The Visual Computer (2019) 35:591–607
https://doi.org/10.1007/s00371-018-1489-7

ORIGINAL ARTICLE

Improving bag-of-poses with semi-temporal pose descriptors for skeleton-based action recognition

Saeid Agahian[1] · Farhood Negin[2] · Cemal Köse[1]

Published online: 21 February 2018

Abstract

Over the last few decades, human action recognition has become one of the most challenging tasks in the field of computer vision. Effortless and accurate extraction of 3D skeleton information has been recently achieved by means of economical depth sensors and state-of-the-art deep learning approaches. In this study, we introduce a novel bag-of-poses framework for action recognition using 3D skeleton data. Our assumption is that any action can be represented by a set of predefined spatiotemporal poses. The pose descriptor is composed of three parts. The first part is concatenation of the normalized coordinate of the skeleton joints. The second part is consisted of temporal displacement of the joints constructed with predefined temporal offset, and the third part is temporal displacement with the previous frame in the sequence. In order to generate the key poses, we apply K-means clustering over all the training pose descriptors of the dataset. SVM classifier is trained with the generated key poses to classify an action pose. Accordingly, every action in the dataset is encoded with key pose histograms. ELM classifier is used for action recognition due to its fast, accurate and reliable performance compared to the other classifiers. The proposed framework is validated with five publicly available benchmark 3D action datasets and achieved state-of-the-art results on three of the datasets and competitive results on the other two datasets compared to the other methods.

Keywords Skeleton-based · 3D action recognition · Bag-of-words · Key poses · Extreme learning machine and RGB-D

1 Introduction

Vision-based action recognition has been extensively studied by many researchers due to its broad applicability on different areas ranging from surveillance, smart home, human computer interaction, robot vision, augmented reality to video summarizing and indexing [2,42,56]. In spite of the enormous efforts, action recognition still remains as a dynamic research field due to the major challenges which yet to be overcome. Among the challenges being faced, variability in view point, speed, acceleration and body size of the subjects, intra-class variation and inter-class resemblance of actions are the most important ones. Moreover, temporal and spatial segmentation of an action in videos, semantic parsing of the actions and sub-actions as well as obtaining enough training data are other challenges which need to be addressed in order to have generic solutions for a robust action recognition [45]. A conventional approach for action recognition task extracts handcrafted features of different modalities (such as RGB, skeleton joint position or depth map [43]) followed by classification of the videos based on the calculated feature vectors [42]. An action is described in three levels: low, mid and high levels [14,51]. In most of the early works, posture has been used as a high-level descriptor of human pose and their concatenation along the joint trajectories for action recognition. However, difficulties in body part detection, reliable pose recovery and high computational cost had been forced researchers to find an alternative track [14]. (We refer the readers to [11] for more details about these methods).

One major disadvantage of the methods that use low-level and mid-level features is their inability to represent complex activities due to their limitations in presenting semantic

✉ Saeid Agahian
saeid@ktu.edu.tr

Farhood Negin
farhood.negin@inria.fr

Cemal Köse
ckose@ktu.edu.tr

[1] Department of Computer Engineering, Faculty of Engineering, Karadeniz Technical University, 61080 Trabzon, Turkey

[2] INRIA Sophia Antipolis, 2004 Route des Lucioles, BP93, 06902 Sophia Antipolis Cedex, France

information [46]. One possible solution came with introduction of high-level semantic features [14], where action description carried out with a sequence of semantic lexicon encapsulating spatiotemporal body pose information. Subsequently, the Microsoft Kinect sensor provided cost-efficient high-level marker-less real-time pose extraction from RGB-D images [19,48] which had been a challenging problem for a long time. Lately, with the resurgence of deep learning methods, reliable and precise pose recovery from RGB images was obtained which were not limited to depth sensors anymore [4,10]. Considerable progress has been recently achieved in accurate marker-less pose detection awarding advantages such as its resistance to variation in view point, scale and appearance of a subject for action description compared to the low- and mid-level features. These privileges have been attracted many researchers to focus on these kinds of inputs and use them extensively for feature extraction tasks [14]. The main challenge to use this information for action recognition is their heterogeneous numeric representation of semantically similar actions.

There have also been a lot of efforts to preserve temporal information. To cope with the evolution of variability in motion patterns, applying temporal pyramid [67] and producing histograms for distinctive segments of actions have gained more popularity. Meanwhile, some methods were used to add temporal features such as speed to describe each pose by keeping temporal information [14,57]. Although in traditional generative methods, it has been shown that the temporal structure is essential for understanding dynamics of the actions, it has not been counted as a critical aspect of deep networks such as ConvNet frameworks. The primary focus of these methods is on appearance and short-term motion (limited number of frames) rather than comprising long-term temporal structure of actions in videos. Nevertheless, temporal ordering is one of the main characteristics of many actions.

Recurrent neural networks (RNN) and long short-term memory (LSTM) networks have achieved remarkable success in text and sound recognition for modeling temporal dependencies in sequences [27]. This has been inspired researchers to use variation of RNN [13,53] and LSTM networks [28,31,73] with skeleton information. Computational complexity of these networks makes it unsuitable for real-time and online tasks [18,35]. However, better results have been recently gained for modeling action dynamics using long short-term memory than HMMs and temporal pyramid [28]. Intuitively, it is expected that a video representation incorporating temporal ordering has a better discriminative characteristics, though obtaining an all-embracing representation still remains as a significant challenge.

From the above-mentioned studies, it can be realized that the suggested methods (specially bag-of-words-based methods) still fail to completely model concept of time and relationship between the poses [14,57,67]. In this study, we propose a pose-based action recognition framework to address this problem. Simplicity, interpretability and high processing performance in recognition tasks are the major advantages of our proposed methodology.

The main idea is to describe an action with a sequence of predefined poses and encode it by histogram of those poses. Figure 1 illustrates overall data flow of the proposed method. We describe poses of a sequence by defining a simple and efficient semi-temporal feature. This feature enables us to distinguish between the two actions with the same skeleton configuration and different temporal orders of their key poses. Our proposed descriptor created more discriminative key poses by using training poses for action representation. Embedded temporal information in the key poses helped us to overcome limitations of the bag-of-words methods by encoding actions with histogram of the key poses. The length of feature vector that describes the actions is fixed and independent of the total number of frames. Finally, we use discriminative extreme learning machine [21] for classifying the actions. We tested the proposed methodology on the five publicly available benchmark datasets including 3D skeleton data. The experiments showed that our method is capable of producing state-of-the-art results on three of the datasets only by using skeleton information, while competitive results on the fourth and fifth datasets.

This paper is organized as follows: In Sect. 2, there is a brief explanation of available approaches in the literature followed by details of our approach in Sect. 3. The experimental evaluation and results on the five public datasets are presented in Sect. 4, and finally, we conclude and summarize the paper in Sect. 5.

2 Related work

In this section, we briefly explain pose-based methods that only employ articulated 3D skeleton data for action recognition. It should be noticed that our main focus is on daily living activities performed by a single person (not interactive).

The 3D skeleton data represent relations between the joints and overall configurations of human poses. This information can be extracted from different modalities such as motion capture systems (MoCap), stereo, range sensors [1,18]. As a pioneering study on human action recognition, Johansson [25] showed that availability of the joint position sequence is sufficient to recognize human actions. Yao et al. [64] showed that in indoor action recognition scenarios, using pose-based features resulted in a better recognition performance compared to appearance-based features.

In general, all pose-based action recognition approaches are consisted of two major steps: First, human poses in each frame are described by the features extracted from

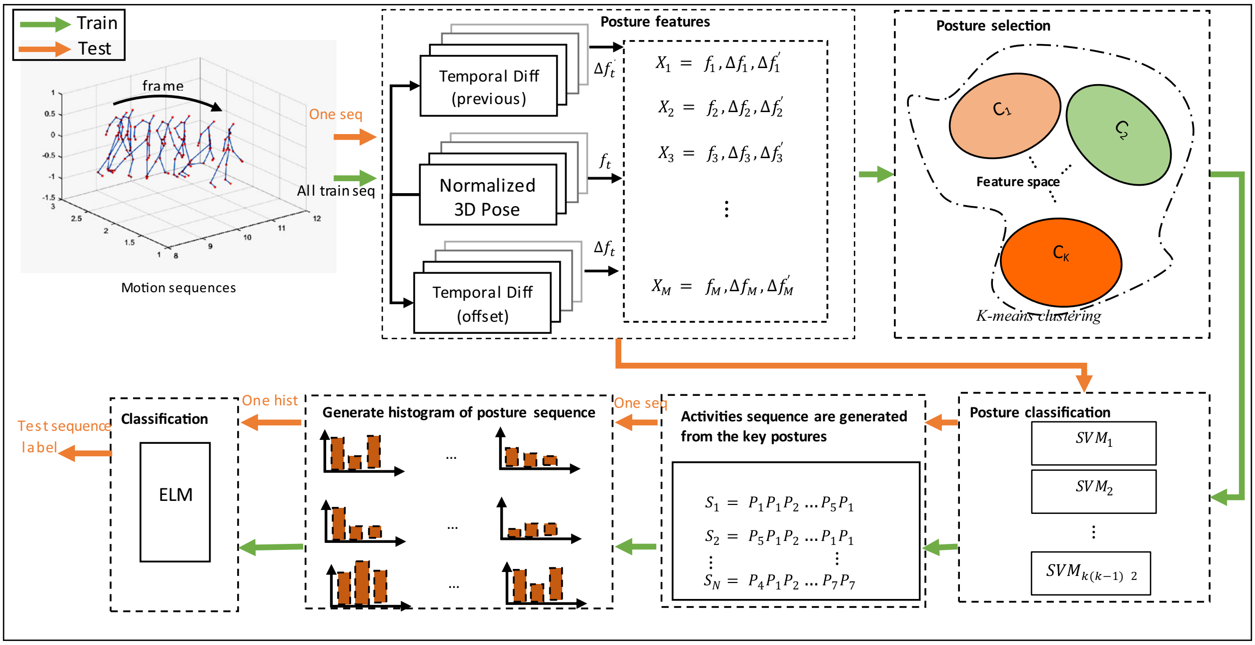

Train
Test
frame
Motion sequences
One seq
All train seq
Posture features
Temporal Diff (previous)
Normalized 3D Pose
Temporal Diff (offset)
$\Delta f_t'$
f_t
Δf_t
$X_1 = f_1, \Delta f_1, \Delta f_1'$
$X_2 = f_2, \Delta f_2, \Delta f_2'$
$X_3 = f_3, \Delta f_3, \Delta f_3'$
$X_M = f_M, \Delta f_M, \Delta f_M'$
Posture selection
C_1
C_2
Feature space
C_K
K-means clustering
Posture classification
SVM_1
SVM_2
$SVM_{k(k-1)\ 2}$
Activities sequence are generated from the key postures
$S_1 = P_1P_1P_2 \dots P_5P_1$
$S_2 = P_5P_1P_2 \dots P_1P_1$
$S_N = P_4P_1P_2 \dots P_7P_7$
One seq
Generate histogram of posture sequence
One hist
Classification
ELM
Test sequence label

