action, and the value of each coordinate grid represents the accuracy rate of action predict label recognized as action true label. From the matrix, we can see that most of the actions achieve more than 90% recognition rates, even 6 out of 20 actions achieve 100% accuracy. Note that only one action Hand Catch gets the lowest accuracy, mainly because it is similar to other one-arm actions. Table 3 shows that our algorithm has better recognition ability for most of the actions.

5 Discussion

5.1 *Practicability*

To evaluate the potential of our proposed method, we compare the recognition rates resulted from different approaches by using the MSRAction3D dataset as shown in Table 4. Obviously, the average rate AVG = 94.19, the maximum rate $R_{\max} = 95.61$ and the minimum rate $R_{\min} = 92.87$ of our method are all larger than that of the others, and its variation SD = 0.91 is the smallest, which indicates that our method has acceptable feasibility and accuracy.

Table 3 The confusion matrix of best recognition result.

Hiw	HoW	H	HCa	FP	HT	DX	DT	DC	HCl	TH	SB	B	FK	SK	J	TSw	Tse	GS	P&T
96.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.70	0.00	0.00	0.00
0.00	96.30	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.70	0.00
0.00	0.00	92.60	3.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.70	0.00	0.00
0.00	0.00	3.85	76.91	0.00	3.85	7.69	3.85	3.85	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	3.85	96.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	3.85	0.00	96.15	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	92.60	3.70	3.70	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	96.67	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	96.67	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.33	0.00	0.00	0.00	0.00	6.67	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	93.33	6.67	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00	96.67	0.00	0.00
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	3.33	0.00	0.00	0.00	0.00	0.00	96.67	4.55
0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	4.55	0.00	0.00	0.00	0.00	0.00	0.00	90.90

Table 4 Methods comparison on MSRAction3D dataset, where $R_{\max}$ represents the best recognition rate, $R_{\min}$ represents the worst recognition rate, AVG represents the average recognition rate, and SD represents standard deviation of recognition rates(unit: %)

Methods	$R_{\max}$	$R_{\min}$	AVG	SD
Random occupancy patterns[31]	86.50	–	–	–
Actionlet ensemble[32]	88.20	–	–	–
HON4D + D_{disc}[33]	88.89	–	82.15	4.18
Spatial and temporal part sets[34]	90.22	–	–	–
HOPC of 3D pointclouds[35]	92.39	74.36	86.49	2.28
Points in a Lie group[36]	89.48	–	–	–
Histograms of action poses + DTW[37]	90.56	–	–	–
Dynemes and forward differences[38]	91.94	–	–	–
Part-based feature vector[39]	95.56	74.39	87.05	3.75
Self-selection classifier (Our Method)	95.61	92.87	94.19	0.91

Table 5 Results of control-experiments.

Times	547 samples	567 samples	Error
1	95.61%	93.47%	2.14%
2	93.78%	93.47%	0.31%
3	93.60%	92.95%	0.65%
4	92.87%	92.23%	0.64%
5	94.88%	92.95%	1.93%
6	93.78%	92.95%	0.83%
7	93.24%	91.34%	1.90%
8	95.43%	93.83%	1.60%
9	94.52%	92.06%	2.46%
10	94.15%	91.34%	2.81%
AVG	94.19%	92.66%	1.53%
SD	0.91%	0.88%	0.03%

5.2 *Robustness*

Generally, some data are inevitably missing in the original dataset. For example, we find that there are 20 samples with missing data in the MSRAction3D dataset. Therefore, we design two sets of simulations with and without missing data, and the results are shown in Table 5. We find that the sample missing data cause the minute reduction of the recognition rate, but the change is very small, and there was no bug and no termination in the proceeding of the program, which suggests that our proposed method has higher robustness for recognizing human actions with fewer missing data.

5.3 *Extensibility*

Our approach has an extensibility including horizontal expansion and vertical extension. Here, so-called horizontal expansion, we indicate that the number and types of

classifiers in OCU can be increased or modified according to the different scenario. In this study, we adopt a two-level model of HAR, and select in each level of OCU different classifiers. Specifically, the employed classifiers are shown as given in Table 2. Generally, we define the classification of human actions according to the structure of the body, and the number of hierarchial layers usually depends on the coarse-grained level. The more coarse-grained the fewer the number of layers is. Therefore, we determine the number of layers on the basis of the actual situation. This is so-called vertical extension. To be specific, we design a two-level hierarchical model in the present work.

