ORIGINAL ARTICLE

Human posture recognition based on multiple features and rule learning

Weili Ding[1] · Bo Hu[1] · Han Liu[2] · Xinming Wang[1] · Xiangsheng Huang[3]

Abstract

The use of skeleton data for human posture recognition is a key research topic in the human-computer interaction field. To improve the accuracy of human posture recognition, a new algorithm based on multiple features and rule learning is proposed in this paper. Firstly, a 219-dimensional vector that includes angle features and distance features is defined. Specifically, the angle and distance features are defined in terms of the local relationship between joints and the global spatial location of joints. Then, during human posture classification, the rule learning method is used together with the Bagging and random subspace methods to create different samples and features for improved classification performance of sub-classifiers for different samples. Finally, the performance of our proposed algorithm is evaluated on four human posture datasets. The experimental results show that our algorithm can recognize many kinds of human postures effectively, and the results obtained by the rule-based learning method are of higher interpretability than those by traditional machine learning methods and CNNs.

Keywords Human posture recognition · Multiple features · Rule learning

✉ Han Liu
liuh48@cardiff.ac.uk

✉ Xinming Wang
wang8297161@stumail.ysu.edu.cn

Weili Ding
weiye51@ysu.edu.cn

Bo Hu
jessical@stumail.ysu.edu.cn

Xiangsheng Huang
xiangsheng.huang@ia.ac.cn

[1] Department of Automation, Institute of Electrical Engineering, Key Laboratory of Intelligent Rehabilitation and Neromodulation of Hebei Province, Yanshan University, 438 West of Hebei Avenue, Haigang District, Qinghuangdao 066004, China

[2] School of Computer Science and Informatics, Cardiff University, Queen's Buildings, 5 The Parade, Cardiff CF24 3AA, UK

[3] Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

1 Introduction

In recent years, the use of skeleton data for human posture recognition has emerged as a popular research topic in the computer vision field. This technology shows good prospects for application in human-computer interaction, rehabilitation medicine, multimedia applications, virtual reality, robot control, and others. In general, postures are different from actions, with the former being static and the latter dynamic. A human posture is a base of actions, and is often taken as the key frame in various action recognition algorithms. Moreover, in some fields, such as physical training, rehabilitation training [8] and sign language communication, a human posture is more important than an action. In noisy workshops and dangerous working environments, posture recognition, as a human-computer interaction mode, is much superior to keystroke control and voice interaction in that it is more accurate, efficient and more natural in interaction.

There are several main methods for posture recognition. One is to use wearable sensors [39], such as wearing accelerometer [2, 3, 16] and pressure sensor [11]. However, wearing such a device makes subjects feel a sense of burden, which compromises the interactive experience. The other one is based on monocular cameras [35]. However, it is susceptible to illumination and background interference, offering

unsatisfactory recognition accuracy and robustness in complex conditions. With the increasingly low cost depth image sensors, RGB-D image based posture and action recognition has become an important research focus in the field of human-computer interaction. Researchers can obtain color and depth images as well as skeleton data of human easily. Many posture recognition algorithms [6, 22] that use skeleton data obtained from Kinect are proposed. These algorithms can not only avoid the influence of illumination, but they also eliminate the need of preprocessing such as segmentation and object detection in complex backgrounds, which enables greatly improved accuracy. However, most of the existing works are focused on the action recognition rather than the posture recognition, with more and more attention being paid to daily actions. Additionally, datasets and algorithms based on posture recognition are still of limited availability. Therefore, in this paper we propose a human posture recognition method, which incorporates several datasets that contain a lot of postures while achieving more accurate posture recognition.

The contributions of this paper are that we extract features at different granular levels and create diverse training subsets for enhanced accuracy in the rule-based classifier. Specifically, to better represent human postures, (1): we extract angle features between joints in the fine-grained level and relative distance features between key body parts in the coarse-grained level. (2): in the classification stage, bagging and random subspace approaches are used to divide the original training dataset into subsets with different samples and features. The final decision is made by voting RIPPER classifiers that are trained on these diverse training subsets. The experimental results show that our algorithm performs better than CNNs for the current datasets even using the same parameters.

The rest of this paper is organized as follows. A review of related work is offered in Sect. 2. The algorithm of human posture is described in Sect. 3. A description of the datasets and the experimental results are provided in Sect. 4. The conclusions are given in Sect. 5.

2 Related works

Most of the traditional posture recognition methods describe human visual information and two-dimensional posture information by extracting features from RGB images. Ramanan and Sminchisescu [36] proposed an algorithm that uses human contour samples to obtain human edge templates and a similarity and gradient descent method to estimate postures. Jiang et al. [18] presented a posture recognition method using convex programming based matching schemes. This method proves to be more efficient than other methods such as the graph-cut or belief propagation methods

for the object matching problem in which a large searching range is involved. However, these methods are sensitive to some unnecessary features extracted from people's clothes, environment interference and illumination in the image.

Souto and Musse [38] proposed an algorithm that uses artificial neural networks to automatically detect human poses in a single image. But this approach uses static image features to determine human skeleton, which requires a large amount of computation to extract features. Mun Wai and Isaac [34] have presented a technique of data-driven MCMC technique to estimate 3D human poses from static images. For pose estimation of three-dimensional human, Sarafianos et al. [37] reviewed the progresses and shortcomings of recent researches on the estimation of 3D human poses. Considering that different input modes and different key features are introduced separately, they conducted an extensive experimental evaluation on the approaches in a synthetic dataset. At the end of the paper, they discussed the findings from the literature review and the experimental results.

Since the advent of the Microsoft Kinect sensor in 2010, more and more researchers have begun developing posture recognition methods based on skeleton data and depth images. Lin et al. [25] proposed a Kinect-based rehabilitation system, which defines two kinds of features, namely, the average distance between 10 joint points of the upper limb and the angle features of 9 adjacent joints compared with the posture to be recognized. The recognition result of the method depends on the setting of the matching threshold, so the robustness is less than ideal. Islam et al. [17] used a Kinect sensor to detect different joint points of human body and further to calculate the average deviation to recognize yoga poses for users. Miranda et al. [33] presented a method that uses the angle between skeletal joints to describe the human postures and a multi-level support vector machine (SVM) and a decision forest are used to classify them. The method, however, offers limited accuracy when recognizing multiple similar postures. Li et al. [22] used angular features to represent six human postures and SVM to classify them. Chen and Wang [6] proposed a method that uses the back propagation (BP) network, SVM, naive bayes to recognize three postures. This method involves no feature extraction and uses the original skeleton data as the input data to the classifier.