Fig. 1 Workflow of the proposed method

raw 3D skeleton data, and second, the final feature vector is calculated for the whole action sequence to be used in classification or reasoning. Han et al. [18] named these two steps as “information modality” and “representation encoding,” respectively. According to this taxonomy, various 3D pose-based features which are used for describing actions are categorized into four groups based on displacement [3,9,59,60], orientation [62], raw joint positions [3,5,53,73] and multi-modal [14,30,38,54,66]. The encoding methods are categorized into three main groups. Concatenation-based approach is the most straightforward encoding method carried out by simply concatenation of the extracted features into a one-dimensional final feature vector [15,59]. The generated feature vector is too long and is therefore practically difficult for classifier to handle the high-dimensional space. Statistical encoding is a common and efficient method for integrating the features which is performed by applying statistical analytics on constituent feature vectors without using any feature quantization operation [22]. A lack of order in feature elements and absence of temporal relation can be considered as the most important drawbacks of these methods. Bag-of-words encoding methods apply coding operator and dictionary learning for mapping a high-dimensional feature vector into a single code word in a dictionary. In a study conducted by Han et al. [18], they extracted different features from skeleton data and applied these three encoding methods to the obtained feature vectors. Their results indicated that the bag-of-words encoding methods gave a better performance compared to the other methods on four benchmark datasets that they examined.

In terms of dictionary learning, the encoding methods are generally divided into two main categories: clustering and sparse coding-based methods [18]. Losing temporal information among the features is a major shortage of these methods. There are studies [26,57,66] in the literature conducted to overcome this deficit and improve reliability of the encoding methods. In order to extract spatial/temporal structure of the poses in each action class, [57,66] used data mining techniques. They grouped skeleton joints in training data by k-means clustering and used cluster centers as the code words which encodes the spatial information of the action. For encoding temporal structure of each action class, they employed mining techniques (such as Contrast Mining in [57]) to extract sub-sequences occurring frequently among sequences of each group. This method benefits from a pose recovery technique that helps to improve pose detection from images. However, applying data mining on both of the encoding steps leads to a high computational cost. Instead of costly process of mining poses, our method uses classification in order to create the pose sequence of each action which makes it more efficient.

Temporal pyramid method is one of the alternatives for representing temporal information in bag-of-words methods [14,26,32]. Most of the studies that have used this method for temporal localization of poses ignore dynamics of action (e.g., segmenting the sequence into equal chunks). As a result, they became incapable of describing the same action performed by different speeds. To reduce the effect of unfit segmentation, Liu et al. [32] proposed a descriptor using a motion energy for clipping the sequences. Rather than using skeletons' energy values for encoding time, our proposed descriptor models temporal dependency of geometrical features by associating each pose word with previous poses with a randomly selected displacement offset. For each pose word, it also keeps the displacement information regarding the previous pose in the sequence.

Selecting representative features in the training step in successful studies carried out by expensive computational methods such as data mining or other feature selection mechanisms [14]. Providing spatial/temporal information using these mechanisms for the bag-of-words methods is accompanied by a higher level of complexity. The most similar bag-of-words method with our study is [34]. In this study, for calculating temporal displacement pose descriptors at frame t with a randomly preselected differential time offset Δt, for each element i, they obtained $\triangle\theta^i = (x_t^i - x_{t-\Delta t}^i, y_t^i - y_{t-\Delta t}^i, z_t^i - z_{t-\Delta t}^i)$. Accordingly, feature vector was constructed by concatenating the calculated $\triangle\theta^i$ for each element $(i \in 1, \ldots, m)$. K-means was applied on extracted pose descriptors on the training data, and encoding was performed by finding the closest cluster center to the obtained pose word. Before feeding the descriptors into a Naive Bayes voting-based classifier, each part of the motion was separately encoded followed by generating a histogram specific to each part. The main difference between their method and ours appears in pose encoding phase which was conducted in low-level and high-level pose encodings, respectively. Each word in our method describes a real pose, while in [34], a word is a directed vector describing each local part. Our descriptor is effective as it produces low-dimensional feature vectors which are independent of the number of the skeleton elements and only depends on the number of the key poses. Lu et al. [34] ignored spatial information, while our method uses spatial information along with the temporal pose information.

Nowadays, due to the extensive progress in the image processing and deep learning-based classification methods such as convolutional neural networks (CNN), researchers have been encouraged to employ these methods for skeleton-based action recognition. However, there are still many challenges that need to be resolved. These methods are designed to accept images as input and cannot capture the dynamic information in skeleton sequences. Therefore, an encoding method including spatiotemporal information of a sequence in two-dimensional image space is required. Some of the studies in the literature suggest converting skeleton pose sequences into an image containing dynamic information and asking the network to classify synthesized images. For example, in [20], Hou et al. proposed a new encoding method called "Skeleton Optical Spectra" (SOS) which transforms skeleton sequences into texture images. The generated textures were used as an input for a CNN network to extract separable features, and classification was performed using the average output of the CNN network.

The proposed approach is a pose-based method which utilizes bag-of-words (bag-of-poses) method for encoding. Our method is distinguished from other existing methods owing to use simple features extracted from raw joint positions of the skeleton data which achieves higher computational efficiency. These features are directly extracted from raw joint positions without transforming them to another space such as Lie Groups [54,55]. The temporal information is embedded into the bag-of-words dictionary without using complex data mining methods [58]. This is performed by generating spatial/temporal poses as words of the dictionary. Therefore, the generated histograms inherently contain temporal information and using multiple histograms is not required for handling time information.

3 Proposed method

As the input, the proposed framework accepts a sequence of high-dimensional vectors of skeleton joints for each action with T frame and J joints for each skeleton:

$$S = \{P_t \mid \forall\, t \in (1, \ldots, T)\} \qquad (1)$$

where $P_t = \left\{p_t^i \mid \forall\, i \in (1, \ldots, J)\right\}$ is the set of skeleton joints at tth frame and $p_t^i = (x_t^i, y_t^i, z_t^i)$ is the ith joint of the skeleton p in tth frame.

The coordinate system of the framework (x, y, z) is defined based on the location of the camera as shown in Fig. 2. The center of the coordinate system matches with the center of the camera. Inspired by the conventional bag-of-words methods, our proposed method describes an action as a sequence of pose-words (key pose). Encouraged readers can refer to [41] which has compiled a comprehensive survey summarizing bag-of-words methods applied on action recognition problems. The overall flow of our framework is shown in Fig. 1. A preprocessing step precedes feature extraction process to make the input skeleton information invariable to subject position, scale and camera view.

3.1 Preprocessing and feature extraction

The preprocessing step makes the input skeleton data:

Transform invariant: In each frame, we transform the origin of the coordinate system from real-world coordinates to hip center of the person. This transformation makes the position of the skeleton joints invariable to the location of the subject.

Scale invariant: In general, people performing an action have diverse ranges of body sizes. In order to have robust action models, the generated action features of different sub-

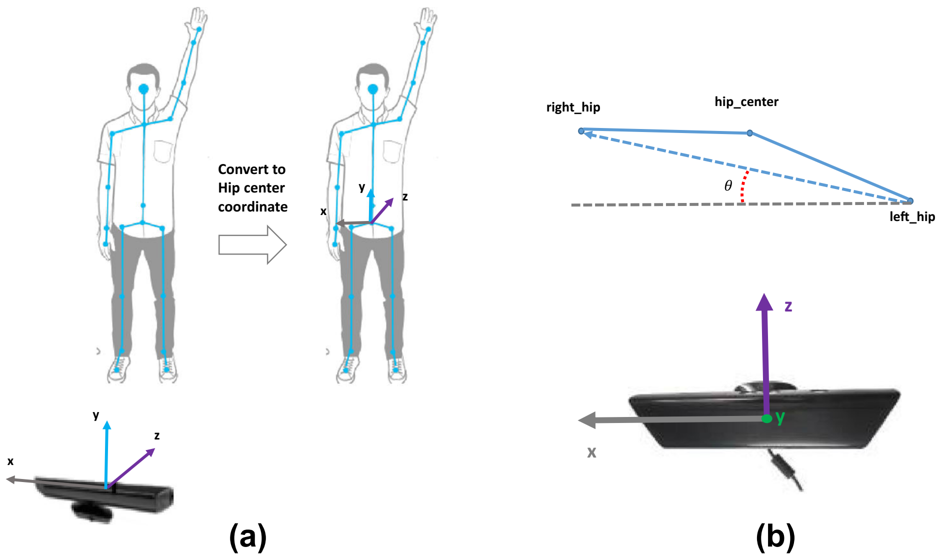


Fig. 2 **a** Assumed settings in the proposed method with the Kinect placed in the origin of the coordinate system **b** rotation of the skeleton toward the origin

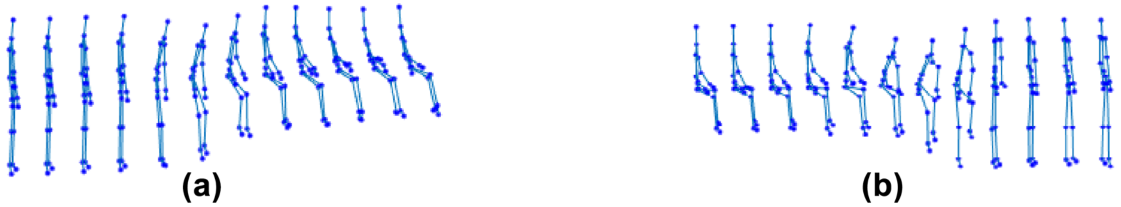


Fig. 3 **a** Sit-down and **b** stand-up actions

jects should preserve consistency among the representations. Different methods have been proposed in the literature for maintaining the scale invariability which among them, we use a method similar to [55]. First, we choose a random pose as a reference and afterwards we rescale the remaining poses' limb sizes to the size of the corresponding body parts in the reference pose which preserves original angles between the pose parts.