6 Conclusion

In conclusion, we have proposed a two-level hierarchical HAR model with self-selection classifiers, where the classifiers without and with time-varying are employed to deal with the first-level and the second-level optimal classification unit, respectively. Then, we have extracted the mean and variance of discrete time series to characterize the first-level coarse actions, and calculated the location and speed of the end-effector to distinguish the fine actions in the second level. We have applied the method to a public dataset, and achieved an average recognition rate 94.19% and the best rate 95.61%, which suggests that our proposed method is efficient in HAR, and hierarchical recognition strategy can better explain the internal mechanism of human action. However, how to recognize the human actions with large scale dataset and deal with the systems with a great number of missing data problems? This open question surely deserves further investigations and may be the content of a future presentation.

References

[1] J. K. Aggarwal, ACM Computing Surveys 43 (2011) 16.

[2] J. Zhang, W. Li, P. O. Ogunbona, *et al.*, Pattern Recogn. **60** (2016) 86.

[3] L. Chen, H. Wei, and J. Ferryman, Pattern Recognition Lett. **34** (2013) 1995.

[4] G. Johansson, Scintific American **232** (1975) 76.

[5] M. Müller, T. Röder, and M. Clausen, Acm Trans. Graph. **24** (2005) 677.

[6] A. W. Vieira, T. Lewiner, W. R. Schwartz, *et al.*, Proceedings of the 21st International Conference on Pattern Recognition (ICPR2012), (2012) 2934.

[7] W. Li, Z. Zhang, and Z. Liu, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Workshops, (2010) 9.

[8] X. Yang and Y. L. Tian, Proceedings of Computer Vision and Pattern Recognition, (2012) 14.

[9] L. Xia, C. C. Chen, and J. K. Aggarwal, IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, (2012) 20.

[10] C. Ellis, S. Z. Masood, M. F. Tappen, *et al.*, Int. J. Comput. Vision **101** (2013) 420.

[11] A. Shahroudy, J. Liu, T. T. Ng, *et al.*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2016) 1010.

[12] H. Wang and L. Wang, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2017) 3633.

[13] Y. Y. Lin, J. H. Hua, N. C. Tang, *et al.*, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2014) 2617.

[14] Y. Du, W. Wang, and L. Wang, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2015) 1110.

[15] T. T. Thanh, F. Chen, K. Kotani, *et al.*, Fund. Inform. **130** (2014) 247.

[16] Z. Zhang, *Microsoft Kinect Sensor and Its Effect*, IEEE Computer Society Press, Washington (2012).

[17] G. T. Papadopoulos, A. Axenopoulos, and P. Daras, Proceedings of the International Conference on Multimedia Modeling (2014) 473.

[18] T. Hastie and R. Tibshirani, J. Roy. Stat. Soc. B **58** (1996) 155.

[19] M. L. Zhang and Z. H. Zhou, Pattern Recogn. **40** (2007) 2038.

[20] I. Rish, J. Univers. Comput. Sci. **1** (2001) 127.

[21] J. R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann, San Francisco, CA (1993).

[22] L. Breiman, J. Friedman, C. J. Stone, *et al.*, CRC Press, Boca Raton (1984).

[23] M. A. Hearst, S. T. Dumais, E. Osuna, *et al.*, IEEE Intelligent Systems and Their Applications **13** (1998) 18.

[24] A. Liaw and M. Wiener, R News **2** (2002) 18.

[25] M. C. Mozer, Complex Systems **3** (1989) 349.

[26] Y. Freund, R. Schapire, and N. Abe, Journal of Japanese Society For Artificial Intelligence **14** (1999) 1612.

[27] L. Breiman, Mach. Learn. **24** (1996) 123.

[28] T. K. Ho, IEEE Transactions on Pattern Analysis and Machine Intelligence **20** (1998) 832.

[29] S. R. Eddy, Curr. Opin. Struct. Biol. **6** (1996) 361.

[30] S. Hochreiter and J. Schmidhuber, Neural Comput. **9** (1997) 1735.

[31] J. Wang, Z. Liu, J. Chorowski, *et al.*, Proceedings of European Conference on Computer Vision (2012) 872.

[32] J. Wang, Z. Liu, Y. Wu, *et al.*, The IEEE Conference on Computer Vision and Pattern Recognition, (2012) 1290.

[33] O. Oreifej and Z. Liu, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2013) 716.

[34] C. Wang, Y. Wang, and A. L. Yuille, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2013) 915.

[35] H. Rahmani, A. Mahmood, D. Q. Huynh, *et al.*, Proceedings of European Conference on Computer Vision (2014) 742.

[36] R. Vemulapalli, F. Arrate, and R. Chellappa, The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), (2014) 588.

[37] M. Barnachon, S. Bouakaz, B. Boufama, *et al.*, Pattern Recogn. **47** (2014) 238.

[38] I. Kapsouras and N. Nikolaidis, J. Vis. Commun. Image R. **25** (2014) 1432.

[39] H. Chen, G. Wang, J. H. Xue, *et al.*, Pattern Recogn. **55** (2016) 148.