Agarwal and Triggs [1] proposed a relevance vector machine (RVM) regression method that employs contour information to estimate human postures. This method requires matching with multiple templates and is therefore time-consuming. Zainordin et al. [41] proposed a method to classify postures by setting the threshold distance, angle between joints, and establishing a set of rules based on the skeleton and depth information. However, this method is only suitable for classifying a few postures due to the reliance of its recognition accuracy on the posture kernel

formulation training. Georgakopoulos et al. [13] proposed a method that can automatically recognize any user-defined postures. Nine features which represent specific body parts are generated from the user's posture skeleton information. The features are input into SVM to generate attitude learning models to recognize postures. Elforaici et al. [9] proposed a method in which convolutional features are extracted from color images and transfer learning is involved to train convolutional neural networks (CNNs) for recognizing human postures from RGB and depth images. Li et al. [23] proposed a method that uses the anthropometry and the BP neural network to recognize human postures with the person oriented to the Kinect sensor in different directions. The deep learning method exhibits relatively good recognition rates, but it is difficult to interpret the resulting mode. The method also requires very large data sets and time-consuming parameter regulation work to achieve high performance.

To sum up, most of the existing works are image-based methods. As such, we propose a posture recognition algorithm for the skeletal information obtained by Kinect.

3 Proposed approach

The proposed approach for human posture recognition is based on the skeleton information extracted from a Kinect sensor. Figure 1 illustrates the stages involved in this approach. First, multiple features were defined, including the angle features and the distance features between joints. Then bagging and random subspace methods were used to create rule ensembles based on the RIPPER rule learning algorithm, which allowed training 100 rule sets that make up a rule ensemble for final classification by majority voting.

3.1 Extraction of multiple features

The Kinect sensor can acquire real-time 3D position information of 20 human joints, which can be expressed in x, y and z coordinates in meters. In the original data, each posture is recorded as the absolute position of 20 joints of human body, the skeleton information is denoted as $J = \{j_1, j_2, j_3, \ldots, j_N\}$, where, $j_i = (x_i, y_i, z_i)$ refers to the coordinate position of joint i, and N = 20 is the total number of skeleton joints. The label of each joint is defined as shown in Fig. 2.

Any two joints form one skeleton segment. As shown in Table 1, a total of 23 skeletal segments are defined as $S_i = \{S_1, S_2, \ldots, S_{23}\}$. Each skeletal segment S_i consists of two joint points in the
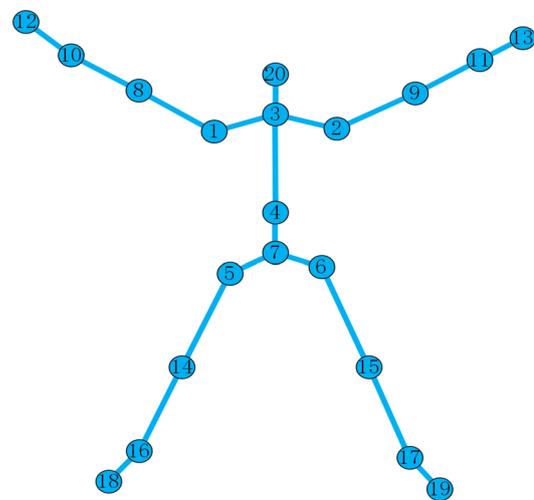

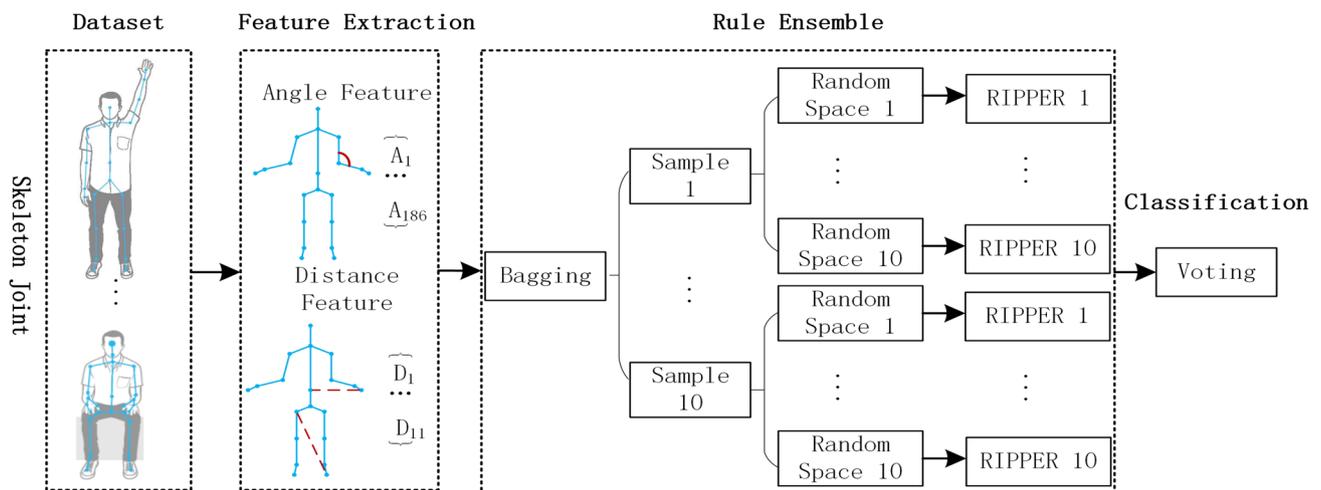**Fig. 2** The label of each joint points

Fig. 1 Overview of the proposed approach

Table 1 Composition of skeletal segments

S_i	Joint point	S_i	Joint point	S_i	Joint point
S_1	$\{j_3, j_{20}\}$	S_9	$\{j_{12}, j_{10}\}$	S_{17}	$\{j_{17}, j_{15}\}$
S_2	$\{j_3, j_1\}$	S_{10}	$\{j_{13}, j_{11}\}$	S_{18}	$\{j_{18}, j_{16}\}$
S_3	$\{j_3, j_2\}$	S_{11}	$\{j_{10}, j_1\}$	S_{19}	$\{j_{19}, j_{17}\}$
S_4	$\{j_1, j_2\}$	S_{12}	$\{j_{11}, j_2\}$	S_{20}	$\{j_{16}, j_5\}$
S_5	$\{j_8, j_1\}$	S_{13}	$\{j_7, j_3\}$	S_{21}	$\{j_{17}, j_6\}$
S_6	$\{j_9, j_2\}$	S_{14}	$\{j_{14}, j_5\}$	S_{22}	$\{j_5, j_6\}$
S_7	$\{j_8, j_{10}\}$	S_{15}	$\{j_{15}, j_6\}$	S_{23}	$\{j_{18}, j_{19}\}$
S_8	$\{j_9, j_{11}\}$	S_{16}	$\{j_{16}, j_{14}\}$		

table, where the spatial coordinates are expressed as: $j_a = (x_a, y_a, z_a), a = 1, 2, \ldots, 20, j_b = (x_b, y_b, z_b), b = 1, 2, \ldots, 20, b \neq a$