Rotation invariant: To make skeleton joints invariant to the camera view, a specific rotation is performed with respect to the specified view point of the camera. As shown in Fig. 2b, this transformation ensures that the projection of the vector passing from left hip to right hip on ground plane stays parallel with x axis in the real world coordinates.

Given a normalized pose, the next step generates a pose descriptor. Lillo et al. [30] classified features of the pose descriptors in two categories:

i. *Geometric descriptor*: These descriptors represent the spatial configuration of the skeleton joints in each frame. They use calculated angle between the skeletal vectors or computed distance between the joints using different metrics.
ii. *Motion descriptor*: Although the geometric descriptors are capable of defining spatial configurations of skeleton joints, they are unable to encode dynamic information of the poses. In order to codify motion dynamics in representation of pose motion descriptors, information such as velocity, speed, derivation and optic flow is used in the calculations. Motion descriptors also avoid the ambiguity between the two poses, while they embody different action characteristics with similar spatial configurations (Fig. 3).

While the proposed descriptor intrinsically contains geometric information, it also tries to keep the track of dynamic of actions by taking into account temporal dependency between the consecutive frames. The final pose representation is composed of different combinations of the descriptor types. One popular combination strategy is to concatenate all of the extracted features. Due to an increase in dimensions of the descriptor, cost of the classification also proportionally scales up. In order to keep the dimensionality manageable, most frequently dimensionality reduction procedure such as PCA or LDA has been used. Although dimensionality reduction brings efficiency to processing of the descriptors, it is computationally expensive and sometimes does not culminate the accuracy [16]. An alternative strategy called feature engineering rather than blind concatenation of features tries to single out the most representative ones in the feature set. Feature engineering is usually done either manually (handcrafted) or automatically (learning based), e.g., supervised sparse dictionary learning, neural network, genetic programming, CNN or random decision forests [71]. Since feature selection mechanisms are computationally expensive, they cannot be a suitable option for a real-time application [18]. Unlike feature selection-based methods, our features are similar to the one in [9] and give an efficient pose description.

As illustrated in Fig. 1, for describing spatial configuration of the skeleton in each frame, we define the feature vector for tth frame as

$$f_t = \{\&(x_t^i, y_t^i, z_t^i) \mid \forall\, i \in (1 \cdots J)\}$$

which concatenates normalized coordinate of the skeleton joints. (J is number of joints of the skeleton.) As mentioned before, in order to model the temporal dependency between the poses in different frames and make the descriptors to embody information of similar action configurations with composite temporal dependencies, we define another vector Δf_t that models temporal dependency by taking into account a randomly selected frame offset (t'):

$$\Delta f_t = \begin{cases} f_t & 1 \leq t < t' \\ \frac{f_t - f_{t-t'+1}}{\| f_t - f_{t-t'+1} \|} & t' \leq t \leq T \end{cases} \tag{2}$$

If the current pose occurs before the offset, the calculated vector contains regular joint features. Otherwise, it calculates the distance between the current pose and all of the dependent poses in the range of the temporal offset. We also calculate $\Delta f_t'$ as another feature vector comprising displacement of the current pose with the previous pose in the sequence.

$$\Delta f_t' = \begin{cases} f_t & t = 1 \\ f_t - f_{t-1} & 2 \leq t \leq T \end{cases} \tag{3}$$

The final feature vector of tth pose is composed of $X_t = [f_t, \Delta f_t, \Delta f_t']$ which is concatenation of spatiotemporal features and its dimension $D = 3 * J * 3$ is linearly dependent on the number of the skeleton joints.

3.2 Key poses selection

Similar to the bag-of-words methods, our framework represents a sequence of an activity with a set of initially learned key poses (words in the dictionary). The dictionary of the key poses therefore needs to be learned, and subsequently, high-dimensional pose features are encoded into a single word. Conventionally, there are two ways to learn the dictionaries:

i. The first way is to divide the feature space into subregions and then express each region with its representative (the code word). K-means algorithm has been widely used for this purpose [16,26,57].

ii. The second way is to determine distributions of the features using a generative model. Gaussian mixture model (GMM) is the most popular method used for this purpose. The K-means algorithm generates the words from feature vectors based on hard assignments (i.e., uses Euclidean distance to find the closest center), while GMM performs soft assignment instead (i.e., it uses probability distribution of the features for code words assignment rather than mean value) [41].

The accuracy of classification is directly related to quality of the trained dictionary and feature encoding. In case of K-means algorithm, as dimensionality of the feature vectors increases, Euclidean distance performs poorly and starts to generate unreliable encodings. Therefore, to improve dictionary learning and encoding, we perform it in two steps (Fig. 1) [16]. To generate pose words for the dictionary (Key poses), the K-means algorithm is applied on the pose feature vectors of all the training frames:

$$\text{Poses} = \left\{ \bigcup_m \bigcup_t X_t(m) \right| \\ \forall\, m \in (1, 2, \ldots, M) \text{ and } \forall\, t \in (1, 2, \ldots, T)\} \tag{4}$$

where M is the number of trials in the training set and T is the number of frames. Consequently, the feature space is divided into a K clusters and their corresponding cluster centers. The obtained cluster centers are considered as the key poses and passed to the next step of the framework.

3.3 Pose classification and encoding

To resolve problem of the Euclidian distance in the encoding phase, we train a set of SVM classifiers using the key poses of dictionary and carry out assignments using the classification. For implementing, we use LIBSVM [6] library in which one-against-one method is used for classification of the key poses. We train $S = \frac{K*(K-1)}{2}$ binary SVMs for classification of K poses. For assignment of the feature vectors to the key poses, we use learned binary SVMs with "max wins" voting strategy. Using hyperplanes for classification of the pose words yields in a better assignment results than K-means [16].

3.4 Action representation using key pose histograms

In this step, we use the trained SVM classifier to convert each action's feature vector into a sequence of the key poses. The sequence of the produced poses has a variable length due to variety of the frame number in the videos. For classification of the variable length sequences, methods such as Hidden

Markov Model, Bayesian Network and Dynamic Time Warping are used [42]. For classification of the activities, we can use discriminative classifiers such as SVM, KNN and ANN. Normalizing the length of the feature vectors to a fixed length is usually done in two ways: sampling video frames to the desired size and then extracting the feature vectors. The other method quantifies values of the feature vectors and use the histogram of quantized values to describe the entire action [45]. We describe each activity with a fixed length feature vector, and we then calculate histogram of the sequence containing constituent key poses. Prior to these calculations, the length of histograms is determined with the number of extracted key poses.

3.5 Action classification

There are several popular classifiers such as KNN, SVM, ANN and random forest for classification of the fixed length feature vectors. In this work, we use extreme learning machine (ELM) classifier [21] in order to classify actions. ELM is a single-layer feed-forward neural network classifier which has been successfully applied in various applications and has shown high learning speed and viable accuracy. For the first time, Minhas et al. [36] used this classifier in motion-based features to detect human actions and they obtained promising results. Moreover, this method is not limited to low class number and small-scale classifications and can be used in large-scale realistic tasks. Varol and Salah [52] used ELM for action recognition of realistic video clips and achieved acceptable results by considering heavy computational cost of deep neural network methods. In recent years, this method also has been used to detect human actions with skeleton data [9,65].

4 Experimental evaluation and results

We evaluated our method on five challenging benchmark datasets. We assume that there is only one person performing the assigned actions. This explains that why we observe a drop in performance when interactive actions are evaluated.

UTKinect action dataset: This dataset [62] was collected by Xia et al. at the University of Texas at Austin in 2012. The data were captured by Kinect v1 in 30 fps and included 10 actions. Each action was performed by 10 subjects (9 men and 1 woman) for 2 times. In total, 200 sequences exist in the dataset. The dataset included RGB, depth and skeleton where the sequences were manually clipped. Similarly, skeleton data in each frame were represented by Euclidean position of 20 joints relative to the origin. Variability of subjects' position and orientation toward the camera, variation of performance among different patients and noticeable difference in speed and duration of the actions are the main challenges of this dataset. Human–object occlusions and out of field-of-view body parts make the sensor unable to recover all of the body parts and add them up to the challenges being faced in this dataset.

CAD-60 dataset: Daily activities rarely occur in controlled laboratory environment. This has motivated researchers at the Cornell University to create CAD-60 dataset [49] for actions occurring in the real environments. Four subjects performed 12 different actions in 5 different environments where depth, RGB and skeleton data for each instance are captured by Kinect v1 sensor in 30 fps. Each action is performed at least one time by each subject. In total, dataset includes 60 sequences with an average length of 45 s for each action. Skeleton data for each frame are presented with Euclidean position of 15 joints by taking sensor coordinates as the reference point. Insufficient training data and variable background are the main challenges of this dataset. The actions are performed with different laterality as one of the subjects is left-handed. In order to compensate the effect of laterality, some of the proposed methods [40,47,49] also added a mirrored version of these instances to the training data to achieve invariance toward handedness of the subjects.

UTD-MHAD dataset: UTD-MHAD [7] is a multimodal dataset which was released by the University of Texas for the multimodal activity recognition. The data were captured by Kinect v2 at 30 fps and a wearable inertial sensor. Four data modalities including RGB, depth, skeleton and inertial signal were registered in temporal synchronized mode using these sensors. The dataset includes 27 action. These actions were performed by 8 subjects (4 men and 4 women) in an environment with a fixed background. Every subject performed each action for 4 times. The skeleton data for each frame were presented by Euclidean position of 20 joints with respect to the sensor coordinates. In another taxonomy, this dataset categorized actions in four sub-categories: sport actions (e.g., bowling, tennis serve and baseball swing), hand gestures (e.g., drawing x, triangle and circle), daily activities (knocking the door, standing and sitting) and training exercises (e.g., arm curl, lunge and squat).

MSR action 3D dataset: MSR action 3D dataset [29] is the first public RGB-D action dataset which was created by Microsoft Research Redmond. The dataset was recorded by Kinect v1 in 15 fps and included 20 actions involving different body parts. Each action was performed by 10 subjects for 2–3 times. In total, 567 sequences exist in the dataset with the lengths varying between 13 and 67 frames. Each sequence included an action which was manually segmented. The dataset also included depth and skeleton data of each action. Skeleton in each frame was represented by Euclidean position of 20 joints relative to the origin which was the sensor coordinate. In all instances, subjects performed actions in a fixed position facing toward the camera with a controlled background.

Table 1 Summary of the datasets

Dataset name	Actions	Subjects	Sequences	Joints	Year
UTKinect [62]	10	10	199	20	2012
CAD-60 [49]	12	4	60	15	2011
UTD-MHAD [7]	27	8	861	20	2015
MSR action 3D [29]	20	10	557	20	2010
MSRC-12 [15]	12	30	594 (6244 instance)	20	2012

Table 2 Investigating parameters of our approach

Dataset name	Investigated intervals and steps		
	Temporal offset	Key poses numbers	Neuron numbers
UTKinect [62]	4:1:20	100:10:200	500:100:3500
CAD-60 [49]	10:10:150	100:10:250	500:100:3500
UTD-MHAD [7]	4:1:20	150:10:280	500:100:3600
MSR action 3D [29]	4:1:20	150:10:250	500:100:3500
MSRC-12 [15]	4:1:11	100:10:200	500:100:3100

MSRC-12 dataset: MSRC-12 dataset [15] is a bigger dataset compared to the previous ones which makes it more suitable to evaluate scalability of our approach. This dataset contains only skeleton information and was collected by Microsoft Research to evaluate the effects of different instruction modalities on recognition of actions. It was recorded in 30 frames per second and stored position of 20 skeleton joints at each frame. 30 subjects performed 12 actions which were divided into two groups of 6 actions each. In total, there were 594 action sequences in the dataset each containing one action performed by one subject with 5 different instructions recorded in succession. (In total, there were 6244 action instances in the dataset.) For segmentation of the sequences, we use labeling information in [22].

A summary of general characteristics of the five datasets used in our experiments for evaluating the proposed method is shown in Table 1.

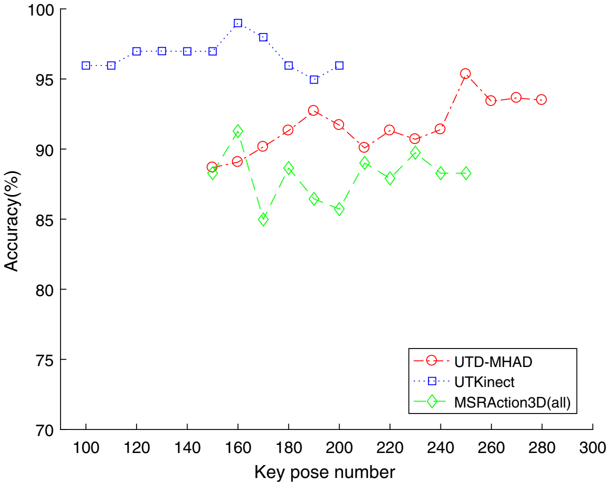


Fig. 4 Evaluating the key pose parameter

Experimental settings: In our experiments, 3D coordinates of the skeleton joints are converted from world coordinates into subject coordinates by taking hip center joint as the coordinate system's origin in each frame. The obtained results in each dataset are compared to the methods that use only skeleton data for recognition tasks. Three input parameters of our framework are individually tuned for each dataset. The first parameter is the temporal offset (t') which is used for constructing temporal difference (Δf_t) in the feature vector. The second parameter is the number of clusters in k-means clustering method which is used to extract key poses from all of the training poses. In other words, it represents the number of key poses. The last parameter that needs to be tuned is the number of neurons in the hidden layer of the ELM. We start with big steps and wider range of parameters and narrow down the intervals to find the optimal values. As shown in Table 2, we empirically determine the optimal intervals and the best fit of the step size which ensure the best overall performance of the recognition framework. We perform a random initialization of the cluster centers in k-means method to calculate the key poses. The proposed method is therefore repeated 20 times on each dataset, and the best result is reported and is compared with the state-of-the-art approaches.

We also illustrate detailed process of parameter investigation in Figs. 4, 5 and 6. Due to different protocols of each dataset, we depict the tuning process for 3 of the evaluated datasets. It can be seen that a lower number of the key poses fails in representing the poses in the dataset, while a larger number of poses increases the noise and drops the accuracy. There is an optimum spot in each dataset for the key pose number which gives the best performance.

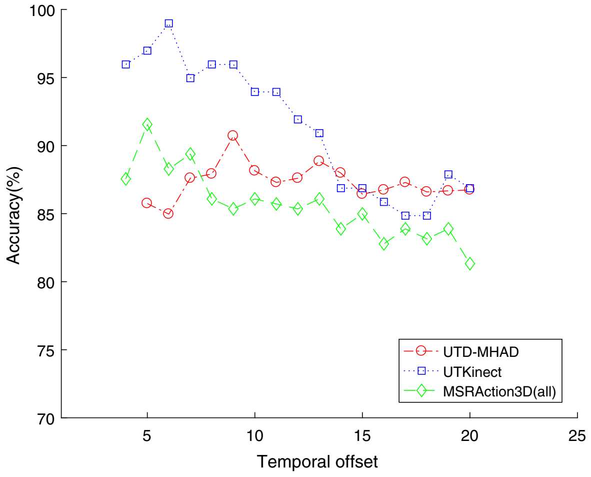


Fig. 5 Evaluating the temporal offset parameter

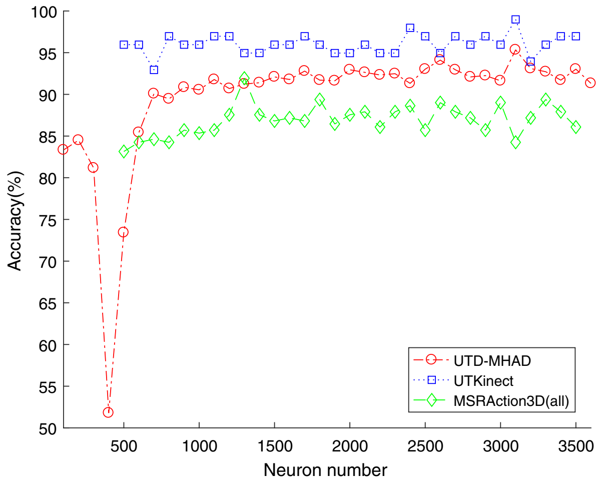
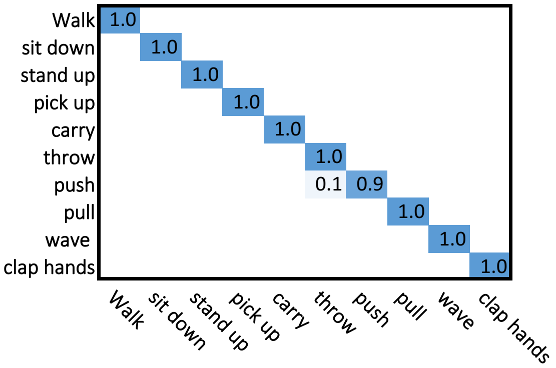


Fig. 6 Evaluating the neuron parameter

Table 3 Comparison with the state-of-the-art results on *UTKinect action* dataset

Feature engineering	Method	Accuracy (%)
Handcrafted	HOJ3D [62] (LOSeqO)	90.9
	Lie group [54]	97.8
	Spatiotemporal SHs [65]	93.0
	Pairwise joints [32]	94.4
	Our method	98.9
Learned representations	RDF-based [37]	92.0
	Max-margin multitask [63] (LOOCV)	98.8
	LMNN [35] (LOOCV)	98.0
RNN-LSTM	Multilayer LSTM [69]	95.9
	ST-LSTM [31]	95.0
	TS-LSTM [28]	96.9



	Walk	sit down	stand up	pick up	carry	throw	push	pull	wave	clap hands
Walk	1.0									
sit down		1.0								
stand up			1.0							
pick up				1.0						
carry					1.0					
throw						1.0				
push						0.1	0.9			
pull								1.0		
wave									1.0	
clap hands										1.0

Fig. 7 Confusion matrix of UTKinect dataset

From Fig. 5, it can be clearly seen that the accuracy drops significantly when the offset value increases. A small number of neurons have negative effect on accuracy of UTD-MHAD dataset, whereas it improves linearly as the number of neurons grows. For the other datasets, accuracy fluctuations are not significant.