Then the direction vector of the linear equation of skeletal segment S_i is denoted as follows:

$$v_i(v_x, v_y, v_z) = (x_b - x_a, y_b - y_a, z_b - z_a) \tag{1}$$

Thus the angle between the two skeletal segments S_a and S_b is defined as:

$$Angle = arcos \frac{(v_{xa} * v_{xb} + v_{ya} * v_{yb} + v_{za} * v_{zb})}{\sqrt{(v_{xa}^2 + v_{ya}^2 + v_{za}^2) * (v_{xb}^2 + v_{yb}^2 + v_{zb}^2)}} \tag{2}$$

Here the direction vector of S_a is $v_a(v_{xa}, v_{ya}, v_{za})$, and the direction vector of S_b is $v_b(v_{xb}, v_{yb}, v_{zb})$.

In this study, 253 angular values were obtained from the defined angle between two skeletal segments. After removal of 67 redundant angles, 186 angle features were extracted finally. We define them as: $Angle_i = [Angle_1, Angle_2, \ldots, Angle_{186}]$. Here, the angle features of three-dimensional space are rotation and scale invariant, and they play an important role in the recognition process.

Next, we define the relative distance features of 11 groups of joint points, which are shown in Table 2. Here, the distance feature $D_i = \{d_{ix}, d_{iy}, d_{iz}\}, i \in [1, 11]$, where, $d_{ix} = (x_a - x_b); d_{iy} = (y_a - y_b); d_{iz} = (z_a - z_b)$. The distance features represent the global human posture, as a complement to angular features.

Finally, a 219-dimensional feature vector $f_i = \{f_1, f_2, \ldots, f_{219}\}$ is generated, which includes 186 angular features and 33 distance features. The angle features can

describe the relationship between two skeletal segments as well as local human postures. The distance features of human posture show the relative distance between the joint points, which can roughly describe the movement of limbs. The combination of angle features and distance features permits more comprehensive representation of postures.

3.2 Classification method

As mentioned in Sect. 1, the classification process entails the Bagging approach, the random subspace method, and the RIPPER rule learning algorithm for creating rule ensembles.

The Bagging approach (stands for bootstrap aggregating), which was proposed by [4], is used here to draw n different versions of training data through random sampling with replacement. In this way, some instances may be selected more than once into the new training sample s_i, whereas some other instances may never be selected. On average, each sample s_i is expected to represent 63.2% of the instances in the original training set [21, 26, 27]. This indicates that the base classifiers trained (using the same learning algorithm) on the n samples are likely to be diverse [5, 19], because the n samples cover different parts of the original training set. The procedure of the Bagging approach is illustrated in Fig. 3.

The random subspace method, which was proposed by [15], is used here to create diversity among m feature subsets. Since each feature subset fs_j represents a random subspace of the full feature set, which leads to the diversity among the randomly selected feature subsets, the m base classifiers trained on the m feature subsets are more likely to be diverse [5, 19]. The random subspace method was originally used as an effective way of creating decision tree ensembles and its resulting models are referred to as random decision forests [14]. The random subspace method involves a similar procedure to the Bagging approach, as shown in Fig. 3. In the sampling stage, however, features instead of instances are selected. Hence, the random subspace method is also known as feature bagging.

The RIPPER algorithm, which was proposed by [7], is aimed at training rule-based classifiers through the separate-and-conquer strategy of rule learning [12] as illustrated in Algorithm 1.

Table 2 Composition of distance feature D

D_i	D_1	D_2	D_3	D_4	D_5	D_6	D_7	D_8	D_9	D_{10}	D_{11}
j_a	j_{12}	j_{12}	j_{12}	j_{13}	j_{13}	j_{13}	j_{19}	j_{18}	j_{20}	j_{20}	j_{20}
j_b	j_{20}	j_4	j_6	j_{20}	j_4	j_5	j_5	j_6	j_4	j_5	j_6

Fig. 3 The procedure of bagging [29]
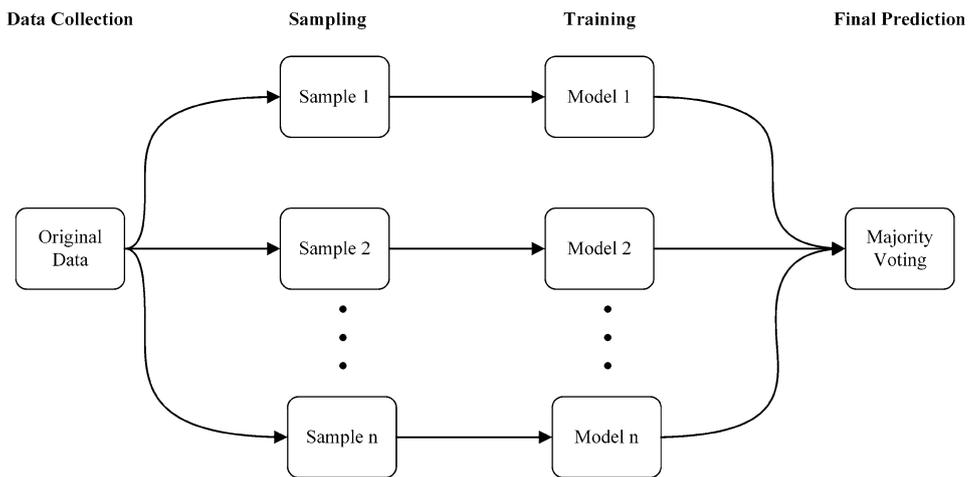
Algorithm 1: Rule learning algorithm (Kononenko and Kukar, 2007)

1 **Input:** A set of training sample T
2 **Output:** An ordered set of rules RS
3 **while** $T \neq \varphi$ **do**
4   generate a single rule R from the training set T
5   delete all instances covered by rule R
6   **if** *the generated rule R is not good* **then**
7   generate the majority rule and empty the training set T
8   **end**
9   add rule R into the rule set RS
10 **end**

At each iteration of learning a single rule (shown in line 2 of the algorithmic procedure illustrated in Algorithm 1), an attribute-value pair (e.g. $x_1 > 2$) that can maximize the rule quality is selected as a condition (an antecedent of the rule), and the process is repeated until the stopping criterion of learning this rule is satisfied. Once the rule has been finalized following the above process, the rule would normally have covered the same class of training instances. In this case, the learning of the above rule is finished. It is then required to find all the instances that are covered by this rule and delete these instances from the training set, in order to initiate the learning of the next rule from the remaining instances.