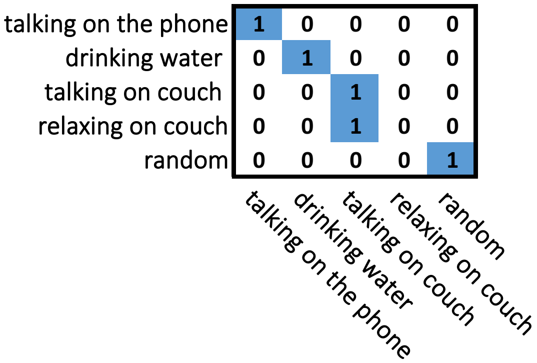
In the seminal work based on the UTKinect action dataset [62], the authors used leave-one-sequence-out (LOSeqO) protocol for their evaluations. In this protocol, they randomly selected one sequence at a time from the entire dataset as the test instance and used the remaining sequences as the training data. This process was repeated in certain times, and average of the obtained results was used as the final performance [68]. In our experiments, we follow cross-subject protocol in [54]. Subjects 1, 3, 5, 7 and 9 are selected for training and subjects 2, 4, 6, 8 and 10 for testing. This evaluation protocol is more realistic since the test subjects' actions are kept out of the training set. We use Table 2 to find the optimized parameters for UTKinect action dataset. We obtained the best performance by setting temporal offset to 6, key pose number to 160 and number of neurons to 3100. The results and comparisons with the state-of-the-art methods are shown in Table 3. As far as we know, the best performance achieved among all the skeleton-based approaches using UTKinect action dataset is obtained by our method, as shown in Table 3. Based on the confusion matrix (Fig. 7), 10% of the test samples of "push" action are misclassified as "throw" action. Similarity between poses of the two actions and noise in skeleton joint positions are potentially the main causes of the recognition failure. Our method successfully recognizes nine out of ten actions with 100% accuracy.

Sung et al. [49] presented two types of protocol called "new person" and "have seen" for evaluating CAD-60 dataset. They used precision/recall measures to evaluate their

Table 4 Comparison with the state-of-the-art results on *CAD-60* dataset

Feature engineering	Method	Precision (%)	Recall (%)
Handcrafted	MEMM [49]	67.9	55.5
	3D posture [16]	77.3	76.7
	Pose kinetic energy [47]	93.8	94.5
	Decision-level fusion [74][a]	96.4	84.6
	Our method	98.5	99.0
Learned representations	M-L codebooks of key pose [72]	97.4	95.8
	Self-organizing neural int [40]	91.9	90.2
	RF-key pose [39] (Random + Still)	81.8	80.0

[a] Notice that in this method, both depth and skeleton information are used



	talking on the phone	drinking water	talking on couch	relaxing on couch	random
talking on the phone	1	0	0	0	0
drinking water	0	1	0	0	0
talking on couch	0	0	1	0	0
relaxing on couch	0	0	1	0	0
random	0	0	0	0	1

Fig. 8 Confusion matrix of "living room" actions related to subject 3

proposed method. In our experiment, we adopt "new person" protocol for evaluations. This protocol was defined as a leave-one-subject-out cross-validation. One subject was therefore used for testing, while the other three subjects were kept for training. In CAD-60 dataset, one of the four subjects was left-handed (subject number 3). We use mirroring operations before constructing the feature vector in order to convert laterality of the actions and to make it similar to the right-handed actions. Zhu et al. [74] achieved state-of-the-art results on CAD-60 dataset. In their approach, subject number 2 was considered for testing and the other three subjects (1, 3 and 4) for training. We adopt the same setting in our experiments. Length of the actions in this dataset was longer than the previous one. Using Table 2, we tried different parameter intervals and step sizes. By examining all the possible scenarios for the parameters in these intervals, we obtained the best performance with value of 50 for temporal offset, 210 for key pose number and 3100 for number of neurons on CAD-60 dataset. Performance of our method and comparisons with the successful approaches in the literature using CAD-60 are shown in Table 4. It can be noticed from Table 4 that our proposed method achieved competitive performance compared to the handcrafted skeleton-based methods. Except subject 3, all of the actions in different environments 3 are recognized with 100% success. As it is clear from the confusion matrix (Fig. 8), recognizing "talking on coach" action instead of "relaxing on coach" is the only failure occurred in subject three's instances. Insufficient training samples are the main reason for this failure. Since there is only one test instance available for "relaxing on coach" related to the subject 3, the calculated precision turns out to be undefined value of 0/0. To compute average precision of the actions in "living room" environment, we consider this value as zero.

The common practice in UTD-MHAD dataset [7] was to perform cross-subject evaluation protocol which was suggested by its providers. In this protocol, half of the subjects (1, 3, 5 and 7) were taken for training and the other half (subjects 2, 4, 6 and 8) for testing. In our experiments, we use the same setting for evaluating our proposed method. Similar to the previous datasets, we investigate the optimal parameters through the values indicated in Table 2. We obtained the best performance with value of 9 for temporal offset, 250 for key pose number and 3100 for number of neurons on UTD-MHAD dataset. (The evaluation of these parameters are shown in Figs. 4, 5 and 6.) To the best of our knowledge, the best performance among all the skeleton-based approaches on UTD-MHAD dataset is obtained by our method as shown in Table 5. Analysis of the confusion matrix in our method on this dataset (Fig. 9) showed that actions sharing common poses lead to inaccurate recognition. For instance, "throw" action is classified with 75% accuracy, while in 20% of samples, it is misclassified as "draw x." In a similar situation, "jog" is misclassified in 25% of times as "walk" action. Nevertheless, 18 actions out of 27 are recognized with a 100% accuracy. Existence of distinctive poses leads the framework to distinguish these actions with a perfect accuracy.

There are two settings which have been used in the previous studies to evaluate MSR action 3D [29] dataset. The first one was proposed in the seminal paper [29] of MSR action 3D dataset where all of the actions were divided into three sub-categories (AS1, AS2 and AS3) shown in Table 6. Every sub-category was consisted of 8 action classes whose training and classifications were independently performed on each category. In sub-categories AS1 and AS2, actions with sim-

Table 5 Comparison with the state-of-the-art results on UTD-MHAD

Feature engineering	Method	Accuracy (%)
Handcrafted	Kinect and inertial [7][a]	79.1
	Kinect and inertial fusion [8][a]	91.5
	ELC-KSVD [70]	76.1
	Cov3DJ [22]	85.5
	Our method	95.3
CNN	SOS_ based CNN [20]	86.9
	JTM_ CNN [61]	85.8

[a] Notice that in this method, in addition to skeleton, depth information is also used

Table 6 Three action subsets of *MSR action 3D*

AS1	AS2	AS3
Horizontal arm wave	High arm wave	High throw
Hammer	Hand catch	Forward kick
Forward punch	Draw x	Side kick
High throw	Draw tick	Jogging
Hand clap	Draw circle	Tennis swing
Bend	Two hand wave	Tennis serve
Tennis serve	Forward kick	Golf swing
Pickup and throw	Side boxing	Pickup and throw

ilar motions were grouped together. These categories were used for evaluating distinctive ability of algorithms for recognizing actions with similar structure. Sub-category AS3 contained actions were consisted of complex body dynamics and were used for evaluation of diversity of a method. The overall performance of a system is obtained by averaging the performance of sub-categories.

The second experimental protocol which was suggested in [60] kept all of the 20 actions in a single set for training and testing without splitting the dataset. This makes the classification even harder compared to the first setting. In our experiments, we use both of the settings. For the first protocol, we use cross-subject cross-validation similar to [54]. We consider half of the subjects (1, 3, 5, 7 and 9) for training and the other half (2, 4, 6, 8 and 10) for testing. By examining all of the possible scenarios for the parameters indicated in Table 2, we obtained the best performance by setting the temporal offset to 4, number of key poses to 100 and number of neurons to 3100 on MSR action 3D dataset. For the second protocol, we detect value of the temporal offset as 5, number of key poses as 160 and number of neurons as 1300. The evaluation of these parameters is illustrated in Figs. 4, 5 and 6, respectively. The performance of our method on MSR action 3D dataset with the two protocols and their comparisons with skeleton-based state-of-the-art methods are shown in Table 7. (Results of the second protocol are under the column indicated with all.) Depending on the feature type, the methods are categorized into handcrafted or automatic types.

Our proposed method achieved acceptable performance among the handcrafted methods when features are calculated only in Euclidean space without transformation into