For the RIPPER algorithm, the selection of an attribute-value pair at each iteration of learning a single rule is made by evaluating the rule quality [based on the FOIL information gain shown in Eq. (3)] after adding an attribute-value pair as an antecedent of this rule,

$$IG_{r_i} = p_{r_i} \times \left(log_2 \left(\frac{p_{r_i}}{p_{r_i} + n_{r_i}} \right) - log_2 \left(\frac{p}{p+n} \right) \right) \quad (3)$$

where p_{r_i} and n_{r_i} represent, respectively, the number of positive and negative instances covered by rule r_i, whereas p and n represent, respectively, the number of positive and negative instances in the initial training subset from which the learning of rule r_i starts.

On the other hand, the RIPPER algorithm also requires pruning of each rule r_i once the learning of the rule r_i is complete before the learning of the next rule can start. In particular, incremental reduced error pruning (IREP) is adopted to simplifying each rule r_i, based on the rule-value metric shown in Eq. (4).

$$w_{r_i} = \frac{p_{r_i} - n_{r_i}}{p_{r_i} + n_{r_i}} \quad (4)$$

IREP is designed to prune each rule by starting from evaluating the last antecedent of rule r_i in terms of the rule-value metric w_{r_i}. If the value of w_{r_i} increases after removal of the last antecedent of rule r_i, the above pruning process is repeated until the value of w_{r_i} decreases. In other words, if the value of w_{r_i} does not increase after removal of the last antecedent of rule r_i, the pruning process should be stopped immediately and the last antecedent of rule r_i should not be removed.

Once a whole set of rules have been trained, a global optimization stage is involved to further enhance the quality of the rule set. More details about how the RIPPER algorithm works for the whole rule learning and pruning procedure can be found in [7].

The whole framework of training classifiers is designed to involve three levels. Level 1 is to create n samples of training data through the Bagging approach; level 2 is to create m feature subsets based on each of the n training samples, using the random subspace method; and level 3 is to train a base classifier based on each of the $m \times n$ feature subsets, using the RIPPER algorithm. The final classification is made

Fig. 4 MSR-action3D posture depth image

Fig. 5 MSRC-12 posture RGB image

Fig. 6 UTKinect action posture RGB image

Fig. 7 Baduanjin action posture RGB image

by fusing the outputs of the $m \times n$ base classifiers through majority voting.

4 Experiments

In this section, the data sets used for this study are described alongside the details on the experimental setup. Moreover, the experimental results are discussed in a comparative way.

4.1 Datasets

We have performed an extensive evaluation on our proposed method using four datasets. The first three datasets were extracted from the public action databases MSR-Action3D, Microsoft MSRC-12, and UTKinect-Action. The fourth dataset, called "Baduanjin posture", was built by ourselves using the Kinect sensor.

The MSR-Action3D dataset [24] was collected from 20 actions: high arm wave, horizontal arm wave, hammer, hand catch, forward punch, high throw, draw x, draw tick, draw circle, hand clap, two hand wave, side-boxing, bend, forward kick, side kick, jogging, tennis swing, tennis serve, golf swing, pick up and throw. There are 10 subjects, and each subject performs each action 2 or 3 times. We extract 20 postures from the MSR-Action3D dataset to build the MSR-Action3D posture dataset, which consists of 3224 frames. As shown in Fig. 4, the first posture is highly similar to the fourth and sixth ones, and the thirteenth one is highly similar to the twentieth one, which causes difficulties in posture recognition.

The second posture dataset used in this paper was established from the Microsoft MSRC-12 dataset of Research Cambridge [10]. It was collected from 30 people who performed 12 gestures. We extracted 5884 frames from 719359 frame action samples, and built a new posture dataset. Figure 5 shows the 12 postures.

The third posture dataset was extracted from the UTKinect-Action dataset [40]. We chose 10 action types of this dataset: walk, sit down, stand up, pick up, carry, throw, push, pull, wave hands and clap hands. There were 10 subjects, and a total of 3795 frames were extracted. This dataset was collected to investigate variations in different views: right view, frontal view, left view and back view. In addition, the background clutter and human-object interactions in some postures add new challenges to posture recognition. Figure 6 shows the 10 postures.
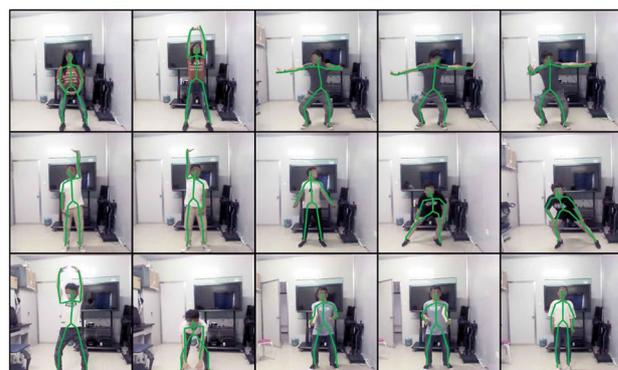
We have also collected a new dataset of rehabilitation postures. It is called Baduanjin dataset, which is collected in accordance with the standard operating procedures. Baduanjin is a traditional method of fitness, which is often used to improve the physical constitution, balance and joint flexibility of patients with motor dysfunction in China. We defined 15 types of postures, and collected them using a Kinect sensor. In our test, each action was performed by 10 subjects. Figure 7 shows the 10 postures.