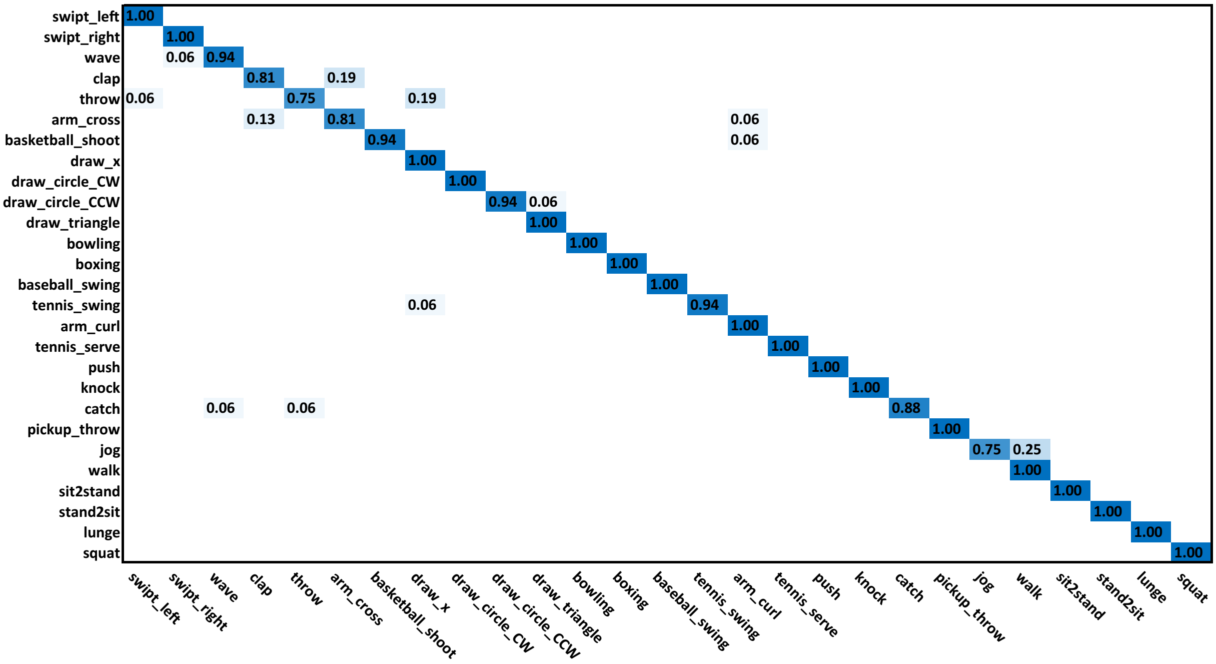


Fig. 9 Confusion matrix of *UTD-MHAD dataset*

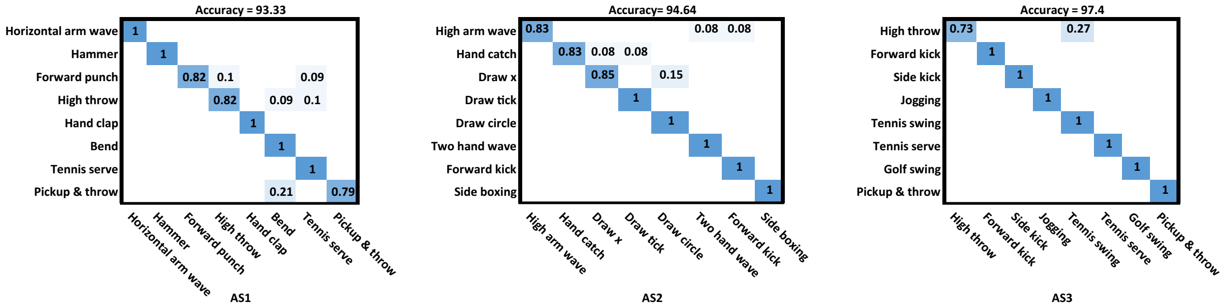
Table 7 Comparison with the state-of-the-art results on *MSR action 3D*

Feature engineering	Method	Accuracy (%)				
		AS1	AS2	AS3	Average	All
Handcrafted	Pose-based [57]	–	–	–	90.2	–
	HOJ3D [62]	–	–	–	78.9	–
	Lie group [54]	95.3	83.9	98.2	92.5	89.4
	Spatiotemporal SHs [65]	89.7	91.7	92.5	90.9	–
	Pairwise joints [32]	–	–	–	93.8	–
	RRV [17]	–	–	–	–	93.4
	Trajectory let [44]	96.4	97.5	100	97.9	–
	Our method	94.3	94.6	97.7	95.4	91.9
Learned representations	LMNN [35]	–	–	–	97.1	–
	Moving pose lets [50]	89.8	93.5	97.0	93.5	93.6
	Max-margin multitask [69]	–	–	–	95.6	90.5
RNN or LSTM	HBRNN-L [13]	93.3	94.6	95.5	94.5	–
	ST-LSTM [31]	–	–	–	94.8	–
	TS-LSTM [28]	95.2	96.4	100	97.2	–

another space such as [54,65]. The approaches such as [35] that employed data mining techniques to select distinctive features achieved superior results. However, performance improvement in action recognition in these methods coincided with an increase in computational cost particularly in the training phase. As shown in Table 7, our methods generated relatively better results compared to [13,50, 65] on AS3 which contain actions with complex structures.

In the first protocol, compared to the other two subcategories, actions in sub-category AS1 are more challenging for our framework and resulted in a less accuracy in performance due to complexity of the actions. It can be clearly seen from the confusion matrix (Fig. 10) that "pickup&throw" action is correctly classified in 79% of the test samples. However, this action misclassified in 21% of the samples as "bend." In AS2 sub-category, the highest misclassification rate happens in "hand catch" action, where it is misclassified in 8% of the samples as "draw x" and in 8% as "draw tick." In the final sub-category AS3, the highest misclassification rate belongs to "high throw" action, where it is misclassified in 27% of the samples as "tennis swing." Lack of producing distinctive key poses for each action class is the main reason for recognition failure. For example, in case of "high throw" action, our approach generates the same key poses with different temporal orders compared to the other two confusing actions. Even though the generated poses are comprised time information, during complex action encoding procedure, the framework loses the temporal order of poses in the sequence for some of the actions. The confusion matrix of the second protocol shown in Fig. 11 indicates that the highest classification error occurs during "hand catch" action with 58% correct recognition including 17% as "high throw" and the rest as "horizontal arm wave," "high arm wave" and "side boxing" with 8% each. This can be attributed to the similar poses available in the sequence of actions. Except these actions, "forward punch" and "high throw" actions have the highest error rates and are correctly classified with 73%. However, our method is capable of correctly recognizing 10 actions out of 20 with 100% accuracy by using this protocol.

Studies conducted on MSRC-12 dataset used two common cross-subject experimental protocols in their evaluations. Leave-one-out protocol that has been used in [22] took action instances of 29 subjects for training and the remaining one subject for testing. This process was replicated for all of the subjects in the dataset and the final result was reported as average accuracy of all the performances. Second protocol [17,44] used half of the instances (instances of odd numbered subjects) for training and the other half for testing. We use the second protocol in our evaluations. Optimal parameters are detected and set according to Table 2. The results of our evaluations and comparisons with skeleton-based state-of-the-art methods are reported in Table 8. As shown in Table 8, our method achieves competitive performance compared to handcrafted methods and surpasses representation learning methods which is expected to perform better as the number of instances increases. The conducted evaluations also indicate that our method is scalable and performs reliably as the number of instances grows. (MSRC-12 size is almost 7 times UTD-MHAD dataset.) The second protocol is taken into account in order to evaluate our method on this dataset. Considering Table 2 and by examining all the possible values of the inputs through this dataset, the best performance is achieved with temporal offset of 9, number of key poses of 150 and number of neurons of 600. Figure 12 shows that



AS1 – Accuracy = 93.33

	Horizontal arm wave	Hammer	Forward punch	High throw	Hand clap	Bend	Tennis serve	Pickup & throw
Horizontal arm wave	1							
Hammer		1						
Forward punch			0.82	0.1			0.09	
High throw				0.82		0.09	0.1	
Hand clap					1			
Bend						1		
Tennis serve							1	
Pickup & throw						0.21		0.79

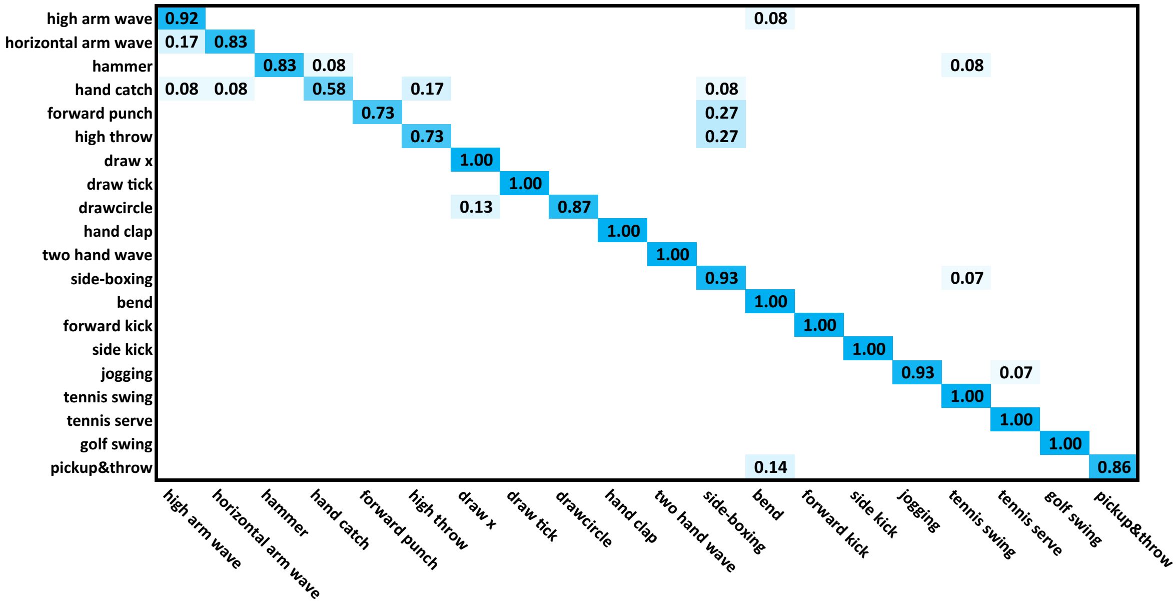
AS2 – Accuracy= 94.64

	High arm wave	Hand catch	Draw x	Draw tick	Draw circle	Two hand wave	Forward kick	Side boxing
High arm wave	0.83					0.08	0.08	
Hand catch		0.83	0.08	0.08				
Draw x			0.85		0.15			
Draw tick				1				
Draw circle					1			
Two hand wave						1		
Forward kick							1	
Side boxing								1

AS3 – Accuracy = 97.4

	High throw	Forward kick	Side kick	Jogging	Tennis swing	Tennis serve	Golf swing	Pickup & throw
High throw	0.73				0.27			
Forward kick		1						
Side kick			1					
Jogging				1				
Tennis swing					1			
Tennis serve						1		
Golf swing							1	
Pickup & throw								1

Fig. 10 Confusion matrices of MSR action 3D dataset



	high arm wave	horizontal arm wave	hammer	hand catch	forward punch	high throw	draw x	draw tick	drawcircle	hand clap	two hand wave	side-boxing	bend	forward kick	side kick	jogging	tennis swing	tennis serve	golf swing	pickup&throw
high arm wave	0.92												0.08							
horizontal arm wave	0.17	0.83																		
hammer			0.83	0.08													0.08			
hand catch	0.08	0.08		0.58		0.17						0.08								
forward punch					0.73							0.27								
high throw						0.73						0.27								
draw x							1.00													
draw tick								1.00												
drawcircle							0.13		0.87											
hand clap										1.00										
two hand wave											1.00									
side-boxing												0.93					0.07			
bend													1.00							
forward kick														1.00						
side kick															1.00					
jogging																0.93		0.07		
tennis swing																	1.00			
tennis serve																		1.00		
golf swing																			1.00	
pickup&throw													0.14							0.86

Fig. 11 Confusion matrix of MSR action 3D dataset (All)

due to the existence of similar poses in the actions, the highest recognition error of our method belongs to “beat both” and “had enough” actions with 81 and 86% accuracy, respectively.

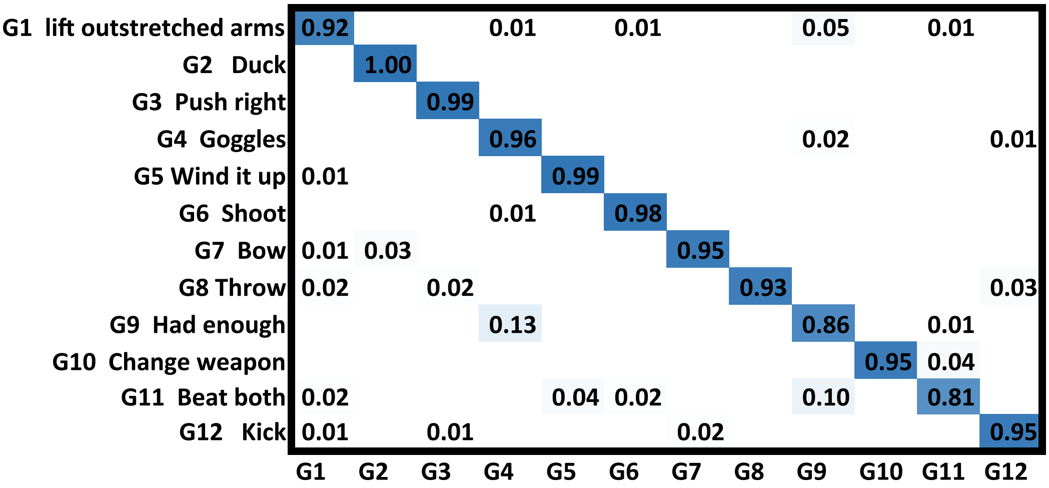
Based on our experiments, small-sized codebook does not generate sufficient diverse code words to discriminate all of the actions and the one with a large size is highly prone to be affected by noise. Most of the key pose-based methods usually use HMM to define an action and model temporality and accordingly, the number of the generated key-poses is limited. One of the main privileges of our method to the key pose-based methods is that rather than generating action sequence using the key-poses, we find available key poses using a dictionary populated with sufficient key poses where the absence of a key-pose is still a valuable information. However, the higher number of key poses may increase the noise in recognition. Tuning the number of the key poses is an important task having a great impact on robustness of the recognition in our method.

5 Conclusion and future work

In this study, we proposed a novel bag-of-poses framework for 3D action recognition based on a set of predefined spatiotemporal poses. Most of the studies available in the literature regarding pose-based action recognition have used generative or bag-of-poses approaches. The main disadvantages of the generative methods are their exceeding needs for training data and challenging parameter tuning which is usually performed manually. Accordingly, the main drawback of the bag-of-poses approaches is not to consider the concept of time among the poses through encoding an action. As a solution, our main objective is to improve the bag-of-poses approach by embedding temporal information using the key pose descriptors. The proposed descriptor enables us to distinguish between the two poses with the same skeleton configurations and different temporal order in an action sequence. The pose descriptor is extracted from Euclidean coordinates of the skeleton joints without transforming the

Table 8 Comparison with the state-of-the-art results on *MSRC-12*

Feature engineering	Method	Accuracy (%)	
		Cross-subject	*LoSubO*
Handcrafted	Cov3DJ [22]	91.7	93.6
	RRV [17]	93.8	94.7
	Hierarchical model [24]	–	94.6
	ASM [23]	–	97.6
	ELC-KSVD [70]	90.2	–
	Position offset + NBNN [34]	–	90.2
	Trajectory let [44]	94.9	95.1
	Our method	94.2	–
Learned representations	DF selected features [38]	–	94.03(5-fold)
CNN	ConvNets [12]	84.4	–
	JTM_CNN [61]	93.1	–
	SOS_based CNN [20]	94.2	–
	Enhanced skeleton visualization [33]	96.6	–



	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12
G1 lift outstretched arms	0.92			0.01		0.01			0.05		0.01	
G2 Duck		1.00										
G3 Push right			0.99									
G4 Goggles				0.96					0.02			0.01
G5 Wind it up	0.01				0.99							
G6 Shoot				0.01		0.98						
G7 Bow	0.01	0.03					0.95					
G8 Throw	0.02		0.02					0.93				0.03
G9 Had enough				0.13					0.86		0.01	
G10 Change weapon										0.95	0.04	
G11 Beat both	0.02				0.04	0.02			0.10		0.81	
G12 Kick	0.01		0.01				0.02					0.95

Fig. 12 Confusion matrix of MSRC-12

coordinates into another space. The suggested framework is validated with five publicly available benchmark 3D action datasets and produced state-of-the-art results on the three datasets, while competitive results on the fourth and fifth datasets. In our method, the major aspect that needs to be improved is to recognize interactive actions between the subjects. This is mainly because the framework does not benefit from the context information and interaction with the objects in the environment. As a future study, we will investigate this subject to improve the results by utilizing both depth and contextual information.

References

1. Aggarwal, J., Xia, L.: Human activity recognition from 3d data: a review. Pattern Recognit. Lett. **48**, 70–80 (2014)
2. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. (CSUR) **43**(3), 16 (2011)
3. Amor, B.B., Su, J., Srivastava, A.: Action recognition using rate-invariant analysis of skeletal shape trajectories. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 1–13 (2016)
4. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. arXiv preprint arXiv:1611.08050 (2016)
5. Chaaraoui, A.A., Padilla-Lpez, J.R., Climent-Prez, P., Flrez-Revuelta, F.: Evolutionary joint selection to improve human action recognition with rgb-d devices. Expert Syst. Appl. **41**(3), 786–794 (2014)
6. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 27 (2011)
7. Chen, C., Jafari, R., Kehtarnavaz, N.: Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: IEEE International Conference on Image Processing (ICIP), pp. 168–172. IEEE (2015)
8. Chen, C., Jafari, R., Kehtarnavaz, N.: A real-time human action recognition system using depth and inertial sensor fusion. IEEE Sens. J. **16**(3), 773–781 (2016)