4.2 Experimental setup and results

This experiment was established on the KNIME Analysis Platform, which allowed easier integration of algorithms and more convenient manipulation or visualization of data. We used the Bagging node (a part of the Weka plugin), where

Table 3 Comparison methods and parameters setting

		Baduanjin	MSR3D	UTKA	MSRC
	Support vector machines	C = 2	C = 5	C = 3	C = 5
		L = 0.001	L = 0.001	L = 0.001	L = 0.001
		P = 1.0E−12	P = 1.0E−12	P = 1.0E−12	P = 1.0E−12
	k = polynomial kernel				
	Fuzzy ruler learning	The fuzzy norm = min/max norm			
		the shrink function = volume border			
	Decision trees	The usual default Gini index = default			
		min number records per node = 2			
	K-nearest neighbor	K = 1			

the size of each bag (a percentage of the training data size) was set to 100, and the button of calculating the out of bag was set as false. The number of iterations was set to 10, i.e., the Bagging approach was used to draw 10 training samples, and the Random Subspace method was used to draw 10 feature subsets on each of the 10 training samples, by setting the size of each subspace to 0.5. The RIPPER algorithm was used for training 10 base classifiers (rule sets) on the 10 feature subsets drawn from each training sample, where the RIPPER algorithm was set to involve 2 runs of rule optimization and using 1/3 of the training data for rule pruning. Therefore, the adoption of the whole framework for ensemble creation (based on Bagging, Random Subspace and RIPPER) produced 100 base classifiers in total. All the algorithms were tested using the 10-fold cross-validation method. The proposed method was compared with five common classification methods and convolutional neural networks.

We have performed parameter selection for SVM and KNN by cross-validation [20]. The optimized parameters for SVM and KNN and other three common classification algorithms are listed in Table 3. We have also conducted experiments using these five algorithms with default settings of parameters. For SVM, C is the complexity constant, L is the tolerance parameter, P is the epsilon for round-off error and K is polynomial kernel. For KNN, K is the number of nearest neighbors used in classification. According to Table 3, the selected parameters for the SVM algorithm are not the same for different datasets while those for KNN are the same.

We use PCA and wrapper-based feature selection to reduce the feature dimensionality. The results are shown in Table 4. According to Table 4, using our method without feature selection achieves the best accuracy for the 4 datasets. Feature dimensionality is significantly reduced to a range between 34 and 53 using PCA, but the accuracy is dropped slightly. With the combination of genetic search and ZeroR classifier, the number of features decreased dramatically to 10, but the performance obtained using this feature selection method also declined. As a matter of the fact, in the

Table 4 The accuracy and the number of selected features of our method after feature selection

Dataset	PCA	ZeroR classifier	without feature selection
Baduanjin	98.3%/34	97.8%/10	99.6%/219
MSR3D	88.2%/42	76.7%/10	91.5%/219
UTKA	97.2%/53	91%/10	98.1%/219
MSRC	96.6%/43	90.8%/10	97.6%/219

Table 5 Comparison of recognition accuracy on the MSR-Action3D posture dataset

Method	Accuracy (%)
Support vector machines	72.5
Fuzzy ruler learning	85.9
Decision trees	85.4
K-nearest neighbor	88.9
Our method	94.5

stage of random subspace, the original feature set is divided into diverse subsets with lower feature dimensionality, which means that the feature dimensionality is reduced even though there is no feature selection involved. Thus the following experiments are all conducted using 219-dimensional feature vectors.

In the experiment, we have used the angle features and the distance features proposed in this paper for posture recognition. As shown in the MSR-Action 3D posture dataset diagram in Fig. 4, the dataset contains several groups of similar postures, such as the first posture extracted from the waving action and the sixth posture extracted from the high-throw action; the second, fourth and twelfth postures obtained from the horizontal sliding, grabbing and side stroke are also similar among the three groups of postures. These similar postures cause great difficulties in posture recognition. As shown in Table 5, this algorithm has higher recognition rates than the other five algorithms using default settings

Fig. 8 Contrast matrix between our proposed algorithm and the SVM algorithm
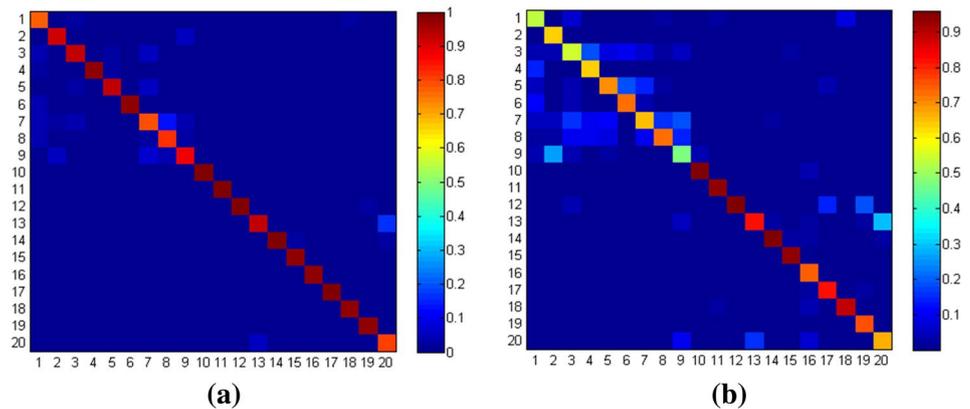

(a)   **(b)**

Table 6 Comparison of recognition accuracy on the MSRC-12 posture dataset

Method	Accuracy (%)
Support vector machines	88.5
Fuzzy ruler learning	95.8
decision trees	93.7
K-nearest neighbor	97.5
Our method	97.6

Table 8 Comparison of recognition accuracy on the Baduanjin posture dataset

Method	Accuracy (%)
Support vector machines	93.2
Fuzzy ruler learning	97.3
Decision trees	99.1
K-nearest neighbor	98.1
Our method	99.6

Table 7 Comparison of recognition accuracy on the UTKinect-Action posture dataset

Method	Accuracy (%)
Support vector machines	80.1
Fuzzy ruler learning	94.2
Decision trees	95.0
K-nearest neighbor	97.6
Our method	98.1

of parameters. The classification confusion matrix of the proposed algorithm and the SVM algorithm in Fig. 8a, b demonstrates that the proposed algorithm performs better in classifying similar postures in the dataset than the SVM algorithm.

The experimental results on the posture data set obtained from the MSRC-12 action dataset are given in Table 6. The dataset also contained some similar postures, such as posture 4 for using telescopes and posture 7 for shooting, posture 10 for head-holding and posture 12 for air hitting. The experimental results show that this method outperforms other contrast algorithms using default settings of parameters in terms of recognition accuracy. Posture-like classification also works well.

The recognition results obtained using the UTK-Action posture data set as the training set are shown in Table 7.

This algorithm also produces better recognition results than other algorithms using default settings of parameters. As indicated in Fig. 6, the collection environment of the dataset is complex. The angle and distance features obtained from the skeleton data used for the algorithm are not affected by the environment background, exhibiting a higher level of robustness.