9. Chen, X., Koskela, M.: Skeleton-based action recognition with extreme learning machines. Neurocomputing **149**, 387–396 (2015)
10. Chron, G., Laptev, I., Schmid, C.: P-cnn: Pose-based cnn features for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 3218–3226
11. Dawn, D.D., Shaikh, S.H.: A comprehensive survey of human action recognition with spatio-temporal interest point (stip) detector. Vis. Comput. **32**(3), 289–306 (2016)
12. Du, Y., Fu, Y., Wang, L.: Skeleton based action recognition with convolutional neural network. In: 3rd IAPR Asian Conference on Pattern Recognition (ACPR), pp. 579–583. IEEE (2015)
13. Du, Y., Wang, W., Wang, L.: Hierarchical recurrent neural network for skeleton based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1110–1118
14. Eweiwi, A., Cheema, M.S., Bauckhage, C., Gall, J.: Efficient pose-based action recognition. In: Asian Conference on Computer Vision, pp. 428–443. Springer
15. Fothergill, S., Mentis, H., Kohli, P., Nowozin, S.: Instructing people for training gestural interactive systems. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1737–1746. ACM
16. Gaglio, S., Re, G.L., Morana, M.: Human activity recognition process using 3-d posture data. IEEE Trans. Hum. Mach. Syst. **45**(5), 586–597 (2015)
17. Guo, Y., Li, Y., Shao, Z.: Rrv: A spatiotemporal descriptor for rigid body motion recognition. IEEE Trans. Cybern. **99**, 1–13 (2018). https://doi.org/10.1109/TCYB.2017.2705227
18. Han, F., Reily, B., Hoff, W., Zhang, H.: Space-time representation of people based on 3d skeletal data: a review. Comput. Vis. Image Underst. **158**, 85–105 (2017)
19. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced computer vision with microsoft kinect sensor: a review. IEEE Trans. Cybern. **43**(5), 1318–1334 (2013)
20. Hou, Y., Li, Z., Wang, P., Li, W.: Skeleton optical spectra based action recognition using convolutional neural networks. IEEE Trans. Circuits Syst. Video Technol. **99**, 1–1 (2017). https://doi.org/10.1109/TCSVT.2016.2628339
21. Huang, G.B., Zhu, Q.Y., Siew, C.K.: Extreme learning machine: theory and applications. Neurocomputing **70**(1), 489–501 (2006)
22. Hussein, M.E., Torki, M., Gowayyed, M.A., El-Saban, M.: Human action recognition using a temporal hierarchy of covariance descriptors on 3d joint locations. In: Twenty-Third International Joint Conference on Artificial Intelligence
23. Ibaez, R., Soria, I., Teyseyre, A., Rodrguez, G., Campo, M.: Approximate string matching: a lightweight approach to recognize gestures with kinect. Pattern Recognit. **62**, 73–86 (2017)
24. Jiang, X., Zhong, F., Peng, Q., Qin, X.: Online robust action recognition based on a hierarchical model. Vis. Comput. **30**(9), 1021–1033 (2014)
25. Johansson, G.: Visual Motion Perception. Scientific American, New York (1975)
26. Kapsouras, I., Nikolaidis, N.: Action recognition on motion capture data using a dynemes and forward differences representation. J. Vis. Commun. Image Represent. **25**(6), 1432–1445 (2014)
27. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. Nature **521**(7553), 436–444 (2015)
28. Lee, I., Kim, D., Kang, S., Lee, S.: Ensemble deep learning for skeleton-based action recognition using temporal sliding lstm networks. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 1012–1020
29. Li, W., Zhang, Z., Liu, Z.: Action recognition based on a bag of 3d points. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 9–14. IEEE (2010)
30. Lillo, I., Niebles, J.C., Soto, A.: Sparse composition of body poses and atomic actions for human activity recognition in rgb-d videos. Image Vis. Comput. **59**, 63–75 (2017)
31. Liu, J., Shahroudy, A., Xu, D., Chichung, A.K., Wang, G.: Skeleton-based action recognition using spatio-temporal lstm network with trust gates. IEEE Trans. Pattern Anal. Mach. Intell. **99**, 1–1 (2017). https://doi.org/10.1109/TPAMI.2017.2771306
32. Liu, M., Chen, C., Liu, H.: Learning informative pairwise joints with energy-based temporal pyramid for 3d action recognition. In: IEEE International Conference on Multimedia and Expo (ICME), pp. 901–906. IEEE (2017)
33. Liu, M., Liu, H., Chen, C.: Enhanced skeleton visualization for view invariant human action recognition. Pattern Recognit. **68**, 346–362 (2017)
34. Lu, G., Zhou, Y., Li, X., Kudo, M.: Efficient action recognition via local position offset of 3d skeletal body joints. Multimed. Tools Appl. **75**(6), 3479–3494 (2016)
35. Luvizon, D.C., Tabia, H., Picard, D.: Learning features combination for human action recognition from skeleton sequences. Pattern Recognit. Lett. **99**, 13–20 (2017)
36. Minhas, R., Baradarani, A., Seifzadeh, S., Wu, Q.J.: Human action recognition using extreme learning machine based on visual vocabularies. Neurocomputing **73**(10), 1906–1917 (2010)
37. Negin, F., Akgl, C.B., Yksel, K.A., Eril, A.: An rdf-based action recognition framework with feature selection capability, considering therapy exercises utilizing depth cameras. J. Theor. Appl. Comput. Sci. **8**(3), 3–22 (2014)
38. Negin, F., zdemir, F., Akgl, C.B., Yksel, K.A., Eril, A.: A decision forest based feature selection framework for action recognition from rgb-depth cameras. In: International Conference Image Analysis and Recognition, pp. 648–657. Springer
39. Nunes, U.M., Faria, D.R., Peixoto, P.: A human activity recognition framework using max–min features and key poses with differential evolution random forests classifier. Pattern Recognit. Lett. **99**, 21–31 (2017)
40. Parisi, G.I., Weber, C., Wermter, S.: Self-organizing neural integration of pose–motion features for human action recognition. Front. Neurorobot. **9**, 3 (2015)
41. Peng, X., Wang, L., Wang, X., Qiao, Y.: Bag of visual words and fusion methods for action recognition: comprehensive study and good practice. Comput. Vis. Image Underst. **150**, 109–125 (2016)
42. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**(6), 976–990 (2010)
43. Presti, L.L., La Cascia, M.: 3d skeleton-based human action classification: a survey. Pattern Recognit. **53**, 130–147 (2016)
44. Qiao, R., Liu, L., Shen, C., van den Hengel, A.: Learning discriminative trajectorylet detector sets for accurate skeleton-based action recognition. Pattern Recognit. **66**, 202–212 (2017)
45. Ramanathan, M., Yau, W.Y., Teoh, E.K.: Human action recognition with video data: research and evaluation challenges. IEEE Trans. Hum. Mach. Syst. **44**(5), 650–663 (2014)
46. Sadanand, S., Corso, J.J.: Action bank: a high-level representation of activity in video. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1234–1241. IEEE (2012)
47. Shan, J., Akella, S.: 3d human action segmentation and recognition using pose kinetic energy. In: IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 69–75. IEEE (2014)
48. Shotton, J., Sharp, T., Kipman, A., Fitzgibbon, A., Finocchio, M., Blake, A., Cook, M., Moore, R.: Real-time human pose recognition in parts from single depth images. Commun. ACM **56**(1), 116–124 (2013)
49. Sung, J., Ponce, C., Selman, B., Saxena, A.: Unstructured human activity detection from rgbd images. In: IEEE International Conference on Robotics and Automation (ICRA), pp. 842–849. IEEE (2012)

50. Tao, L., Vidal, R.: Moving poselets: a discriminative and interpretable skeletal motion representation for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision Workshops, pp. 61–69
51. Tran, D., Torresani, L.: Exmoves: mid-level features for efficient action recognition and video analysis. Int. J. Comput. Vis. **119**(3), 239–253 (2016)
52. Varol, G., Salah, A.A.: Efficient large-scale action recognition in videos using extreme learning machines. Expert Syst. Appl. **42**(21), 8274–8282 (2015)
53. Veeriah, V., Zhuang, N., Qi, G.J.: Differential recurrent neural networks for action recognition. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 4041–4049
54. Vemulapalli, R., Arrate, F., Chellappa, R.: Human action recognition by representing 3d skeletons as points in a lie group. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 588–595
55. Vemulapalli, R., Arrate, F., Chellappa, R.: R3dg features: relative 3d geometry-based skeletal representations for human action recognition. Comput. Vis. Image Underst. **152**, 155–166 (2016)
56. Vishwakarma, S., Agrawal, A.: A survey on activity recognition and behavior understanding in video surveillance. Vis. Comput. **29**(10), 983–1009 (2013)
57. Wang, C., Wang, Y., Yuille, A.L.: An approach to pose-based action recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 915–922
58. Wang, C., Wang, Y., Yuille, A.L.: Mining 3d key-pose-motifs for action recognition. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 2639–2647. IEEE (2016)
59. Wang, J., Liu, Z., Wu, Y.: Learning Actionlet Ensemble for 3D Human Action Recognition, pp. 11–40. Springer, New York (2014)
60. Wang, J., Liu, Z., Wu, Y., Yuan, J.: Mining actionlet ensemble for action recognition with depth cameras. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1290–1297. IEEE (2012)
61. Wang, P., Li, Z., Hou, Y., Li, W.: Action recognition based on joint trajectory maps using convolutional neural networks. In: Proceedings of the 2016 ACM on Multimedia Conference, pp. 102–106. ACM
62. Xia, L., Chen, C.C., Aggarwal, J.: View invariant human action recognition using histograms of 3d joints. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 20–27. IEEE (2012)
63. Yang, Y., Deng, C., Tao, D., Zhang, S., Liu, W., Gao, X.: Latent max-margin multitask learning with skelets for 3-d action recognition. IEEE Trans. Cybern. **47**(2), 439–448 (2017)
64. Yao, A., Gall, J., Fanelli, G., Van Gool, L.: Does human action recognition benefit from pose estimation? In: Proceedings of the 22nd British Machine Vision Conference-BMVC (2011)
65. Youssef, C.: Spatiotemporal representation of 3d skeleton joints-based action recognition using modified spherical harmonics. Pattern Recognit. Lett. **83**, 32–41 (2016)
66. Zanfir, M., Leordeanu, M., Sminchisescu, C.: The moving pose: an efficient 3d kinematics descriptor for low-latency action recognition and detection. In: Proceedings of the IEEE International Conference on Computer Vision, pp. 2752–2759
67. Zelnik-Manor, L., Irani, M.: Event-based analysis of video. In: Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2001, vol. 2, pp. II–II. IEEE (2001)
68. Zhang, J., Li, W., Ogunbona, P.O., Wang, P., Tang, C.: Rgb-d-based action recognition datasets: a survey. Pattern Recognit. **60**, 86–105 (2016)
69. Zhang, S., Liu, X., Xiao, J.: On geometric features for skeleton-based action recognition using multilayer lstm networks. In: IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 148–157. IEEE (2017)
70. Zhou, L., Li, W., Zhang, Y., Ogunbona, P., Nguyen, D.T., Zhang, H.: Discriminative key pose extraction using extended lc-ksvd for action recognition. In: International Conference on Digital lmage Computing: Techniques and Applications (DlCTA), pp. 1–8. IEEE (2014)
71. Zhu, F., Shao, L., Xie, J., Fang, Y.: From handcrafted to learned representations for human action recognition: a survey. Image Vis. Comput. **55**, 42–52 (2016)
72. Zhu, G., Zhang, L., Shen, P., Song, J.: Human action recognition using multi-layer codebooks of key poses and atomic motions. Signal Process. Image Commun. **42**, 19–30 (2016)
73. Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., Xie, X.: Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks. AAAI **2**, 8 (2016)
74. Zhu, Y., Chen, W., Guo, G.: Fusing multiple features for depth-based action recognition. ACM Trans. Intell. Syst. Technol. (TIST) **6**(2), 18 (2015)

Saeid Agahian received a MS degree in 2012 in Computer Engineering Faculty at the Karadeniz Technical University, Trabzon, Turkey, and is a Ph.D. candidate under supervision of Prof. Dr. Cemal Köse since 2013 in computer engineering at the Karadeniz Technical University, Trabzon, Turkey. His research interests include image processing, pattern recognition, combinatorial optimization (time tabling), computer vision and EEG signal processing.

Farhood Negin is a Ph.D. candidate in Computer Science at INRIA Sophia Antipolis, France. He has also obtained his M.Sc. from Sabanci University in Istanbul, Turkey. Previously, he worked in a reputable industry-connected European research project in computer vision, artificial intelligence and assistive technologies such as VIPSAFE, Dem@Care and SAFEE projects. He is a member of Cognition Behaviour Technology (Cobtek) team and also The European Network on Integrating Vision and Language (iV&L Net). He is currently working in Spatio-Temporal Activity Recognition Systems (STARS) team at INRIA and working on his thesis in order to develop next-generation technologies in computer vision and human–computer interaction with a focus on activity and gesture recognition.

Cemal Köse received his BS and MS degrees in the Department of Electrical and Electronic Engineering from Karadeniz Technical University, Trabzon, Turkey, in 1986 and 1990, respectively. He received a Ph.D. in the Department of Computer Science from the University of Bristol, Bristol, UK, in 1997. He is currently a professor in the Department of Computer Engineering, Karadeniz Technical University, Trabzon, Turkey. His research interests include parallel processing, pattern recognition, natural language processing, data mining computer vision and computer graphics.