The recognition results obtained using each algorithm on the Baduanjin posture dataset we built here are shown in Table 8. It is also superior to the other five classification algorithms using default settings of parameters in the recognition accuracy of 15 rehabilitation postures. Additionally, our proposed algorithm is based on rule learning. Therefore, the classification model obtained by using the algorithm is an ensemble of rule sets (consisting of rules). Compared with many machine learning and deep learning methods, our proposed algorithm can generate better interpretable models. The KNIME platform can output the model generated by the rule learning algorithm for posture recognition to text, which consists of the feature subset selected by random subspace and the classification rule set of each base classifier. These visible rule sets can be used to recognize different rehabilitation postures, showing more promising applications in rehabilitation than other algorithms.

Accuracy produced by SVM and KNN using optimized parameters in Table 3 on four datasets is shown in Table 9. It is worth noticing that the parameter selection process

Table 9 Accuracy of SVM and KNN using optimized parameters on four datasets

Method	Baduanjin (%)	MSR3D (%)	UTKA (%)	MSRC (%)
SVM	99.9	94.3	99.1	97.8
KNN	100	94.5	99.3	98.6

Table 10 Comparison between the results of our method and those of CNN

Method	Baduanjin (%)	MSR3D (%)	UTKA (%)	MSRC (%)
Lenet	96.2	87.2	95.3	96.5
Alexnet	98.1	86.7	92.4	96.7
VGG-13	95.6	83.4	91.7	95.2
Ours	99.6	94.5	98.1	97.6

Table 11 AUC values of different methods

Method	Baduanjin	MSR3D	UTKA	MSRC
SVM	0.9999	0.9963	0.9986	0.9973
KNN	1	0.9779	0.9963	0.9921
Fuzzy rule	0.9914	0.9532	0.9769	0.9808
Decision tree	0.996	0.9376	0.9754	0.9746
Lenet	0.996	0.9788	0.9823	0.9967
Alexnet	0.9975	0.9591	0.9916	0.9956
VGG-13	0.9989	0.9873	0.9949	0.9981
Ours	1	0.9979	0.9991	0.9996

Table 12 AP values of different methods

Method	Baduanjin	MSR3D	UTKA	MSRC
SVM	0.9994	0.957	0.9929	0.982
KNN	1	0.9513	0.9911	0.9824
Fuzzy rule	0.9744	0.82	0.9467	0.9565
Decision tree	0.993	0.77	0.9349	0.9272
Lenet	0.9726	0.8538	0.8998	0.9779
Alexnet	0.9714	0.753	0.935	0.9548
VGG-13	0.9909	0.8821	0.9692	0.9866
Ours	0.9999	0.9734	0.9959	0.9966

for SVM and KNN is time-consuming. Compared with SVM and KNN, even though the results of these classifiers using optimized parameters are merely better than ours, our algorithm achieves ideal results on all the four datasets. Our method has a stronger generalization ability when it comes to sharing the same parameters for different datasets, which make our algorithm more robust [42]. More importantly, with its base classifier being rule-based, our method can output rules for each posture, making it quite useful and convenient in real-world applications.

According to Table 10, our method tops all in terms of recognition accuracy in the four datasets. In the Baduanjin and MSRC datasets, these four methods all have a recognition accuracy rate above 95%. Alexnet exhibits higher accuracy rates than the other two CNNs but still lower ones than ours. In the MSR3D dataset, our method is the only one that has an accuracy rate of over 90%. In the UTKA dataset, our method has a 6.4% higher recognition accuracy rate than VGG-13. A main reason for this is that the features we extracted contain different granular-level information. Specifically, we extract a 219-dimensional feature vector which consists of 186 angle features (fine-grained level) and 33 distance features (coarse-grained level). Therefore, the features we extracted can capture both the local and global information of different postures.

We saved the results of each fold in 10-fold classification and used the micro-average method in sklearn toolbox to generate the precision and recall (PR) curves and ROC curves. The ROC curves and the PR curves for different datasets are shown in Figs. 9 and 10. The AUC values and AP values are shown in Tables 11 and 12. Our method shows better performance than CNNs and for AUC and AP values, our method is at the top for all the datasets.

5 Conclusion

In this paper, we have proposed a rule ensemble approach for human posture recognition based on multiple features. The approach employs the Bagging approach for random sampling of training data and the Random Subspace method for random selection of feature subsets. This allows diverse rule-based classifiers to be trained using the
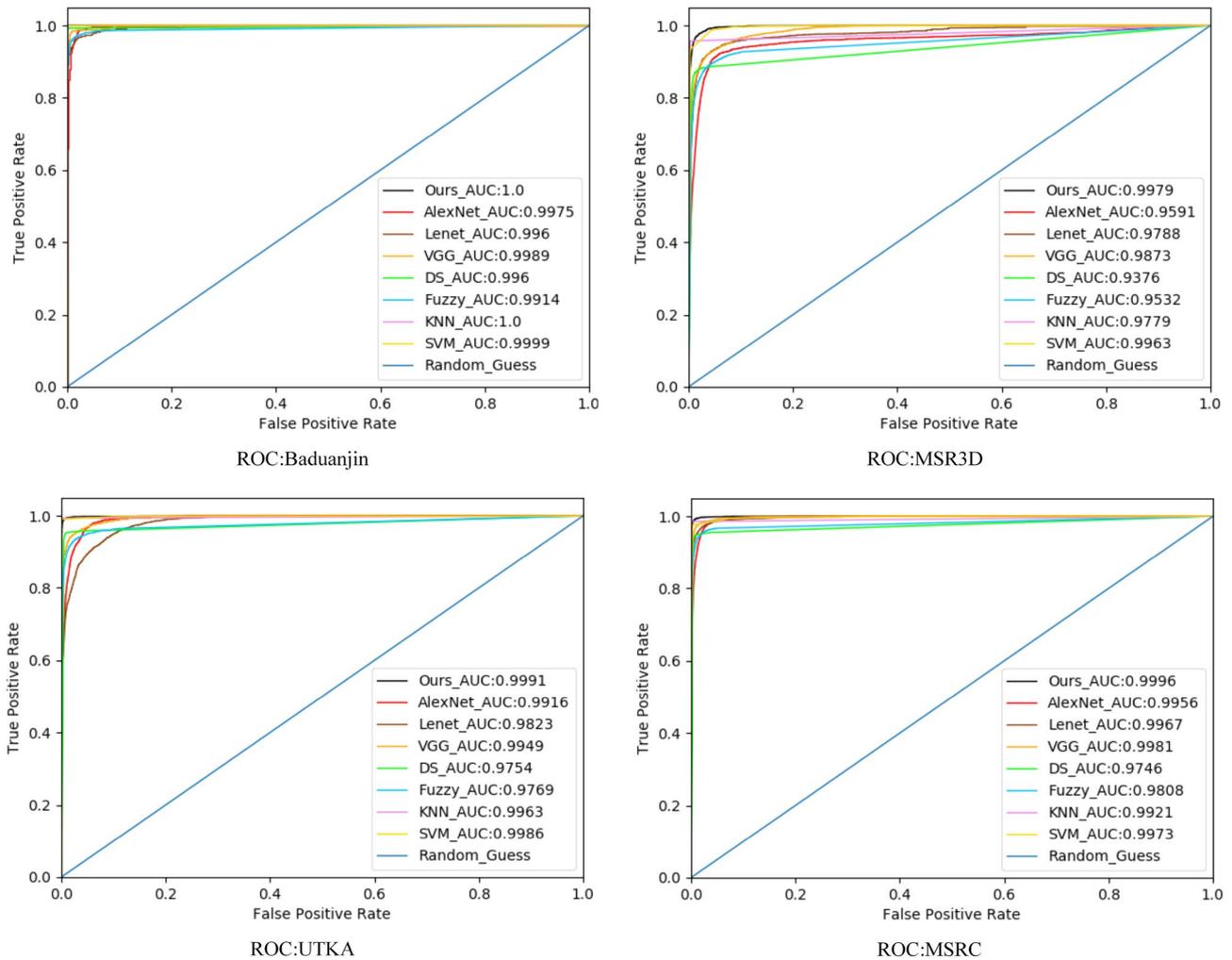
Fig. 9 ROC curves of different methods

RIPPER rule learning algorithm and thus create a high-performance ensemble. In terms of feature extraction, we managed to extract multiple features, which include angel features and distance features between joints. A comparison was made between our proposed approach and five popular learning methods using three public action data sets and one that was built by ourselves. The experimental results show that our proposed approach outperforms the other learning methods.

In the future, we will investigate the techniques of granular computing [28, 30–32] towards extraction of features at multiple levels of granularity and fusion of different features to reduce the dimensionality and the sparsity of feature sets. It is also critical to explore how the extraction of multiple features can increase the diversity among classifiers trained using different feature sets or learning algorithms, so as to enable further advances in the performance of human posture recognition.
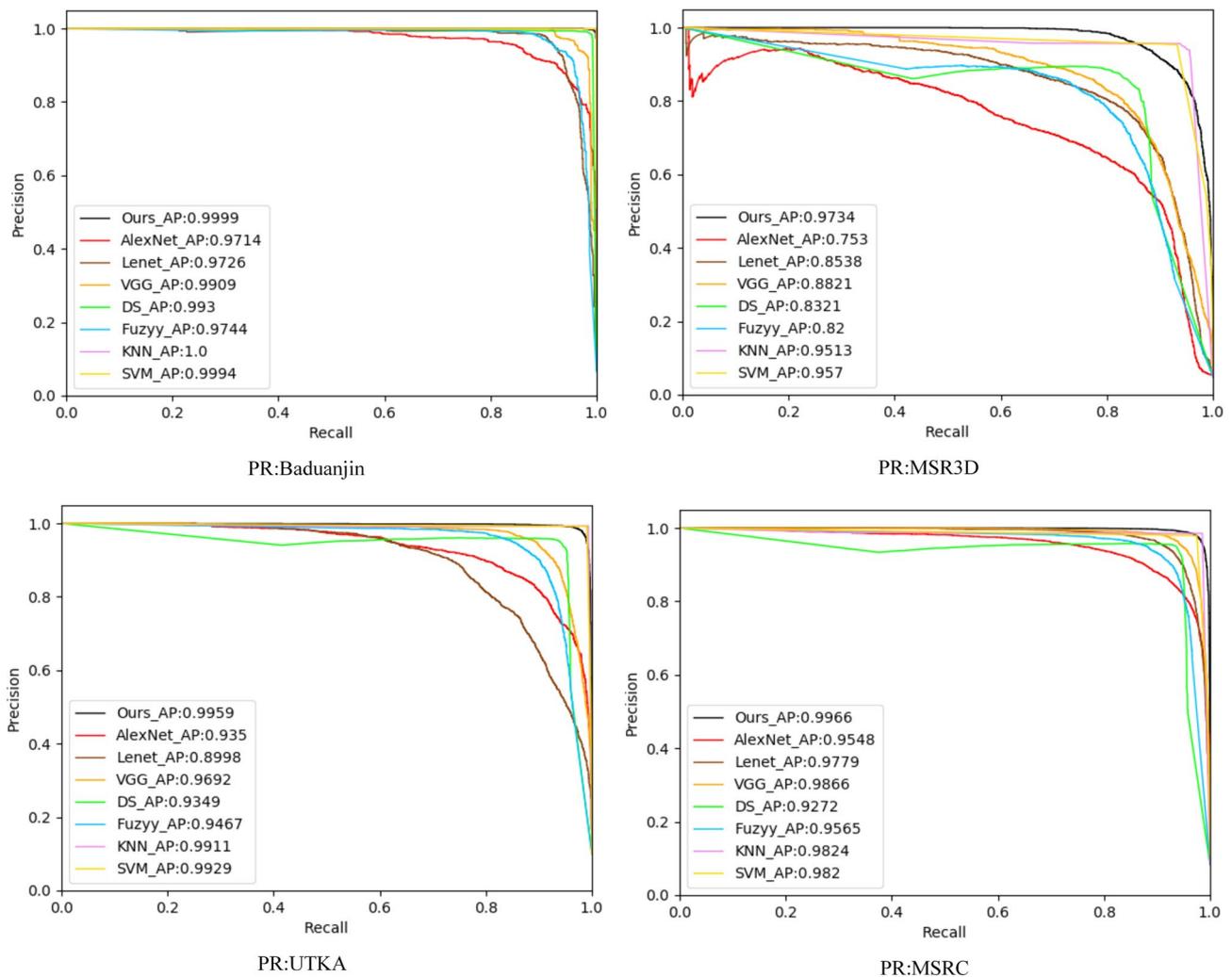
Fig. 10 Precision and recall curves of different methods

References

1. Agarwal A, Triggs B (2004) 3D human pose from silhouettes by relevance vector regression. In: Proc CVPR, vol 2, pp II–882–II–888
2. Allen FR, Ambikairajah E, Lovell NH, Celler BG (2006) Classification of a known sequence of motions and postures from accelerometry data using adapted Gaussian mixture models. Physiol Meas 27(10):935–951
3. Babu A, Dube K, Mukhopadhyay S, Ghayvat H, Jithin KMV (2016) Accelerometer based human activities and posture recognition. In: International conference on data mining and advanced computing, pp 367–373
4. Breiman L (1996) Bagging predictors. Mach Learn 24(2):123–140
5. Brown G, Wyatt J, Harris R, Yao X (2005) Diversity creation methods: a survey and categorisation. Inf Fusion 6(1):5–20
6. Chen K, Wang Q (2016) Human posture recognition based on skeleton data. In: IEEE international conference on progress in informatics and computing, pp 618–622
7. Cohen WW (1995) Fast effective rule induction. In: Twelfth international conference on machine learning. Morgan Kaufmann, Tahoe City, California, USA, pp 115–123

8. Ding WL, Zheng YZ, Su YP, Li XL (2018) Kinect-based virtual rehabilitation and evaluation system for upper limb disorders: a case study. J Back Musculoskelet Rehabil 31:611–621

9. Elforaici MEA, Chaaraoui I, Bouachir W, Ouakrim Y, Mezghani N (2018) Posture recognition using an RGB-D camera: exploring 3D body modeling and deep learning approaches. In: 2018 IEEE life sciences conference (LSC), pp 69–72. https://doi.org/10.1109/LSC.2018.8572079

10. Fothergill S, Mentis HM, Kohli P, Nowozin S (2012) Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI conference on human factors in computing systems

11. Foubert N, Mckee AM, Goubran RA, Knoefel F (2012) Lying and sitting posture recognition and transition detection using a pressure sensor array. In: IEEE international symposium on medical measurements and applications proceedings, pp 1–6

12. Furnkranz J (1999) Separate-and-conquer rule learning. Artif Intell Rev 13(1):3–54

13. Zhang Z, Liu Y, Li A et al (2014) A novel method for user-defined human posture recognition using Kinect. In: Proceedings of the 7th international congress on image and signal processing, IEEE, pp 736–740

14. Ho TK (1995) Random decision forests. In: Proceedings of the 3rd international conference on document analysis and recognition. Montreal, QC, pp 278–282

15. Ho TK (1998) The random subspace method for constructing decision forests. IEEE Trans Pattern Anal Mach Intell 20(8):832–844

16. Hu F, Wang L, Wang S, Liu X, He G (2016) A human body posture recognition algorithm based on BP neural network for wireless body area networks. China Commun 13(8):198–208

17. Islam MU, Mahmud H, Ashraf FB, Hossain I, Hasan MK (2018) Yoga posture recognition by detecting human joint points in real time using microsoft kinect. In: IEEE region 10 humanitarian technology conference, pp 668–673

18. Jiang H, Li ZN, Drew MS (2005) Human posture recognition with convex programming. In: IEEE international conference on multimedia and expo, pp 574–577

19. Jr MPP (2011) Combining classifiers: from the creation of ensembles to the decision fusion. In: 24th SIBGRAPI conference on graphics, patterns, and images tutorials. IEEE, Alagoas, Brazil, pp 1–10

20. Kohavi R (1995) Wrappers for performance enhancement and oblivious decision graphs. Ph.D. Thesis

21. Kononenko I, Kukar M (2007) Machine learning and data mining: introduction to principles and algorithms. Horwood Publishing Limited, Chichester

22. Le TL, Nguyen MQ, Nguyen TM (2013) Human posture recognition using human skeleton provided by kinect. In: International conference on computing, management and telecommunications, pp 340–345

23. Li B, Han C, Bai B (2019) Hybrid approach for human posture recognition using anthropometry and bp neural network based on kinect v2. EURASIP J Image Video Process 1:8. https://doi.org/10.1186/s13640-018-0393-4

24. Li W, Zhang Z, Liu Z (2010) Action recognition based on a bag of 3D points. In: Computer vision & pattern recognition workshops

25. Lin TY, Hsieh CH, Lee JD (2013) A kinect-based system for physical rehabilitation: utilizing tai chi exercises to improve movement disorders in patients with balance ability. In: Modelling symposium, pp 149–153

26. Liu H, Cocea M (2017) Granular computing based approach for classification towards reduction of bias in ensemble learning. Granul Comput 2(3):131–139

27. Liu H, Gegov A (2015) Collaborative decision making by ensemble rule based classification systems, vol 10. Springer, Basel, pp 245–264

28. Liu H, Gegov A, Cocea M (2016a) Rule based systems: a granular computing perspective. Granul Comput 1(4):259–274

29. Liu H, Gegov A, Cocea M (2016b) Rule based systems for big data: a machine learning approach. Springer, Basel

30. Liu H, Cocea M, Ding W (2018) Multi-task learning for intelligent data processing in granular computing context. Granul Comput 3(3):257–273

31. Livi L, Sadeghian A (2016) Granular computing, computational intelligence, and the analysis of non-geometric input spaces. Granul Comput 1(1):13–20

32. Min F, Xu J (2016) Semi-greedy heuristics for feature selection with test cost constraints. Granul Comput 1(3):199–211

33. Miranda L, Vieira T, Martínez D, Lewiner T, Vieira AW, Campos MFM (2014) Online gesture recognition from pose kernel learning and decision forests. Pattern Recognit Lett 39(1):65–73

34. Mun Wai L, Isaac C (2006) A model-based approach for estimating human 3D poses in static images. IEEE Trans Pattern Anal Mach Intell 28(6):905–916

35. Qiong HU, Lei Q, Huang QM (2013) A survey on visual human action recognition. Chin J Comput 36(36):2512–2524

36. Ramanan D, Sminchisescu C (2006) Training deformable models for localization. In: IEEE computer society conference on computer vision and pattern recognition, pp 206–213

37. Sarafianos N, Boteanu B, Ionescu B, Kakadiaris IA (2016) 3D human pose estimation: a review of the literature and analysis of covariates. Comput Vis Image Underst 152:1–20

38. Souto H, Musse SR (2011) Automatic detection of 2D human postures based on single images. In: Sibgrapi conference on graphics, patterns and images, pp 48–55

39. Wang J, Huang Z, Zhang W, Patil A, Patil K, Zhu T, Shiroma EJ, Schepps MA, Harris TB (2017) Wearable sensor based human posture recognition. In: IEEE international conference on big data, pp 3432–3438

40. Xia L, Chen CC, Aggarwal JK (2012) View invariant human action recognition using histograms of 3D joints. In: Computer vision and pattern recognition workshops, pp 20–27

41. Zainordin FD, Lee HY, Sani NA, Yong MW, Chan CS (2012) Human pose recognition using kinect and rule-based system. In: World automation congress, pp 1–6

42. Zhou ZH, Feng J (2017) Deep forest: towards an alternative to deep neural networks. In: Proceedings of the twenty-sixth international joint conference on artificial intelligence, pp 3553–3559

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.